Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.5809

BENCH$_4$T^3: A Framework to Create Benchmarks for Text-to-Triples Alignment Generation

Victor Jesus Sotelo Chico [**Institute of Computing, University of Campinas** | ***v265173@dac.unicamp.br***]
André Gomes Regino [**DIMEC,Center for Information Technology Renato Archer** | ***aregino@cti.gov.br***]
Julio Cesar dos Reis [**Institute of Computing, University of Campinas** | ***jreis@ic.unicamp.br***]

Received: 31 March 2025 • **Accepted:** 26 August 2025 • **Published:** 06 February 2026

Abstract Integrating Large Language Models (LLMs) with Knowledge Graphs (KGs) can significantly enhance their capabilities, leveraging LLMs' text generation skills with KGs' explanatory power. However, establishing this connection is challenging and demands proper alignment between unstructured texts and triples. Building benchmarks demands massive human effort in data curation and translation for non-English languages. The demand for adequate benchmarks for validation purposes negatively impacts research advancements. This study proposes an end-to-end framework to guide the automatic construction of text-to-triple alignment benchmarks for any language, using KGs as input. Our solution extracts relations from input triples and processes them to create accurately mapped texts. The proposed pipeline utilizes data curation through prompt engineering and data augmentation to enhance diversity in the generated examples. We experimentally evaluate our framework for creating a bimodal representation of RDF triples and natural language texts, assessing its ability to generate natural language from these triples. A key focus is on developing a benchmark for the underrepresented Portuguese language, facilitating the construction of models that connect structured data (triples) with text. Our solution is suited to creating a benchmark to improve alignment between KG triples and text data. The results indicate that the generated benchmark outperforms the results of existing solutions. The generative approach benefits from our Portuguese benchmark, achieving competitive results compared to established literature benchmarks. Our solution enables automatic generation of benchmarks for aligning triples and text.

Keywords: Semantic Web RDF Triples to Text Large Language Models

1 Introduction

Development of Large Language Models (LLMs) [Min *et al.*, 2023] has established a new state-of-the-art for generative artificial intelligence and language representation, allowing easier integrations across domains such as e-commerce [Wang and Na, 2024], medical [Thirunavukarasu *et al.*, 2023], education [Lieb and Goel, 2024], and others. LLMs are mainly utilized in question-answering systems to respond to human inquiries. For example, they can provide information on pricing or products before a sale in the e-commerce sector. When answering questions, LLMs rely on the knowledge they gained during their training. However, they can sometimes produce inaccurate answers or "hallucinate" [Huang *et al.*, 2024], particularly when faced with questions outside their scope of knowledge and understanding.

To limit the scope of the answer, solutions have explored techniques like Retrieval-Augmented Generation (RAG) [Fan *et al.*, 2024] to integrate external knowledge. This approach enhances the capabilities of LLMs by allowing them to access information beyond their training data. An example is GRAPH-RAG, which utilizes Knowledge Graphs [Ji *et al.*, 2022] as the external knowledge source. Currently, applications process raw text to build KGs, as structured KGs are more explainable than LLMs and can yield better responses [Pan *et al.*, 2024].

This context relies on indirect communication between KGs and LLMs, demanding text transformation into KGs or vice versa for interaction. To assess the effectiveness of these transformations, it is essential to have well-designed benchmarks that can be adapted across several knowledge domains. While there are existing benchmarks like *WEBNLG* [Gardent *et al.*, 2017], and *Text2KGBench* [Mihindukulasooriya *et al.*, 2023] in this scenario, we observe a complete lack of such resources for non-English languages underrepresented in datasets. There is also a lack of methods to create these benchmarks for specific tasks. Recent studies have highlighted the need for additional benchmarks to contribute to the task of training bimodal encoders, which may help improve information retrieval systems [Chico and dos Reis, 2024].

This investigation designs, develops, and evaluates a novel original framework, named BENCH$_4$T3, for automatically creating benchmarks for triples-to-text alignment. Our solution addresses the lack of triple-text benchmarks, reduces the need for human intervention, and enhances the accessibility of the benchmark creation process for stakeholders. In particular, our solution generates the PoRTA dataset (A Portuguese benchmark generated by applying our framework), suited for the investigated task and addresses the Portuguese language.

This study addresses the following original research questions: **RQ1**: Can BENCH$_4$T^3 contribute to generating benchmarks for underrepresented languages? **RQ2**: Are the embedding models fine-tuned with our generated benchmark better than the others created with existing benchmarks? **RQ3:** Is our benchmark suitable for evaluating natural language generation (NLG) models, particularly for transforming structured triples into coherent text comparable to existing benchmarks? **RQ4:** Do models fine-tuned on our generated benchmark outperform few-shot prompting approaches in generating high-quality Portuguese text? **RQ5:** Does applying data augmentation to build our benchmark improve the adequacy and quality of generated texts?

This research achieves the following contributions:

- An original framework named BENCH$_4$T^3 to enable people (data scientists and engineers) to generate new RDF-to-text benchmarks.
- A high-quality benchmark constructed specifically for RDF-to-text tasks in the Portuguese language – Portuguese RDF-Text Alignment (PoRTA) – supporting advancements in natural language generation and Semantic Web research. The generated benchmark combines linguistic variability, semantic accuracy, and cultural relevance, making it a valuable resource.

Our approach introduces a framework for creating a benchmark to align triples and text through several key tasks. First, the framework performs triple extraction to retrieve facts from KGs by querying them using SPARQL [Pérez *et al.*, 2009]. Second, the solution implements a triple verbalizer, employing prompt engineering to convert the extracted triples into natural language text. Additionally, the framework includes a step where an LLM acts as a data curator to filter out incorrect verbalizations. Finally, the solution explores an LLM for data augmentation, generating paraphrase examples that increase the diversity of grammatical structures and enhance the overall number of examples.

Our evaluation ensures a systematic assessment through fine-tuning and retrieval-based tasks by following three steps. Our experimental evaluations utilize the generated benchmark to fine-tune pretrained encoders, enabling them to build a bimodal encoder for triples and text. An encoder model is trained using the training part of the generated benchmark dataset. The fine-tuned encoder generates embeddings for bimodal data (triple-text). We propose a retrieval task that utilizes a test subset of the benchmark to evaluate alignment accuracy using the Normalized Cumulative Discounted Gain (NCDG) metric [Järvelin and Kekäläinen, 2002]. This ensures the effectiveness of the retrieval process in both directions: recovering triples from text and retrieving text from triples. These configurations are applied to assess the correct alignment between text and triples.

Additionally, evaluating Natural Language Generation (NLG) involves existing techniques, such as fine-tuning a pretrained model and employing a Large Language Model (LLM) with a few-shot prompt to convert RDF triples into natural language text. Both approaches serve as generators. We use this generator with a test partition from the translated version of WEBNLG and our generated PoRTA benchmark. We compute grammar-sharing metrics, specifically ROUGE [Lin, 2004] and BLEU [Papineni *et al.*, 2002], to assess the exactness of the generated text compared to the gold standard references. In addition, we incorporate cosine similarity between sentence embeddings as a semantic evaluation metric to better capture meaning preservation. All produced codes and data are publicly available in the repository [1].

This research suggests that developing a customizable framework for automating benchmark generation can alleviate the shortage of language benchmarks, thereby facilitating further exploration of multimodal and multilingual data alignment. This study demonstrates that automatically generated benchmarks enhance the development of more effective bimodal encoders for triples and texts. Our benchmark enables the evaluation of natural text generation from structured triples, providing a reliable benchmark for assessing model performance in Portuguese NLG. Using our proposed BENCH$_4$T^3 framework to create a Portuguese benchmark dataset addresses the scarcity of Portuguese resources, paving the way for future studies on the interactions between KGs and language models.

The remainder of this article is organized as follows: Section 2 discusses a synthesis of related studies. Section 3 presents our framework BENCH$_4$T^3 to generate automatic triple-text benchmarks. Section 4 outlines our experimental methods, including the explored KGs, language models, and evaluation metrics. Section 5 reports on our experimental results. Section 6 discusses our findings. Finally, Section 7 summarizes our conclusions and suggests directions for future investigations.

2 Related Work

Gardent *et al.* [Gardent *et al.*, 2017] presented a pioneering approach for creating benchmarking datasets to train natural language generation (NLG) systems that handle micro-planning tasks, named WebNLG. Their work leveraged *DBpedia* to automatically generate data units, which were then paired with crowdsourced human-authored texts. WebNLG facilitated more sophisticated and varied training for NLG models than previous domain-specific datasets, which often restricted systems to predictable, domain-specific language.

The key novelty lies in its methodology of generating linguistically and syntactically varied text based on structured knowledge base data. Unlike earlier datasets that limited data to shallow syntactic structures, the WebNLG dataset employed advanced content selection techniques, allowing more profound levels of data representation. This created inputs of varying complexity, requiring NLG systems to learn a broader range of linguistic constructs and relationships. The dataset, applied in the WebNLG Challenge [Gardent *et al.*, 2017], aimed to motivate researchers to develop and refine models capable of handling data-to-text generation that closely mirrors real-world knowledge graph structures.

Ferreira *et al.* [Ferreira *et al.*, 2020] presented WebNLG+, a benchmark based on the original WEBNLG challenge. This new benchmark introduces Russian as an additional language, facilitating the alignment between RDF triples and English and Russian. The English dataset expands the categories from the original competition to sixteen categories. The authors conducted preprocessing on WEBNLG, correcting misspellings and addressing the verbalization of missing triples. They ensured that the test partition contained both known and unseen categories, as well as unseen entities. Additionally, the Russian dataset consists of nine categories translated from the English DBpedia. Unlike the previous challenge, the data presented serves as a benchmark for the RDF-to-text task, which is part of the WEBNLG challenge and provides a new

[1] `https://github.com/Visot/BENCH4T3`

way to validate the text-to-RDF task.

Agarwal *et al.* [Agarwal *et al.*, 2021] developed the *KELM* dataset by verbalizing the English Wikidata KG. This dataset is part of the TEKGEN framework, a sequence-to-sequence model designed to generate text from knowledge graph data. KELM comprises approximately eighteen million sentences covering around 45 million Wikidata triples. This extensive coverage offers a wide range of languages and addresses challenges related to entity diversity and relation representation.

The KELM corpus [Agarwal *et al.*, 2021] enhances other methods by focusing on semantic accuracy and reducing the risk of hallucination in the dataset creation task, in addition to its scale. Through alignment and filtering techniques, the authors ensured that the dataset maintains a close match between KG triples and generated text. They employed a fine-tuned BERT model for quality filtering and sequential training of T5 to prevent the generation of irrelevant information. This thorough verification process enhances the factual accuracy of the resulting dataset, making the KELM corpus a suitable pre-training source for language models.

Mihindukulasooriya *et al.* [Mihindukulasooriya *et al.*, 2023] introduced *Text2KGBench*, an ontology-driven benchmark for evaluating LLMs in generating KGs from unstructured text, guided by ontology-specific constraints. This benchmark differs from previous datasets in that it requires LLMs to adhere to domain-specific ontologies, such as Wikidata and DBpedia. This solution covers various concepts across film, music, and politics. *Text2KGBench* was developed with two datasets: *Wikidata-TekGen*, which includes $13,474$ sentences mapped to 10 ontologies, and DBpedia-WebNLG, which contains $4,860$ sentences across 19 ontologies. The benchmark also introduces seven unique evaluation metrics to measure the effectiveness of the model. These metrics evaluate the accuracy of fact extraction, alignment with the ontology, and the degree of hallucinations in subject, relation, and object outputs. The validation process includes manual annotations and the generation of unseen sentences.

Our present study provides an end-to-end solution fully designed and implemented to automate the creation of benchmarks for target languages. Our investigation constructs the first bidirectional text-to-triples benchmarking dataset specifically for the Portuguese language. This enables evaluations of RDF triple-to-text generation and semantic alignment for text and triples. This bidirectional design, combined with its focus on Brazilian Portuguese – a language with distinct semantic and syntactic nuances compared to those covered in existing resources – represents a novel and original contribution to natural language processing and KG research, filling an existing gap in available multilingual benchmarks.

3 The BENCH$_4$T^3 Framework

This section details the proposed framework, designed to facilitate the generation of RDF-to-text benchmarks. The framework systematically processes a Knowledge Graph $\mathcal{KG}$ to produce a benchmark comprising RDF triples and corresponding natural language verbalizations. For instance, given an RDF triple like <*Paris, isCapitalOf, France*>, our solution generates verbalized versions into natural language sentences, such as “Paris is the capital of France.”, “The capital city of France is Paris.” or even “France has Paris as its capital.”

Our framework was designed to provide diverse ways of expressing the same underlying knowledge. This diversity helps to align triples and their corresponding textual representations. By generating multiple variations of verbalizations for the same triple, stakeholders can assess how well natural language processing models understand and relate structured data to unstructured text.

The framework includes four primary steps, each addressing a specific aspect of the benchmark creation process. The subsection 3.1, subsection 3.2, subsection 3.3 and subsection 3.4 describe each step in detail, respectively. Figure 1 presents the data flow in the framework method. Algorithm 1 formalizes the methods to generate benchmarks.

3.1 Step 1: Triple Extraction

Step 1 of the framework selects the core entities, properties, and RDF triples to form the benchmark. This step is subdivided into four sub-steps: Entity Selection, Property Selection, Triple Extraction, and Unverbalized Triple Removal. The objective is to ensure a balanced, diverse, and semantically meaningful set of RDF triples suitable for verbalization tasks. Figure 1 presents the substeps of Step 1 (blue rectangle).

Step 1a) Entity Selection: The framework accepts a $\mathcal{KG}$, such as DBpedia or other Linked Open Data (LOD) datasets, as input. By default, it selects all entities and classes available in the $\mathcal{KG}$ (line 2 of Algorithm 1). For instance, in DBpedia, common entities or classes include *dbo:ComicsCharacter*, *dbo:Food*, *dbo:Monument* and *dbo:SportsTeam*.

This approach ensures comprehensive coverage, but may be computationally expensive for large $\mathcal{KG}$. To address this, the user can provide an optional Class List $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ to limit the scope to specific interest classes. This list should adhere to an existing ontology $\mathcal{O}$ composed of Classes ($\mathcal{C}$), Properties ($\mathcal{P}$) and relations ($\mathcal{R}$) (top of Figure 1). For example, a user interested only in the *dbo:Food* class would restrict the selection accordingly.

Step 1b) Property Selection: After defining the target classes, the framework proceeds to property selection (line 3 of Algorithm 1). By default, it retrieves all properties associated with the classes identified in Step 1a. For instance, for the class *dbo:Food*, relevant properties might include *dbo:taste*, *dbo:approximateCalories*, and *dbo:fat*. Similar to Step 1a, the user can refine the property selection by providing a Property List $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$, focusing on a specific subset of properties. Similar to the $\mathcal{C}$ list, the $\mathcal{P}$ list should adhere to the ontology $\mathcal{O}$.

Step 1c) Triple Extraction: Once the classes and properties are defined, the framework extracts RDF triples ($\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$) from the $\mathcal{KG}$ using SPARQL queries (line 4 of Algorithm 1). The framework ensures balanced extraction, aiming to retrieve an equal number of triples for each instance, property, and class. This balance is essential for creating a fair and unbiased benchmark and is critical for evaluating natural language generation models. For instance, consider the *dbo:Food* class and the property *dbo:approximateCalories*. The framework retrieves triples such as:

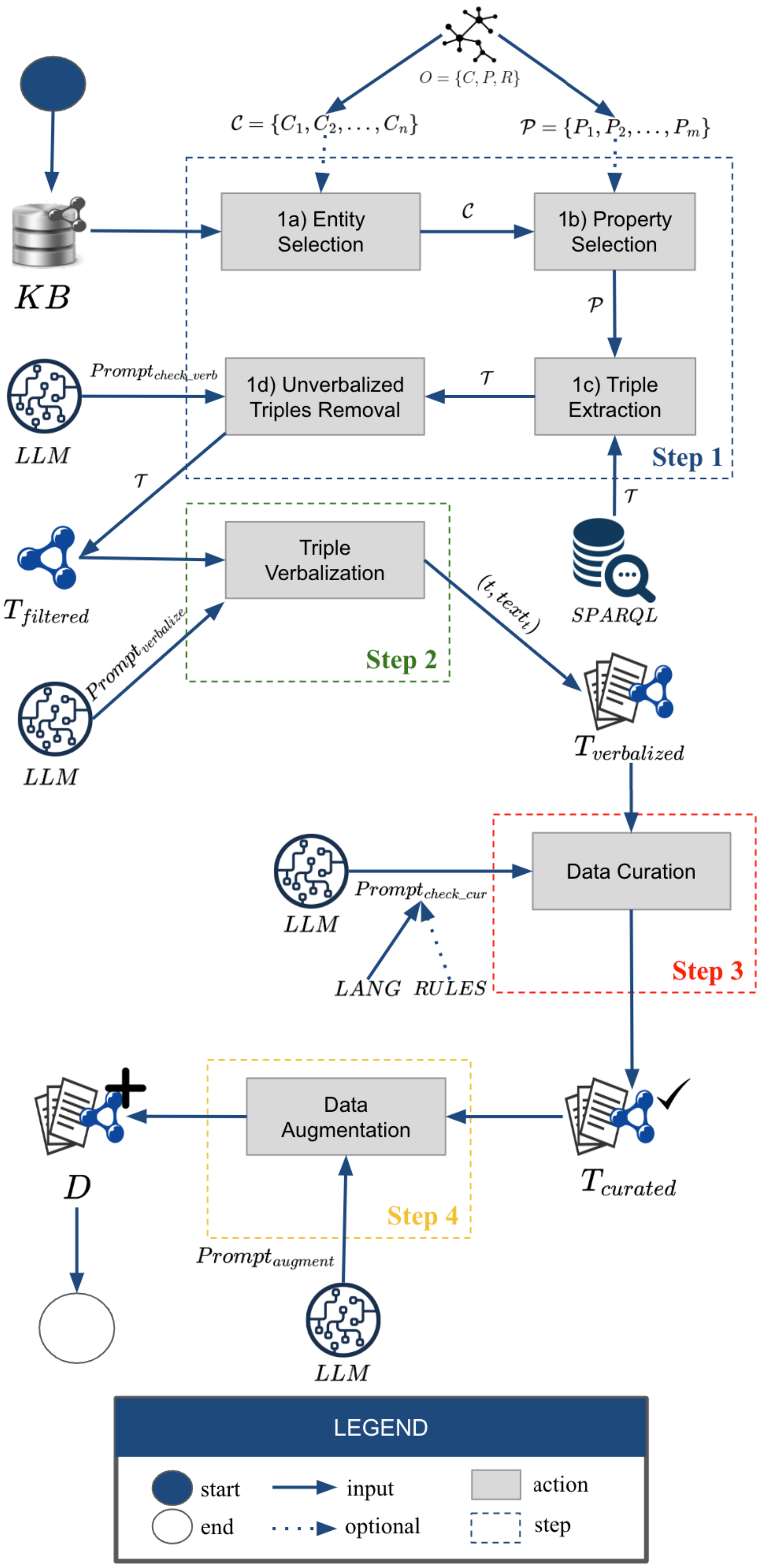


Figure 1. Overview of the proposed framework for RDF-to-text benchmark generation. The process consists of four steps: Triple Extraction (Step 1 - blue rectangle); Triple Verbalization (Step 2 - green rectangle); Data Curation (Step 3 - red rectangle); and Data Augmentation (Step 4 - yellow rectangle). Input/output flows are represented as arrows, actions as gray boxes, and optional components as dashed lines.

```
Algorithm 1 The BENCH$_4T^3$ Algorithm
Require: $O$, $KG$, $Prompt_{check\_verb}$, $Prompt_{verbalize}$, $Prompt_{check\_cur}$, $Prompt_{augment}$, $lang$, m, n, rules
Ensure: $\mathcal{D}$ (final augmented dataset)
 1: procedure SelectValidateAndAugmentTriples($O$, $KG$, $prompt$)
                                            ▷ Step 1: Triple Selection and Filtering
 2:     $E \leftarrow$ SelectEntitiesAndClasses($O$)                        ▷ Step 1a
 3:     $P \leftarrow$ SelectProperties($O$)                                ▷ Step 1b
 4:     $T \leftarrow$ ExtractTriples($KG$, $E$, $P$, n)                    ▷ Step 1c
 5:     $T_{filtered} \leftarrow \emptyset$
 6:     for $t \in T_{extracted}$ do
 7:         if CanBeVerbalized($t$, $Prompt_{check\_verb}$) then
 8:             $T_{filtered} \leftarrow T_{filtered} \cup \{t\}$           ▷ Step 1d
 9:         end if
10:     end for
                                                  ▷ Step 2: Triple Verbalization
11:     $T_{verbalized} \leftarrow \emptyset$
12:     for $t \in T_{filtered}$ do
13:         $text_{lang} \leftarrow$ GenerateVerbalization($t$, $Prompt_{verbalize}$, $lang$)
14:         $T_{verbalized} \leftarrow T_{verbalized} \cup \{(t, text_{lang})\}$
15:     end for
                                                        ▷ Step 3: Data Curation
16:     $T_{curated} \leftarrow \emptyset$
17:     for $(t, text_{lang}) \in T_{verbalized}$ do
18:         if IsAligned($t$, $text_{lang}$, $Prompt_{check\_cur}, rules$) then
19:             $T_{curated} \leftarrow T_{curated} \cup \{(t, text_{lang})\}$
20:         else
21:             FlagForReview($t$, $text_{lang}$)
22:         end if
23:     end for
                                                    ▷ Step 4: Data Augmentation
24:     $\mathcal{D} \leftarrow \emptyset$
25:     for $(t, text_{lang}) \in T_{curated}$ do
26:         $\mathcal{V} \leftarrow$ GenerateTextVariations($t, text_{lang}, Prompt_{augment}, m$)
27:         $\mathcal{D} \leftarrow \mathcal{D} \cup \{(t, \mathcal{V})\}$
28:     end for
29:     return $\mathcal{D}$                    ▷ Return the final augmented dataset
30: end procedure
```

```
<http://dbpedia.org/resource/Pasta> rdf:type dbo:
    Food .
<http://dbpedia.org/resource/Pasta> dbo:
    approximateCalories 371
```

Users can define the maximum number of triples to retrieve, ensuring the benchmark remains manageable while maintaining diversity (parameter n in line 4 of Algorithm 1).

Step 1d) Unverbalized Triple Removal: In the final substep, the framework removes triples that lack sufficient semantic information for verbalization (line 8 of Algorithm 1). For instance, the triple:

```
<http://dbpedia.org/resource/Pumpkin_Spice_Spam>
dbo:servingSize 56.7 .
```

It is considered incomplete because it does not specify the unit of measurement for the value of 56.7. Conversely, a triple like:

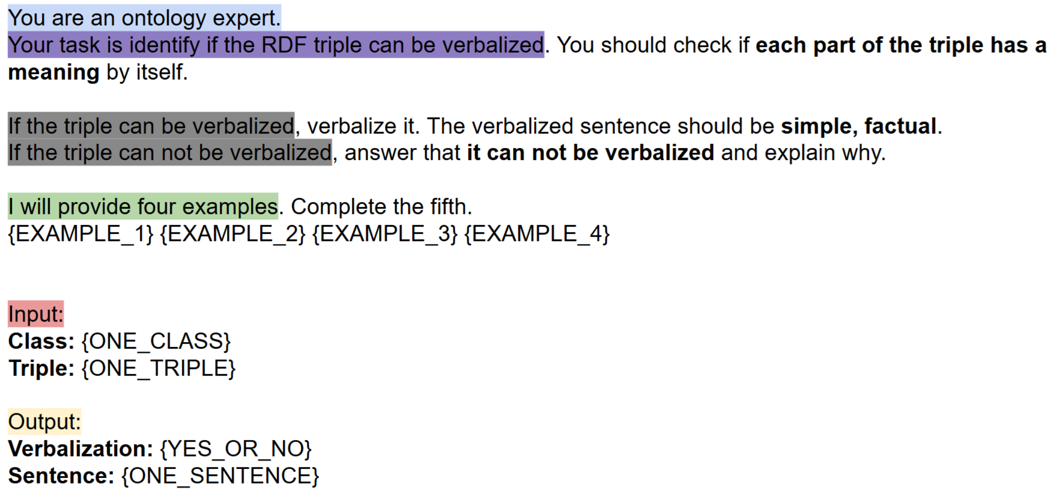
```
<http://dbpedia.org/resource/Samgye-tang>
dbo:servingTemperature "Hot or Warm" .
```

It is suitable because it contains complete and interpretable information. To identify unverbalized triples, the framework employs a language model, treated here as a black-box function that generates natural language responses (triple verbalizations) from textual prompts. A with a specialized prompt designed to evaluate the semantic adequacy of each triple ($Prompt_{check\_verb}$). The output of this sub-step is a set of unique, verbalizable triples ($T_{filtered}$) that serve as input for subsequent steps in the framework. Figure 2 shows the prompt template employed in the unverbalized detection task.

3.2 Step 2: Triple Verbalization

The second step of the framework (green rectangle in Figure 1) transforms RDF triples into natural language sentences. The goal is to produce a verbalized text in a target language for each triple generated in Step 1. This step ensures a one-to-one mapping between triples and texts, creating the core data necessary for the RDF-to-text benchmark.

The input to this step is the curated set of RDF triples $T_{filtered}$ outputted from Step 1. For each triple, the BENCH$_4$T^3 framework uses a language model to generate



You are an ontology expert.
Your task is identify if the RDF triple can be verbalized. You should check if **each part of the triple has a meaning** by itself.

If the triple can be verbalized, verbalize it. The verbalized sentence should be **simple, factual**.
If the triple can not be verbalized, answer that **it can not be verbalized** and explain why.

I will provide four examples. Complete the fifth.
{EXAMPLE_1} {EXAMPLE_2} {EXAMPLE_3} {EXAMPLE_4}

Input:
Class: {ONE_CLASS}
Triple: {ONE_TRIPLE}

Output:
Verbalization: {YES_OR_NO}
Sentence: {ONE_SENTENCE}

Figure 2. Prompt to identify unverbalized triples. The colors represent important parts of the prompt. In blue, how the LLM should behave and act; in purple, the task definition; in gray, the constraints; in green, the few-shot examples; in red, the input, and in yellow, the expected output.

verbalizations (line 13 of Algorithm 1). A tailored prompt $Prompt_{verbalize}$ is designed to: a) interpret and extract meaningful information from the triple; b) generate a verbalization, creating a grammatically correct and contextually appropriate natural language sentence; and c) translate the text to a target language specified by a user $lang$. This must ensure the final output aligns with the linguistic requirements of the benchmark.

For example, consider the following triple:

```
<http://dbpedia.org/resource/Churrasco>
dbo:servingTemperature "Hot" .
```

The framework generates the verbalized text in English: “The serving temperature of churrasco is hot.”. The language specified by the user is Portuguese. The text is translated into Brazilian Portuguese: “*A temperatura que um churrasco deve ser servido é quente*.” using the prompt engineering to conduct this translation.

This step’s output is a 1-to-1 mapping of triples and verbalized texts $T_{verbalized}$, ensuring that each RDF triple corresponds to a unique verbalization (line 13 of Algorithm 1). This mapping is the foundation for the subsequent steps.

3.3 Step 3: Data Curation

The third step of the framework (red rectangle in Figure 1) is dedicated to ensuring the accuracy and consistency of the text-triple pairs generated in Step 2. This step aims to identify and filter out any issues caused by hallucinations or inconsistencies introduced by the LLMs during the verbalization process.

The input for this step is the set of aligned triples and their corresponding verbalized text ($T_{verbalized}$). Each text-triple pair undergoes an evaluation process using an LLM as a data curator. The prompt to curate the data ($Prompt_{check\_cur}$) verifies two primary aspects for each pair:

1. **Semantic Alignment**: Ensures that the text reflects the information contained in the triple without introducing hallucinations or inaccuracies;
2. **Compliance with User-Defined Rules**: The framework allows users to specify natural language rules that the generated text must follow. These rules are integrated into $Prompt_{check\_cur}$ and interpreted by the language model during curation. This feature is useful for maintaining stylistic or contextual consistency across generated texts. As an illustration, consider the following example of a user-defined rule:

User-Defined Rule

You are a system that generates natural language sentences from RDF triples. Ensure that your output complies with the following rule:

Rule Name: No Personification of Companies
instructioncolorDefinition: If the subject of the triple is a company, do not use human-like verbs (e.g., “decided”, “felt”) or pronouns like “he” or “she”, unless explicitly indicated in the triple.

Example:

- **Triple:** (*Google*, *acquired*, *Fitbit*)
- **Correct:** Google acquired Fitbit.
- **Incorrect:** She decided to acquire Fitbit.

The language model evaluates the text-triple pairs based on the rules in natural languages and semantic alignment criteria (line 18 of Algorithm 1). The language model flags the pair for review if an issue is detected, such as a hallucination or a rule violation (line 21 of Algorithm 1). An example of a flagged pair is:

- Triple: *<http://dbpedia.org/resource/Pumpkin_Spice_Spam> dbo:creatorOfDish <http://dbpedia.org/resource/Hormel>*
- Text: “Mister Hormel is the dish Pumpkin Spice Spam creator.”
- Rationale: The text introduces a hallucination by personifying “Hormel”, a company. The LLM identifies

this inconsistency labeling as suspicious.

The output of this step is a curated dataset consisting of validated text-triple pairs ($T_{curated}$). Suspicious pairs flagged by the language model are excluded or returned to the user for manual review. The curated dataset represents a high-quality benchmark component free from semantic errors and rule violations.

3.4 Step 4: Data Augmentation

The fourth and final step of the framework focuses on augmenting the verbalized dataset by generating multiple text variations for each triple (yellow rectangle of Figure 1). The goal is to associate each RDF triple with a diverse set of natural language texts that express the same semantics, ensuring richness and variety in the benchmark while maintaining semantic consistency.

The input to this step is the curated list of triple-text pairs obtained from Step 3 ($T_{curated}$). The framework generates additional textual variations using an LLM for each triple in the list (line 26 of Algorithm 1). The number of additional texts is specified by the parameter m. The LLM is tailored to align with the language specified by the user.

For example, consider the triple *<http://dbpedia.org/resource/Samgye-tang> dbo:serving Temperature Hot or Warm*. The initial text generated in Step 2 might be: “The samgye-tang can be served hot or warm.”. Through this augmentation process, additional texts with equivalent semantics are generated, such as:

- “The samgye-tang has serving temperature hot or warm.”
- “One should serve samgye-tang in hot or warm temperature.”

This augmentation increases the benchmark’s linguistic variability, supporting more robust training and evaluation of natural language generation systems. An essential requirement of this step is that all augmented texts for a given triple must preserve the same semantic meaning. To this end, the language model is guided by a prompt ($Prompt_{augment}$) emphasizing semantic fidelity to ensure consistency. The augmented texts are reviewed automatically to check for potential deviations from the triple’s intended meaning. Due to the prompt approach, the quality of the results depends on the LLM used during creation; some LLMs may struggle to understand the instructions and could introduce errors.

This step outputs a comprehensive dataset D where each RDF triple is associated with a set of diverse, semantically consistent textual representations. This dataset forms the framework’s final product: a language-specific RDF-to-text benchmark that can be used to train and evaluate text generation models.

4 Evaluation Methodology

This section presents the experimental evaluation methodology designed and conducted to assess the effectiveness of our framework. Our evaluation generates and evaluates the *Portuguese RDF-Text Alignment (PoRTA)* benchmark generated by our BENCH$_4$T^3 framework. Subsection 4.1 presents details of how we applied our framework to create the *PoRTA* benchmark, in addition to other benchmarks used in our experiments. Subsection 4.2 presents the procedures conducted to evaluate the benefits of the constructed benchmark in triple text alignment through its use to build a bimodal encoder. Subsection 4.3 covers the application of our benchmark to the Natural Language Generation task to transform the triple into a natural language text.

4.1 Benchmarks

4.1.1 WEBNLG-PT

A translated Portuguese version of WEBNLG-v3 [Gardent *et al.*, 2017], a benchmark from a challenge competition to transform triples into text. This benchmark includes examples of triples and their corresponding natural English text, featuring ten classes extracted from DBpedia. We took each English sentence presented in the datasets and passed it to Google Translate to Portuguese. We only filter the one-to-many triple-text pairs, one triple alignment with texts that share the same meaning in the original benchmark, resulting in 483 unique properties.

4.1.2 Portuguese RDF-Text Alignment (PoRTA)

The PoRTA benchmark is the final output of our proposed framework, following the four steps described in Section 3. This benchmark was generated using *Llama 3.3 70B* [2], focusing on aligning RDF triples and their textual representations in the Portuguese language, addressing the gap in non-English benchmarks. To the best of our knowledge, this refers to an original RDF-to-text Portuguese benchmark to be fully available to the community[3].

In Step 1 (Triple Extraction), we selected ten classes from DBpedia inspired by the WebNLG dataset, initially targeting English. One class, *ComicsCharacter*, was excluded due to a lack of associated properties, resulting in nine final classes from DBpedia: *Airport*, *Astronaut*, *Building*, *City*, *Food*, *Monument*, *SportsTeam*, *University* and *WrittenWork*.

From these classes, we extracted 123 properties (with a minimum of 3, a maximum of 36, and an average of 13.6 per class). We generated $7,424$ RDF triples (with a minimum of 44 triples per property, a maximum of 800, and an average of 244 per property). To ensure relevance, properties were selected based on their *rdfs:domain* and *rdfs:range*. At the end of this step, $1,604$ invalid triples were identified and removed because they were deemed unverbalized by the languagem model using the applied method, resulting in $5,820$ valid triples.

In Step 2 (Triple Verbalization), each valid RDF triple was transformed into a natural language text in Portuguese, resulting in one text per triple.

In Step 3 (Data Curation), a quality control process identified 69 triple-text pairs for manual review, ensuring high

[2] `https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/`

[3] https://huggingface.co/datasets/porta-dataset/PoRTA

alignment accuracy between the triples and their verbalized representations. These pairs were excluded from the benchmark.

In Step 4 (Data Augmentation), our solution enriched the dataset by generating additional textual variations for each triple. We set the augmentation parameter $n = 5$, resulting in five unique texts per triple, in addition to the one already generated, yielding six textual variations per RDF example.

At the end of this process, the *PoRTA* benchmark comprises $5,751$ RDF triples paired with their textual representations (six per example). Table 1 presents the distributions of these benchmarks. For *PoRTA*, we initially split the benchmark by filtering out the unique properties. Then, we separated examples with unseen properties from the test partition for challenge models trained on it. The remaining samples were randomly divided into training (80%) and validation (20%) sets.

Table 1. Benchmark distribution: WEBNLG-PT, a translated version of WEBNLG-v3 benchmark, and PoRTA, our benchmark generated by our framework BENCH$_4$T^3, the number in up side represents the number of pairs (triple-text) presented in the benchmark, and the bolded number represents the number of unique triples presented into the benchmark.

	Train	**Dev**	**Test**
WEBNLG-PT	7686 (3115)	961 (402)	1418 (521)
PoRTA (Ours)	18380 (3676)	4600 (920)	5775 (1155)

Table 2 presents the mean SELF-BLEU [Zhu *et al.*, 2018] score, which measures grammar diversity, for the test partition of PoRTA and WEBNLG in their respective languages (English original benchmark and Portuguese translation).

Table 2. SELF-BLEU mean distribution for n-gram 2 and 4 in case one-to-many (triple-text) examples for test partition

	Language	**size**	**SELF BLEU-2**	**SELF BLEU-4**
PoRTA	pt	1155	0.7006	0.4829
WEBNLG	en	501	0.6323	0.3692
	pt (translated)	501	0.6608	0.4455

Figure 3 presents a violin plot illustrating the distribution of the SELF-BLEU-2 metric for the one-to-many relationship. This analysis focuses on triples whose textual representation appears in more than one natural sentence. The results indicate that the PoRTA benchmark, in the test partition, tends to produce examples with a higher lexical overlap, as shown by the concentration of SELF-BLEU-2 values in the higher ranges. This suggests that PoRTA often reuses consecutive 2-grams more frequently when converting triples into natural text, which may result in lower diversity in the generated outputs. In contrast, the WEBNLG-EN and WEBNLG-PT benchmarks exhibit a broader distribution, indicating greater variability in the text associated with a triple.

Figure 4 presents a violin plot that illustrates the distribution of the SELF-BLEU-4 metric for the one-to-many relationship, in which a single triple is expressed in multiple natural sentences. The results indicate that the PoRTA benchmark maintains a controlled level of lexical overlap, avoiding excessively high SELF-BLEU-4 values close to 1, which would signify nearly identical text generations. This suggests that PoRTA effectively balances consistency and diversity by generating paraphrased variations while preserving meaning. In contrast, the WEBNLG-EN and WEBNLG-PT benchmarks exhibit broader distributions, with some cases approaching higher SELF-BLEU-4 values. This trend may indicate a greater tendency toward repetitive phrasing.

4.2 Information Retrieval Task Procedure

Figure 5 presents the overall evaluation pipeline for assessing our generated benchmark in the triple-text alignment task. This evaluation consists of two main sub-tasks: 1) a Finetune Encoder, which is responsible for tuning a model to align text triples using benchmark training data; and 2) an Information Retrieval (IR) task, in which we assess the effectiveness of the tuned models across our two presented benchmarks. The rationale is to determine the extent to which the generated benchmark is suited to help fine-tune a bimodal encoder model that contributes to IR tasks using this embedding for retrieving texts and triples.

Fine-tune encoder: In the fine-tune encoder step, we consider a benchmark (WEBNLG-PT or *PoRTA*). We take a pretrained encoder and use the training and validation to finetune the encoder (Approach 1: Fine-tune Encoder). This generates a fine-tuned encoder to create a representation for triples and text. For the *PoRTA* benchmark, we compare two flavors of the same version: one without augmentation (Figure 1 - Step 3-only) and the other with augmentation (Figure 1 - Step 4) [full version of the framework]. We aimed to understand the role of Step 4 of our framework in the model's outcome effectiveness.

We choose the e5-multilingual [Wang *et al.*, 2024] model as our pretrained language model for the fine-tuning process. This option is based on the methodology used to develop e5, which involved curated datasets. Furthermore, its effectiveness in symmetric search [Zhang and Braun, 2024], which consists of recovering text with similar meanings, makes it suitable for our information retrieval experiment.

Retrieval System: At this stage, we assessed the effectiveness of the finetuned encoder using both benchmarks. The inputs to the retrieval system, referred to as t_i and, T_i depend on the experiment's configuration. For instance, t_i might represent textual queries, while T_i could correspond to KG triples. Conversely, t_i may denote triples, with T_i representing text. Both configurations are evaluated to assess the encoder's generalizability and effectiveness (retrieving text from triples and vice versa).

From the (t_i, T_i) pairs from the test partition, we passed the t_i to the encoder to act to the query ('Step 2 Embed data' in Figure 5) obtaining a query vector V_i, and the others $T = T_i, T_h, .., T_j$ values are separated to act as search spaces. Then, both are passed to the retrieval (Step 3 in Figure 5) and the encoder. This maps each T value into a vector representation. Afterward, the retrieval module computes the cosine similarity of (t_i, T), recovering the one with the highest similarity T_x.

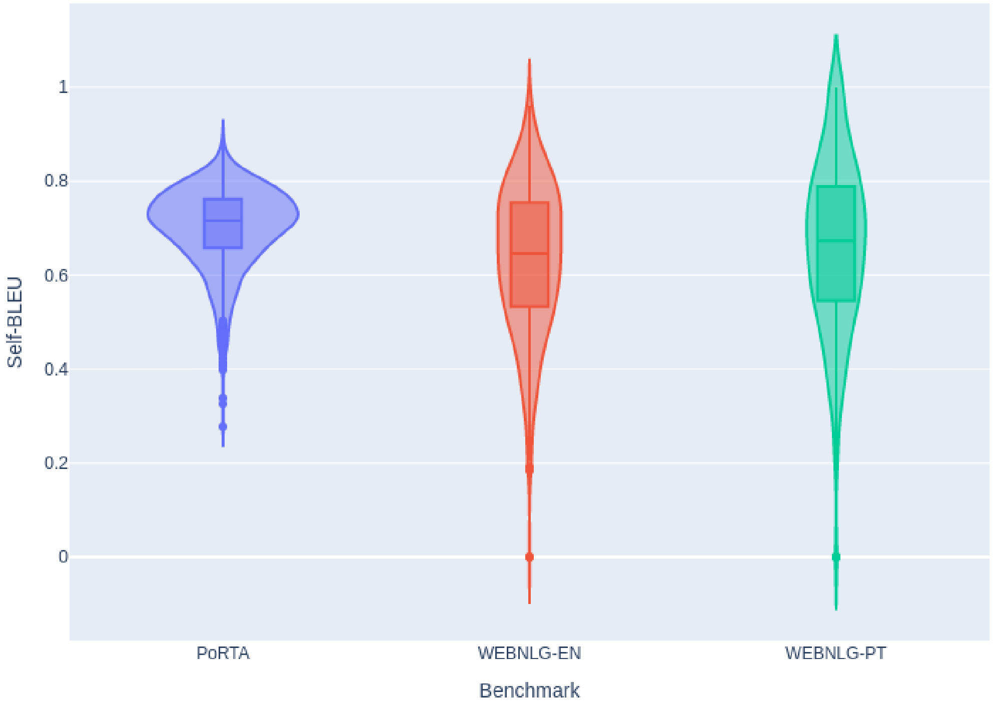


Figure 3. SELF-BLEU-2 Violin distribution for one-to-many (triple-text) examples in test partition

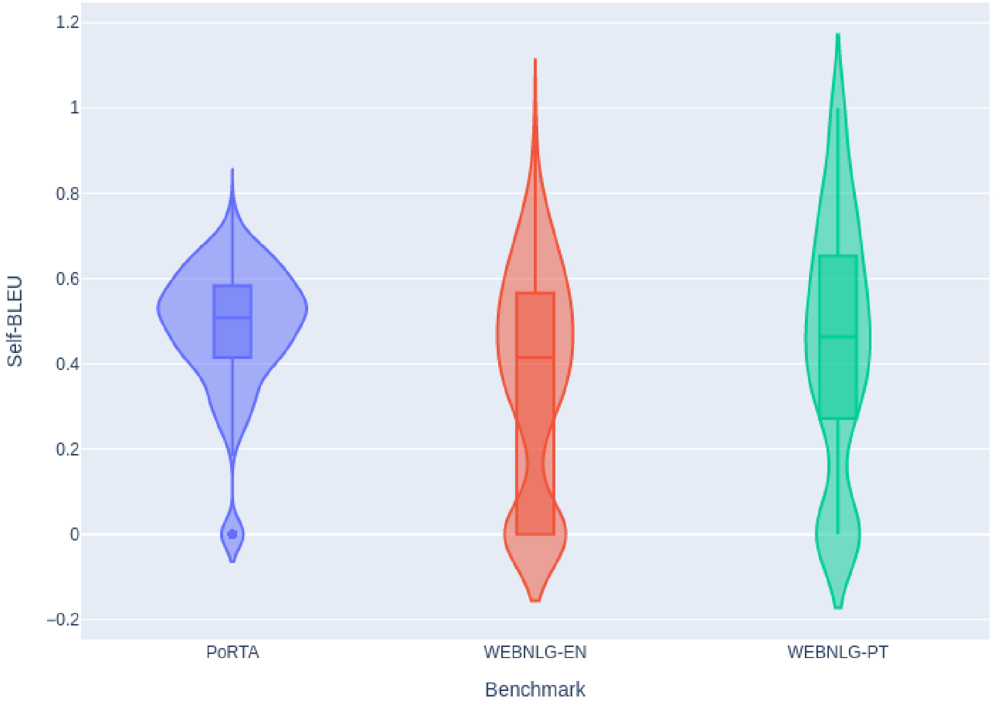


Figure 4. SELF-BLEU-4 Violin distribution for one-to-many (triple-text) examples in test partition

Metrics: To evaluate how effectively our benchmark aligns text and triple representations, we employ an Information Retrieval (IR) metric to validate the trained encoder. Specifically, we calculate the Normalized Discounted Cumulative Gain (NDCG) [Järvelin and Kekäläinen, 2002] using the test dataset from both benchmarks. Since we focus on the quality of top-ranked retrievals, we report NDCG@1, which assesses ranking effectiveness in the IR task. Subsection 5.1 details the information retrieval results.

4.3 Triple to Text Generation Task Procedure

Figure 6 presents our experimental procedure for validating the triple-to-text generation process. This evaluation involves two parallel approaches to convert triples into text: 1) fine-tuning a Sequence-to-Sequence model using the WEBNLG and PoRTA benchmarks, and 2) employing few-shot prompt techniques. Both methods serve as generators using data from the benchmark.

Once the generators are ready, we assess their effectiveness through evaluation. Triples from the test benchmark

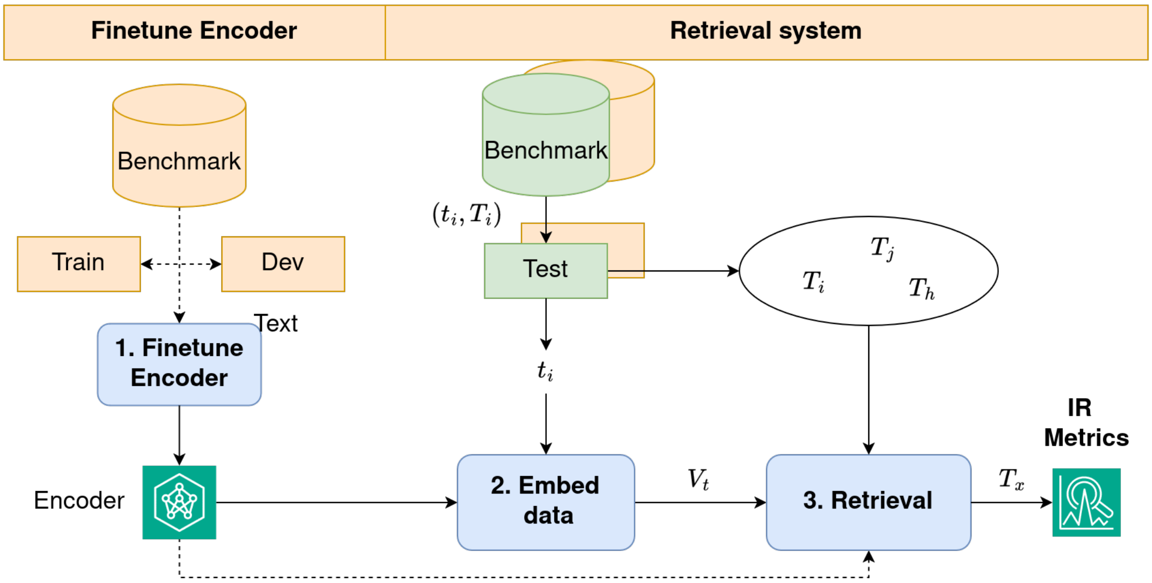


Figure 5. Evaluation pipeline to evaluate the benchmark over the information retrieval task. 1) Finetune the encoder: create an encoder using train and dev partition; 2) Embed data: transform t_i into V_i, its vector representation; 3) Retrieval: perform a retrieval task for finding the T_x to align the t_i query.

($Triples$) are input into the fine-tuned model to produce generated text (T_F). Similarly, the few-shot prompts generate text (T_{LLM}). Both generated texts are then compared with reference texts from the benchmark ($Text$), using ROUGE scores for evaluation.

4.3.1 Approach 1: Finetune Seq2Seq Generator

Algorithm 2 summarizes the first approach, which uses a pretrained Seq2Seq model to generate textual representations from structured data. The benchmark dataset $\mathcal{B}$ is initially divided into three subsets: training ($\mathcal{T}_{train}$), validation ($\mathcal{T}_{val}$), and test ($\mathcal{T}_{test}$).

During the training phase, the model is fine-tuned on the training subset $\mathcal{T}_{train}$, mapping structured triples to their corresponding textual descriptions. This fine-tuning process spans 20 epochs and includes an early stopping mechanism based on the ROUGE-1 score, which is evaluated on the validation set $\mathcal{T}_{val}$. Training stops if the model's performance does not improve according to the ROUGE-1 metric, helping to prevent overfitting.

Once fine-tuning is completed, the optimized model $\mathcal{M}^*$ is assessed on the test set $\mathcal{T}_{test}$, generating textual outputs for previously unseen triples. The final fine-tuned model is designated as the primary generator for structured-to-text transformations.

We decided to use the FLAN-T5 [?] model as our pretrained model due to its multilingual capabilities, which allow it to handle both English and Portuguese. This model demonstrated strong results in sequence-to-sequence tasks, making it appropriate for our requirement to convert original English text into Portuguese phrases. Subsection 5.2.1 presents the experimental results corresponding to this approach.

4.3.2 Approach 2: Few-Shot Generator

Algorithm 3 presents the second approach, which utilizes a language model to convert structured data into text through a few-shot prompting strategy. Rather than fine-tuning a specific model, this method selects a subset of 20 instances from the training dataset, denoted as T_{train}. Each selected instance comprises a structured triple and corresponding textual description, forming the few-shot prompt together.

Algorithm 2 Fine-tune Pretrained Seq2Seq Model (Approach 1)

```
Require: Pretrained Seq2Seq model $\mathcal{M}$, Benchmark dataset $\mathcal{B}$, Epochs = 20, Early stopping on ROUGE-1
Ensure: Fine-tuned model $\mathcal{M}^*$
Split $\mathcal{B}$ into Train ($\mathcal{T}_{train}$), Validation ($\mathcal{T}_{val}$), and Test ($\mathcal{T}_{test}$)
for epoch = 1 to 20 do
    Fine-tune $\mathcal{M}$ on $\mathcal{T}_{train}$ (triples → text)
    Evaluate $\mathcal{M}$ on $\mathcal{T}_{val}$ using ROUGE-1
    if Early stopping criteria met then
        Stop training
    end if
end for
Test $\mathcal{M}^*$ on $\mathcal{T}_{test}$ (triples)
Return fine-tuned model $\mathcal{M}^*$
```

During inference, for each test instance in T_{test}, the language model receives the prepared few-shot prompt along with the target structured triple. The model then generates the corresponding textual description based on pre-trained knowledge and the in-context learning paradigm. This approach avoids explicit model fine-tuning while allowing for efficient text generation from structured data.

In this approach, we utilized Mistral 7B [Jiang *et al.*, 2023] to evaluate the few-shot methodology. This decision was based on selecting an LLM that demonstrates strong effectiveness in state-of-the-art applications, distinguishing it from Llama 3.3, which was used to establish our PoRTA benchmark.

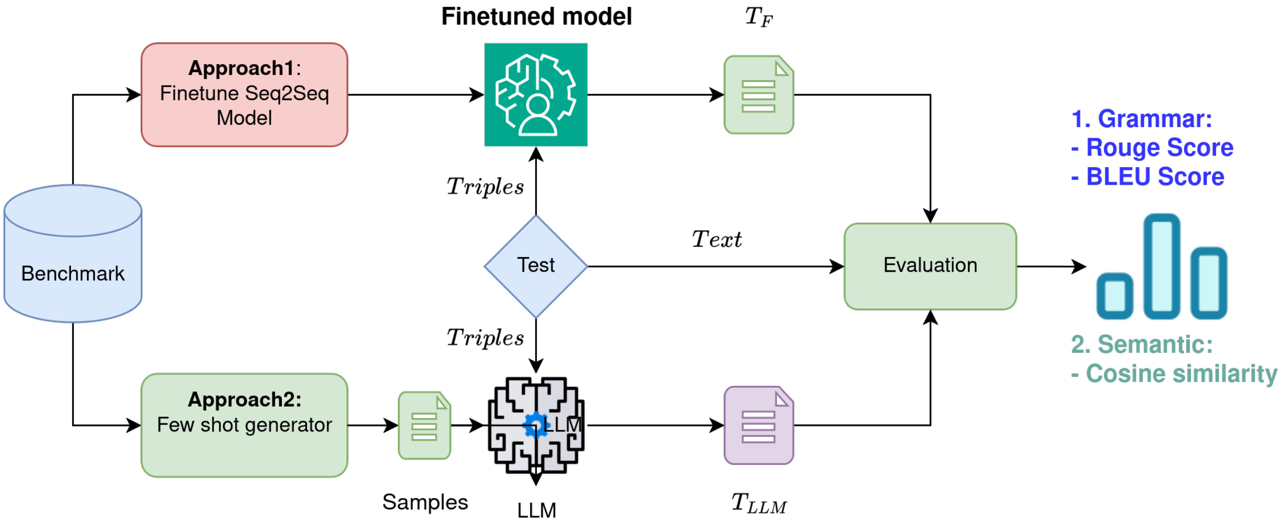


Figure 6. Evaluation procedure for triple-to-text generation. We compare two approaches: (i) fine-tuned Seq2Seq models and (ii) few-shot prompting with large language models (LLMs). The generated texts are evaluated using two categories of metrics: (1, in blue) grammar-focused metrics—ROUGE and BLEU scores; and (2, in green) semantic similarity—cosine similarity between embeddings.

Algorithm 3 Few-shot Generation with LLM (Approach 2)

```
Require: LLM $\mathcal{L}$, Benchmark dataset $\mathcal{B}$, Sample size = 30
Ensure: Generated text for $\mathcal{T}_{test}$
Sample 20 instances $(T_{fs}, Text_{fs})$ from $\mathcal{T}_{train}$
Construct few-shot prompt using $(T_{fs}, Text_{fs})$
for each triple $t$ in $\mathcal{T}_{test}$ do
    Generate $Text_t$ using $\mathcal{L}(T_{fs}, Text_{fs}, t)$
end for
Return generated text $\{Text_t\}$ for $\mathcal{T}_{test}$
```

In the following, we present examples of the structure of our prompt template. We emphasize the primary instruction in blue and the variables provided to the language model in red. The first variable specifies that the output should be in JSON format, while the examples include the 20 generated instances from the benchmark. Additionally, for each iteration, the language model receives a new triple that needs to be transformed into natural language text.

Prompt for RDF Triple Verbalization

You are an expert in natural language understanding and RDF. Your task is to generate a **natural language representation text of RDF triples**. In other words, given a triple from a **knowledge graph**, you should transform it into a fluent and natural sentence in Portuguese. Below are some examples of triples and their corresponding natural language representations: **{format_instructions}**

Examples: {examples}

With the examples above as reference, generate a natural language equivalent for the following triple:
Triple: {triple}
Text:

Subsection 5.2.2 presents the results obtained using the few-shot approach.

Metrics: To evaluate the generative capabilities of our benchmark for converting triples to text, we analyze ROUGE [Lin, 2004], and BLEU [Papineni *et al.*, 2002] scores. We specifically use ROUGE to measure the common subsequences between the generated text and the reference text, focusing on n-grams of sizes 1, 2, and L. This approach reflects recall and structural coherence. However, recall alone does not provide a complete picture of accuracy. BLEU, which is calculated using n-grams up to BLEU-2, measures precision, ensuring that the output includes expected phrasing based on the benchmark references. We used these metrics because they are widely accepted for evaluating text generation tasks. While ROUGE and BLEU assess fluency and alignment, it is important to note that they do not measure factual accuracy, which is essential for effectively verbalizing triples.

Furthermore, we use the cosine similarity as a semantic evaluator, focusing on sentence embeddings comparison based on the Multilingual-e5-base [Wang *et al.*, 2024] model. Unlike traditional n-gram-based metrics, this approach captures the semantic similarity between the generated text and the reference texts using the vectors embeddings. This is especially important in triple-to-text generation, where the meaning can be preserved even if the wording differs. We chose the Multilingual-e5-base model due to its strong performance across various languages, its open-source nature, and its effectiveness in zero-shot or multilingual scenarios. By calculating cosine similarity between sentence embeddings, this method provides a more robust measure of meaning preservation, enabling a comprehensive assessment of generation quality that goes beyond mere syntactic matching.

5 Experimental Results

This section describes our experimental results. Subsection 5.1 presents the results for the **Information Retrieval Task**, in which a bimodal encoder trained with our benchmark is challenged. Subsection 5.2 demonstrates the results of applying our benchmark in a **Generative Task** to transform triples into natural text, by reporting on the results of using the two approaches presented in Subsection 4.3. They were designed to evaluate the effectiveness of our benchmark and provide examples of the generated outputs.

5.1 Results regarding the Information Retrieval Task

Table 3 presents the results of applying the distinct encoders to IR tasks for the two benchmarks provided. We observed that mE5, without any fine-tuning, achieved acceptable values in NDCG recovering triples given a text query (Text-Triple) while showing lower values for recovering text when given a triple query (Triple-Text). The fine-tuned encoder trained on the WEBNLG-PT dataset obtained lower values, showing a considerable decrease in the NDCG score compared to models without tuning. We noticed that querying with triple remains challenging, resulting in lower values in this IR task.

The encoder trained with our benchmark before applying any augmentation (cf. Figure 1 - Step 3) demonstrated a slight improvement in the Text-Triples IR task. We further observed improved scores for Triples-Text across both benchmarks, with the WEBNLG-PT showing the most significant enhancement. Using the benchmark after augmentation (Figure 1 - Step 4) slightly improved over *PoRTA*. However, WEBNLG-PT performs marginally better for Text-Triple. We observe a minor decrease in the Triple-Text IR task.

5.2 Results regarding the Triple to Text Generation Task

We evaluate the generation of natural text using triples as input, applying the generator approaches outlined in Section 4.3. Table 4 presents the quantitative assessment results using ROUGE and BLEU metrics, comparing the effectiveness of two main configurations: fine-tuned models and few-shot prompting. Subsection 5.2.1 presents the results of the finetuning evaluation, whereas Subsection 5.2.2 presents the results of the few-shot learning evaluation. Table 5 reports on the cosine similarity scores computed using the Multilingual-e5-base encoder to assess the semantic alignment between generated texts and their reference counterparts for both WEBNLG and PoRTA. Subsection 5.2.3 presents examples of text generation for each approach and setting.

5.2.1 Results on the finetuning evaluation

For the WEBNLG dataset, the fine-tuned FLAN-T5-BASE model achieves a ROUGE-1 score of 0.6275, indicating a strong ability to capture keywords from the reference text. The ROUGE-2 score is 0.4028, reflecting a moderate outcome in generating accurate bigrams. Meanwhile, the ROUGE-L score reaches 0.5844, demonstrating adequate alignment with longer sequences of the reference outputs. The BLEU score is also 0.5306, suggesting that the model produces outputs with considerable lexical similarity to the ground truth.

On the PoRTA dataset, FLAN-T5-BASE performs slightly worse than on WEBNLG. The ROUGE-1 score is 0.5116, lower than WEBNLG's, indicating that the model has more difficulty capturing essential words. The ROUGE-2 score is 0.2731, which reflects a weaker ability to generate accurate consecutive word pairs. The ROUGE-L score is 0.4748, showing that although some more extended phrase structures are preserved, it is not as effective as with WEBNLG. Finally, the BLEU score is 0.4319, indicating that the outputs are less lexically aligned with the references than those in the WEBNLG dataset.

All grammar metrics show noticeable improvement with PoRTA without augmentation. The ROUGE-1 score increases to 0.5349, indicating that the model captures keywords slightly better. Similarly, the ROUGE-2 score rises to 0.2907, revealing a small but positive effect on generating accurate bigrams. The ROUGE-L score improves to 0.4985, meaning that the structure of longer phrases is better preserved. Additionally, the BLEU score increases to 0.4646, indicating that the generated texts are more closely aligned with the reference outputs in terms of word choice.

The semantic alignment measurement results described in Table 5 show that, when fine-tuned on the WEBNLG dataset and evaluated on the same, the model achieves a cosine similarity of 0.9704, reflecting strong semantic alignment between the generated outputs and the reference texts. While this suggests that FLAN-T5-BASE effectively captures and reproduces the meaning present in the reference examples, it may also reflect limitations of the WEBNLG benchmark itself, whose structured and relatively homogeneous nature could lead to overfitting and may not adequately challenge the model's generalization capabilities.

When evaluated on PoRTA benchmark, the FLAN-T5-BASE model shows slightly lower cosine scores. The model trained on PoRTA reaches 0.8879, while the variant trained without data augmentation slightly improves to 0.8887. These small differences suggest that data augmentation has minimal influence on semantic similarity. Additionally, the PoRTA benchmark presents a greater challenge for the model, likely due to its higher lexical and structural variability. When the model trained on PoRTA is evaluated on the WEBNLG dataset (i.e., in a cross-domain setup), the score increase to 0.9655 (compared to our benchmark). Despite being trained on the training partition of PoRTA, the finetuned model encounters challenges with the test partition. Our benchmark demonstrates more robust and stable performance, preventing unexpected score increases.

5.2.2 Results on the few-shot learning evaluation

In the few-shot learning setting, where the model is not fine-tuned, but is provided with a few examples to guide its generation, the effectiveness improves significantly across both datasets.

For WEBNLG, the ROUGE-1 score reached 0.7243, notably higher than that of the configuration fine-tuned FLAN-T5-BASE model (0.6275). This indicates that Mistral 7B is superior at capturing essential words from the reference texts. The ROUGE-2 score is 0.5279, showing significant improvement over FLAN-T5-BASE (0.4028). This indicates that the model can produce accurate bigrams. The ROUGE-L score is 0.6751, indicating improved structural alignment with the reference texts. Lastly, the BLEU score rises to 0.6136, re-

Table 3. NDCG@1 for evaluating IR system in triple-text alignment, using me5-base without tuning, a fine-tuned with WEBNLG-PT benchmark, fine-tuned with PoRTA benchmark (without augmentation) and Fine-tuned with augmentation.

	WEBNGL-PT		**PoRTA**	
	Text-Triple	**Triple-Text**	**Text-Triple**	**Triple-Text**
mE5-base	0.9193	0.8368	0.9670	0.9247
F. WEBNLG-PT	0.7946	0.5624	0.6233	0.6026
F. PoRTA	0.9424	**0.9270**	0.9870	0.9878
F. PoRTA without augmentation	**0.9616**	0.9040	**0.9904**	**0.9913**

flecting a more substantial lexical match with the reference outputs.

In the case of PoRTA, the few-shot Mistral 7B model outperformed the fine-tuned FLAN-T5-BASE model. The ROUGE-1 score is 0.7072, representing a substantial increase over the baseline FLAN-T5-BASE (0.5116) and the augmented version (0.5349). The ROUGE-2 score reached 0.5107, significantly higher than FLAN-T5-BASE (0.2731) and FLAN-T5-BASE without augmentation (0.2907), indicating notable improvement in generating accurate word sequences. The ROUGE-L score is 0.6656, showing that Mistral 7B effectively maintains sentence structures. Finally, the BLEU score reached 0.6634, indicating the best alignment with the reference texts among all tested models.

In the few-shot setting, the semantic alignment was evaluated using Mistral 7B, comparing it with the results from both the WEBNLG and PoRTA datasets (cf. Table 5). When prompted with WEBNLG examples, the model achieved a cosine similarity of 0.9813 on WEBNLG and 0.8917 on PoRTA. These results indicate that while the model performs exceptionally well within its domain, it experiences a moderate decline in semantic alignment when applied to the more challenging PoRTA benchmark. In the reverse configuration, using PoRTA as the few-shot input, the cosine scores are 0.9809 on WEBNLG and 0.8922 on PoRTA. Despite the inherent difficulty of PoRTA, the model maintains high semantic fidelity in both target datasets. These results suggest that PoRTA introduces greater variability and complexity, which slightly impacts the model's ability to maintain semantic precision compared to WEBNLG.

5.2.3 Analysis on Examples of generation over WEBNLG and PoRTA

We present a qualitative analysis based on examples generated over WEBNLG and PoRTA. Table 6 and Table 7 present examples of different models generating text from RDF triples using the WebNLG and PoRTA.

Table 6 examines a triple from WEBNLG "*Olga Bondareva | deathDate | 1991-12-01*" — which contains an incorrect date. The reference texts in Portuguese and English state that 'Olga Bondareva died on December 9, 1991', contradicting the triple. Despite this inconsistency, most models, especially Mistral-webnlg and Mistral-PoRTA, generated text faithful to the triple, stating that 'she died on December 1, 1991'. Some models introduced errors, such as Finetuned-PoRTA without augmentation, which produced an utterly wrong output. Spelling errors appeared, with some models generating 'Bondeva' instead of 'Bondareva'.

Table 7 evaluates a triple from the PoRTA benchmark "Princess Princess_(manga) | magazine | Wings_(Japanese_magazine)" and compares model outputs with reference texts. The expected output should indicate that 'Princess Princess was published in Wings magazine'. However, several models introduced misinterpretations. Finetuned-webnlg incorrectly described 'Princess Princess as a company', while Mistral-webnlg mistakenly referred to Wings as a newspaper. Although Mistral-PoRTA generated a reasonable output, it slightly altered the original phrasing. Some fine-tuned PoRTA models struggled with Portuguese grammar, incorrectly using articles and nouns. Additionally, Finetuned-PoRTA without augmentation produced an incomplete and incoherent output, suggesting instability in its text generation.

6 Discussion

The mE5-base pretrained encoder achieved satisfactory results in both benchmarks without any fine-tuning. In the case of Text-Triple, the higher values may be related to the similarity in grammatical structure between the text and the triple elements. The Triple-Text benchmark produced lower values, which can be attributed to the pretrained encoders not being fine-tuned to effectively represent triples in a vector space. The IR task is more challenging in WEBNLG-PT compared to our generated benchmark, possibly because WEBNLG-PT includes unseen entities in the test partition, making the retrieval task harder.

Finetuning the encoder using the WEBNLG-PT benchmark has decreased the NDCG@1 score. This may suggest that their examples are unsuitable for creating an intermediate vector space. They were designed to convert triples to text and may not work well when fine-tuning the encoder. The translation might produce incorrect terms that are unsuitable for correct alignment.

Using our generated benchmark (*PoRTA* – without augmentation – Figure 1 step 3) to fine-tune an encoder does not decrease the NDCG score for either benchmark. This suggests that our benchmark effectively facilitates learning well-aligned relationships between text and triples. The NDCG metrics improved across both IR tasks. This indicates that we achieved a better encoder to represent the triples and recover the correctly aligned text effectively.

Our last encoder utilized augmentation examples. Although this approach did not yield a significant improvement compared to the encoder that did not use augmentation, this benchmark could negatively impact training compared to the WEBNLG benchmark application. This suggests that our

Table 4. Quantitative results for computing ROUGE-1, ROUGE-2, ROUGE-L, and BLEU were obtained for the WEBNLG and PoRTA benchmarks, focusing on the main configurations of fine-tuned models (FLAN-T5) and few-shot prompts with Mistral 7B.

		WEBNLG			**PoRTA**			**WEBNLG**	**PoRTA**
	Data	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**BLEU**	**BLEU**
FLANT5-BASE	webnlg	0.6275	0.4028	0.5844	0.6031	0.3825	0.5547	0.5306	0.4319
FLANT5-BASE	PoRTA	0.5116	0.2731	0.4748	0.6000	0.4076	0.5680	0.3891	0.4471
FLANT5-BASE	PoRTA without augmentation	0.5349	0.2907	0.4985	0.6215	0.4498	0.5983	0.4125	0.4646
Few shot mistral	webnlg	0.7243	0.5279	0.6751	0.7208	0.4834	0.6775	0.6136	0.6308
few shot mistral	PoRTA	0.7072	0.5107	0.6656	0.7436	0.5142	0.7047	0.5881	0.6634

Table 5. Cosine Similarity Evaluation with Multilingual-e5-base for WEBNLG and PoRTA Using Fine-tuned and Few-shot Configurations

		WEBNLG	**PoRTA**
	Data	**Cosine**	**Cosine**
FLANT5-BASE	webnlg	0.9704	0.8878
FLANT5-BASE	PoRTA	0.9655	0.8879
FLANT5-BASE	PoRTA without augmentation	0.9665	0.8887
Few shot mistral	webnlg	0.9813	0.8917
few shot mistral	PoRTA	0.9809	0.8922

augmentation examples did not damage the fine-tuning process. The task for recovering text given a triple query (triple-text) showed a slight decrease in effectiveness (NDCG@1). This may be because our data augmentation focused on text rather than triples, which may affect the representations of the triples.

In the fine-tuned setup, WEBNLG scored consistently outperforming PoRTA scores across all metrics. This indicates that the FLAN-T5 model is better suited for the WEBNLG dataset, likely due to the structured nature of WEBNLG training data. Nevertheless, PoRTA still demonstrated competitive effectiveness, mainly when augmentation techniques are not applied (PoRTA without augmentation), as these techniques enhanced scores across all evaluation metrics. Notably, the BLEU score for PoRTA achieved 0.4646, suggesting improved lexical alignment with the reference outputs.

The observed improvements in PoRTA, when not using augmentation, suggest that augmentation may not reflect an improvement over grammar metrics. The generated examples (cf. Table 6 and Table 7) demonstrate that, despite sharing a similar grammar, the examples generated by PoRTA with augmentation have a closer meaning to the references.

From the perspective of cosine similarity, the FLAN-T5-BASE model achieved strong semantic alignment on WEBNLG (0.9704), but notably lower values on PoRTA (0.8878). This reinforces the idea that FLAN-T5 is better adapted to structured datasets, such as WEBNLG, where surface-level patterns are more predictable and semantically aligned with reference texts. When fine-tuned directly on PoRTA, the effectiveness remained comparable (0.8879), while removing augmentation slightly improved the semantic similarity to 0.8887. These results suggest that PoRTA is more challenging in terms of lexical overlap, as reflected by lower BLEU and ROUGE scores. However, fine-tuned models still achieve substantial semantic similarity, as indicated by high cosine scores, showing that meaning is preserved even when surface forms diverge. This suggests that traditional n-gram metrics may underestimate performance on PoRTA, and that embedding-based metrics offer a more accurate representation of semantic fidelity in this benchmark.

In the few-shot setting, Mistral 7B outperformed FLAN-T5 across both datasets. The most significant gains were observed in PoRTA, where ROUGE-1 improved from 0.5349 (FLAN-T5 without augmentation) to 0.7072 (Mistral 7B), and BLEU jumped from 0.4646 to 0.6634. This suggests that a strong general-purpose LLM, such as Mistral 7B, can generate high-quality outputs for PoRTA from few examples, potentially reducing the need for extensive labeled training data. Another key highlight is that Mistral's improvement over FLAN-T5 is more pronounced in PoRTA than in WEBNLG. In the few-shot setting, Mistral 7B outperformed FLAN-T5 across both datasets. The most significant improvements are observed in PoRTA, where the ROUGE-1 score rises from 0.5349 (for FLAN-T5 with weights averaging) to 0.7072 (for Mistral 7B), and the BLEU score increased from 0.4646 to 0.6634. This demonstrates that Mistral 7B, as a strong general-purpose language model, can generate high-quality outputs relying on PoRTA, potentially decreasing the reliance on extensive labeled training data.

While fine-tuning FLAN-T5 demonstrated solid outcomes, PoRTA greatly benefited from few-shot learning with Mistral 7B. This makes Mistral 7B a strong option for scenarios with limited fine-tuning resources. This suggests that PoRTA benefits significantly from a few-shot learning approach, possibly because of its diverse or less structured nature. The ability of Mistral 7B to generalize from a few examples contributes to this enhanced performance.

In the few-shot case, cosine similarity scores confirmed Mistral 7B's superior semantic results. When prompted with WEBNLG examples, Mistral achieved 0.9813 on WEBNLG and 0.8917 on PoRTA. Conversely, using PoRTA as the few-shot input yields 0.9809 on WEBNLG and 0.8922 on PoRTA. These results highlight that, despite PoRTA's more diverse (lower grammar, e.g., ROUGE score), it still supports high-quality, meaning-preserving generation. The minimal drop in similarity when using PoRTA as input further demonstrates its potential as a reliable and challenging benchmark for evaluating semantic generalization in few-shot scenarios.

BENCH$_4$T^3 generated a comprehensive benchmark for Portuguese, providing more examples than currently available and achieving strong NDCG results in the IR task (addressing RQ1). We found that the fine-tuned encoder models with *PoRTA* outperformed the results of models trained with the WEBNLG-PT benchmark in configuration triples-text and text-triples, constructing a robust bimodal encoder for triple-text data (answering question RQ2).

Using our benchmark effectively transforms triples into text for the Portuguese language through a fine-tuning process and

Table 6. Generative examples for converting triples to text from WEBNLG. This example represents a triple that contains inaccurate information in the webnlg benchmark and the correct information in the text pair. The bolded entries correspond to models with data augmentation.

Triple	***Olga_Bondareva | deathDate | 1991-12-01***
Original	*Olga Bondareva morreu em 9 de dezembro de 1991* *Olga Bondareva died on December 9, 1991. (From the original dataset)*
Finetuned-webnlg	Olga Bondeva morreu em 01/12/1991
Finetuned-PoRTA without augmentation	Olga Bondeva foi muito feito em 1 de julho de 1991
Finetuned-PoRTA	Olga Bondareva foi muito em 1 de dezembro de 1991
Mistral-webnlg	Olga Bondareva faleceu em 1º de dezembro de 1991
Mistral-PoRTA	Olga Bondareva morreu em 1º de Dezembro de 1991

Table 7. Generative examples for converting triples to text from PoRTA benchmark, The bolded entries correspond to models with data augmentation.

Triple	**Princess_Princess_(manga) | magazine | Wings_(Japanese_magazine)**
Original	*A revista Princess Princess foi publicada na revista Wings.* *Princess Princess magazine was published in Wings magazine. (Translated by authors)*
Finetuned-webnlg	Princess Princess é uma empresa manga em Wings.
Finetuned-PoRTA without augmentation	Princess Princess é um manga que é a revista Wings (magazine ja
Finetuned-PoRTA	A manga Princess Princess é do revista Wings (japanesa)
Mistral-webnlg	A mangá Princess Princess é publicada no jornal Wings, do Japão.
Mistral-PoRTA	O mangá Princess Princess apareceu na revista Wings (publicação japonesa).

a few-shot prompting approach, creating coherent text that is competitive with the existing benchmark (addressing RQ3). We identified that few-shot approaches outperformed the results obtained by fine-tuned models, creating more coherent text and achieving better scores in grammar-sharing metrics (addressing RQ4).

Concerning the ROUGE and BLEU metrics, the data augmentations used in our benchmark generations do not appear to have a positive impact. However, upon reviewing the example generations, we demonstrated that the augmentation techniques produce text that better aligns with the existing reference benchmarks, having a positive effect (addressing RQ5).

This study provided valuable lessons, but it has limitations. The human selection of DBpedia classes for non-expert users may lead to bias. LLMs' simultaneous text generation and translation tasks can introduce bias. LLMs may also exhibit biases related to gender, race, and culture that need to be addressed, but remain out of the scope of the present study.

Our novel BENCH$_4$T3 framework offers significant advancements in unified KGs and LLMs. These innovations address key challenges in linking KG data with natural text, without relying on intermediate steps like GRAPH-RAG that require converting text into KGs. Our findings improved the alignment of Triples-Text, highlighting advancements in KG and LLM integration. This enables the sharing of vector space might enhance LLM answers by recovering knowledge from text and KG data.

Beyond its contribution to triple-text alignment and generation, our benchmark has the potential to support a range of tasks that are central to natural language processing research. The structured design of PoRTA and the flexibility of the BENCH$_4$T^3 framework enable its application in data-to-text generation, relation extraction, and knowledge graph construction —key areas of interest in current NLP studies. By providing aligned pairs of structured triples and natural language text, the benchmark facilitates the development and evaluation of models that can bridge symbolic and sub-symbolic representations. These capabilities are increasingly relevant in applications that require models to understand, reason over, or verbalize structured knowledge, such as question answering, summarization with factual grounding, and multilingual knowledge transfer. In this sense, BENCH$_4$T^3 serves as both a benchmark generator and an enabler of broader NLP research directions involving structured data.

7 Conclusion

Generating triple-text benchmarks demands novel solutions to facilitate their construction for distinct domains and underrepresented languages. Existing approaches have primarily focused on deriving new benchmarks from existing ones. This study advanced the state-of-the-art by automating the process of creating benchmarks for specific target languages. We demonstrated that an automatically generated benchmark enhances the effectiveness of bimodal encoders compared to those trained with traditional existing benchmarks translated to the focused language – Portuguese. Our proposed benchmark enhanced the representation of triples, yielding a suitable vector representation for effectively recovering well-aligned texts. Our findings open new avenues for constructing hybrid IR systems by fostering better integration between KGs and natural texts within a shared semantic space. We further demonstrated the suitability for evaluating NLG for triple texts, providing a robust framework for future research in semantic generation and interpretation. Future investigations involve exploring additional KG elements to enrich triple inputs. This may introduce additional diversity in the generated benchmarks, allowing for even higher-quality benchmarks.

Declarations

Authors' Contributions

VJSC is the main contributor of this manuscript. VJSC contributed with the conceptualization, methodology, software, validation, writing. AGR contributed with the conceptualization, methodology, validation, writing. JCR contributed with the funding acquisition, project administration, supervision, writing, revision. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

We acknowledge GoBots for providing the infrastructure required to run the experiments described in this article.

Funding

This study was financed by the National Council for Scientific and Technological Development - Brazil (CNPq) process number 140213/2021-0. In addition, this research was partially funded by the São Paulo Research Foundation (FAPESP) (grants #2022/13694-0, #2022/15816-5 and #2024/07716-6). Additionally, this study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The opinions expressed in this work do not necessarily reflect those of the funding agencies. We thank GoBots for providing the infrastructure used in this research.

Availability of data and materials

The datasets (and/or softwares) generated and/or analysed during the current study are available in link.

References

Agarwal, O., Ge, H., Shakeri, S., and Al-Rfou, R. (2021). Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online. Association for Computational Linguistics. DOI: 10.18653/v1/2021.naacl-main.278.

Chico, V. J. S. and dos Reis, J. C. (2024). 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*, pages 51–62. INSTICC, SciTePress. DOI: 10.5220/0013015100003838.

Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., and Li, Q. (2024). A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6491–6501, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3637528.3671470.

Ferreira, T. C., Gardent, C., Ilinykh, N., Van Der Lee, C., Mille, S., Moussallem, D., and Shimorina, A. (2020). The 2020 bilingual, bi-directional webnlg+ shared task overview and evaluation results (webnlg+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*. DOI: 10.5281/zenodo.6552785.

Gardent, C., Shimorina, A., Narayan, S., and Perez-Beltrachini, L. (2017). Creating training corpora for NLG micro-planners. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics. DOI: 10.18653/v1/P17-1017.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., and Liu, T. (2024). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.* Just Accepted. DOI: 10.1145/3703155.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446. DOI: 10.1145/582415.582418.

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Yu, P. S. (2022). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514. DOI: 10.1109/TNNLS.2021.3070843.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. (2023). Mistral 7b. DOI: 10.48550/arxiv.2310.06825.

Lieb, A. and Goel, T. (2024). Student interaction with newtbot: An llm-as-tutor chatbot for secondary physics education. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '24, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3613905.3647957.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics. Available at: `https://aclanthology.org/W04-1013/`.

Mihindukulasooriya, N., Tiwari, S., Enguix, C. F., and Lata, K. (2023). Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text. In Payne, T. R., Presutti, V., Qi, G., Poveda-Villalón, M., Stoilos, G., Hollink, L., Kaoudi, Z., Cheng, G., and Li, J., editors, *The Semantic Web – ISWC 2023*, pages 247–265, Cham.

Springer Nature Switzerland. DOI: 10.1007/978-3-031-47243-5$_1$4.

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., and Roth, D. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*, 56(2). DOI: 10.1145/3605943.

Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., and Wu, X. (2024). Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599. DOI: 10.1109/TKDE.2024.3352100.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In Isabelle, P., Charniak, E., and Lin, D., editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics. DOI: 10.3115/1073083.1073135.

Pérez, J., Arenas, M., and Gutierrez, C. (2009). Semantics and complexity of sparql. *ACM Trans. Database Syst.*, 34(3). DOI: 10.1145/1567274.1567278.

Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., and Ting, D. S. W. (2023). Large language models in medicine. *Nature Medicine*, 29(8):1930–1940. DOI: 10.1038/s41591-023-02448-8.

Wang, H. and Na, T. (2024). Rethinking e-commerce search. *SIGIR Forum*, 57(2). DOI: 10.1145/3642979.3643007.

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., and Wei, F. (2024). Multilingual e5 text embeddings: A technical report. DOI: 10.48550/arxiv.2402.05672.

Zhang, L. and Braun, D. (2024). Twente-BMS-NLP at PerspectiveArg 2024: Combining bi-encoder and cross-encoder for argument retrieval. In Ajjour, Y., Bar-Haim, R., El Baff, R., Liu, Z., and Skitalinskaya, G., editors, *Proceedings of the 11th Workshop on Argument Mining (ArgMining 2024)*, pages 164–168, Bangkok, Thailand. Association for Computational Linguistics. DOI: 10.18653/v1/2024.argmining-1.17.

Zhu, Y., Lu, S., Zheng, L., Guo, J., Zhang, W., Wang, J., and Yu, Y. (2018). Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1097–1100, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3209978.3210080.