Adapting WebNLG to DBpedia Evolution for Evaluating LLM-based Triple Extraction

Apostolos Baniotis[1,2], Katerina Papantoniou[1,2], Michalis Mountantonakis[1,2], Nikos Kontonasios[1,2], Iordanis Sapidis[1,2], Pavlos Fafalios[1,3], Valantis Zervos[1,2], Yannis Tzitzikas[1,2], Ruofei Lai[4], Amela Fejza[4], Shriram Piramanayagam[4], Pavlos Vougiouklis[4] and Jeff Z. Pan[4]

[1]*Information Systems Laboratory (ISL), Institute of Computer Science (ICS), Foundation for Research and Technology – Hellas (FORTH), Heraklion, Greece*

[2]*Computer Science Department, University of Crete, Heraklion, Greece*

[3]*School of Production Engineering and Management, Technical University of Crete, Chania, Greece*

[4]*Huawei Technologies R&D (UK) Ltd., Edinburgh, United Kingdom*

Abstract

WebNLG is a widely used benchmark for many natural language generation tasks, including triple extraction. However, its current version includes triples from previous versions of DBpedia. This can result in several unexpected cases, such as alignment of text to outdated entities and properties' names. In addition, in the era of Large Language Models (LLMs), this can result in confusing situations where LLMs provide correctly aligned relations and entities that differ from the gold standard, raising important concerns about the reliability of evaluation results. For this reason, in this resource paper, we provide an up-to-date version of the test set of the WebNLG dataset. The dataset includes 1,779 texts and 5,639 triples, and can be used for several use cases, such as text-to-triples generation where the updated schema ensures accurate property mapping and ontology-aligned triple outputs. Finally, we provide experimental results using one-shot prompting on various LLMs (Qwen:3:32B, Mistral-small-3.1:24B, GPT-5) and state-of-the-art pretrained transformer-based encoder only models.

Keywords

Triple Extraction, Benchmark, WebNLG, Large Language Models, Knowledge Graph

1. Introduction

Knowledge Graph (KG) construction from natural language text has become a trendy research topic, with applications ranging from semantic search to question answering (QA) [1, 2, 3]. A crucial factor in this domain has been the availability of high quality benchmarks, that align texts with KG triples. Among them, WebNLG [4] has become one of the most widely used benchmarks. It has seen broad adoption in multiple NLP tasks, predominantly those involving natural language generation and triple extraction. Notably, since its first release in 2017, this benchmark has been cited by over 500 works. In addition, over the years, it has undergone many modifications and extensions [5, 6, 7] amplifying the interest of the research community and the practical usability of this benchmark.

However, because the WebNLG benchmark relies on older versions of DBpedia, a fundamental challenge arises: the alignment between the text, the ontology, and current world knowledge is no longer reliable. In the era of Large Language Models (LLMs), evaluating against an outdated ground truth conflates a model's extraction capabilities with its parametric factual knowledge. This creates

Graph-enhanced LLMs for trustwOrthy Web data management (GLOW), in ISWC 2026 Workshops Joint Proceedings, October 25–26, 2026, Bari, Italy

baniotis@ics.forth.gr (A. Baniotis); papanton@ics.forth.gr (K. Papantoniou); mountant@ics.forth.gr (M. Mountantonakis); nkonto@ics.forth.gr (N. Kontonasios); sapidis@ics.forth.gr (I. Sapidis); fafalios@ics.forth.gr (P. Fafalios); vzervos@ics.forth.gr (V. Zervos); tzitzik@ics.forth.gr (Y. Tzitzikas); lairuofei@huawei.com (R. Lai); amela.fejza@huawei.com (A. Fejza); shriram.piramanayagam@h-partners.com (S. Piramanayagam); pavlos.vougiouklis@huawei.com (P. Vougiouklis); jeff.pan@huawei.com (J. Z. Pan)

0009-0006-3122-1844 (A. Baniotis); 0000-0003-1731-2951 (K. Papantoniou); 0000-0002-1951-0241 (M. Mountantonakis); 0009-0003-5929-0305 (N. Kontonasios); 0009-0005-1367-7015 (I. Sapidis); 0000-0003-2788-526X (P. Fafalios); 0009-0002-9553-1535 (V. Zervos); 0000-0001-8847-2130 (Y. Tzitzikas); 0009-0007-0615-8326 (R. Lai); 0009-0008-2785-4424 (A. Fejza); 0009-0000-0193-564X (S. Piramanayagam); 0000-0002-1781-6319 (P. Vougiouklis); 0000-0002-9779-2088 (J. Z. Pan)

© 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 2026-11-05

CEUR-WS.org/Vol-4313/glow-paper8.pdf

a severe evaluation bottleneck driven by two distinct cases: (a) *Administrative Ontology Drift*, i.e., the majority of changes in evolving KGs are administrative (e.g., predicate renamings like `city` to `locationCity`, or URI redirects). In these cases, the underlying meaning remains unchanged. If an LLM canonicalizes a predicate to the modern DBpedia schema, it remains entirely faithful to the source text but is heavily penalized by the outdated gold standard. Also, (b) *Temporal Factual Changes*, i.e., real-world facts change over time. If a system is tasked with strictly *text-faithful triple extraction* (i.e., extracting exactly what is stated in the input text), evaluating it on an outdated benchmark can affect its performance. Either the model propagates stale facts, or it uses its parametric memory to output an updated fact, which, while true in the real world, is an "intrinsic hallucination" relative to the legacy text.

To illustrate this problem, consider the extraction of facts regarding the Adolfo Suárez Madrid-Barajas Airport. Under the original WebNLG benchmark, the gold-standard triple relies on an outdated property and entity: `(Adolfo_Suárez_Madrid-Barajas_Airport | operatingOrganisation | ENAIRE)`. A modern LLM is likely to generate a triple reflecting the current DBpedia schema: `(Adolfo_Suárez_Madrid-Barajas_Airport | operator | Aena)`. Using the older benchmark, the LLM is penalized, making it difficult to determine whether the model failed at text-faithful extraction, successfully adapted to a modern schema, or hallucinated due to parametric biases. Furthermore, fixing the evaluation to a deprecated DBpedia version is impractical, as outdated URIs no longer resolve against the live endpoint, breaking entity linking, SPARQL validation, and downstream KG population.

To tackle the above issues, in this paper we present WebNLG++, an updated, version-aligned companion to the WebNLG+ test set[1] [8]. WebNLG++ is not intended to replace WebNLG+, but rather to serve as a complementary diagnostic tool. It enables the rigorous evaluation of general-purpose, LLM-based approaches without fine-tuning, containing 1,779 texts and 5,639 triples aligned with DBpedia (version November 2025). The WebNLG++ dataset is publicly available at the provided link[2].

Therefore, our work provides an ontology-aligned update of the WebNLG benchmark ensuring that the accompanying alignments are consistent with the latest DBpedia version. By explicitly categorizing administrative drift versus temporal factual changes (through three use cases), WebNLG++ enables a fairer and more reliable assessment of modern text-to-triple systems. It also contributes to the community by supporting research on ontology evolution and long-term benchmark maintenance. Moreover, our experiments employing both LLM-based and encoder-only transformer-based systems illustrate the advantages of ontology-consistent evaluation and establish reference results to facilitate future research.

Finally, because the primary purpose of a benchmark is reliable evaluation, WebNLG++ focuses on updating the *test* split. This design choice is also motivated by the growing adoption of training-free and prompting-based triple extraction methods, where models are evaluated directly without fine-tuning. Updating the considerably larger training split would require substantially greater maintenance effort as DBpedia continues to evolve. Instead, we provide a semi-automated update pipeline and maintenance guidelines that can also be applied to the training split by researchers wishing to retrain their models.

The rest of this paper is organized as follows: §2 reviews related work. §3 introduces the dataset, including its creation process, its history, and the WebNLG++ modifications. §4 describes our evaluation methodology, with a focus on same-as property evaluation. §5 presents experiments using WebNLG++, including evaluations of LLMs in real-world settings, experiments using the GPT-5 API, triple extraction tasks, and comparisons with encoder only transformer based models. Finally, §6 concludes the paper.

2. Related Work

Models using WebNLG. Several models, including autoencoding and other architectures have been tailored for the task of *triple extraction*, have used WebNLG for their evaluation. Models such as TDEER (BERT-based) [9], UniRel [10], and OD-RTE [11] achieve high F1-scores on WebNLG, indicating their

[1] https://gitlab.com/shimorina/webnlg-dataset/-/tree/master/release_v3.0
[2] https://github.com/baniotis/webnlg_plus_plus_dataset/tree/main

robustness and generalizability across the involved domains. Specifically, UniRel and OD-RTE show high performance on the WebNLG dataset, with F1-scores more than 94%, due to their unified treatment of entity and relation extraction tasks. Moreover, models like SPN [12], OneRel [13] have exhibited high performance. They benefit from rich contextual embeddings and relation-aware architectures. Earlier architectures, such as TPLinker [14] and TDEER (LSTM-based) [9] show lower results, and reflect their efficiency in handling simple sentence structures and smaller datasets. WebNLG has also been used for the evaluation of an LLM-based approach. [15] presents a pipeline that contextual information is dynamically gathered from a KG to enhance LLM-based triple extraction applying both zero and few-shot demonstrations. Indicatively, by using LLaMA-65b [16] the F1 score for the WebNLG was 0.677 instead of 0.377 for the zero-shot mechanism.

Related Benchmarks. In addition to the WebNLG dataset, several benchmarks have been proposed for evaluating the task of triple extraction. An extensively used benchmark is NYT [17]. NYT is a distantly-supervised dataset that automatically aligns Freebase with New York Times news articles. The training set contains about 56k sentences, while the test and the validation sets contain about 5k sentences each. TAC Relation Extraction (TACRED) dataset [18] was developed by the Stanford NLP Group and is a fully supervised large-scale training dataset that consists of about 106k examples and 41 relation types built on English newswire and web text. Based on TACRED, TACRED-Revisited [19] and MultiTACRED [20] followed. FewRel [21] is a large-scale dataset for few-shot relation extraction. It consists of 70k natural language sentences from Wikipedia expressing 100 different relations of Wikidata. DocRED [22] is a document-level relation extraction from plain text dataset constructed from Wikipedia and Wikidata. It contains about 132k entities and 56k relational facts annotated on 5,053 Wikipedia documents and it provides large scale distantly supervised data over 101,873 documents. RED-FM[23] is a filtered and humanly-revised dataset that covers 32 relation types, for about 15k texts in 7 languages. A large-scale alignment between triples (Wikidata triples) and texts (Wikipedia abstracts) was introduced by T-REx, however, it still relies on distant supervision providing limited quality guarantees about the accuracy of the alignment between textual content and triples [24].

More recently, the focus has expanded toward *ontology-driven triple extraction from text*, where the extraction process is strictly guided by a provided TBox (e.g., classes, properties, domain/range constraints) as input. A foundational benchmark for this task is Text2KGBench [25], which evaluates the capability of LLMs to generate knowledge graphs that conform to a predefined ontology while remaining faithful to the input text. Building upon this, the authors in [26] introduced Text2KGBench-LettrIA, which refines the DBpedia-WebNLG subset of Text2KGBench by enforcing strict hierarchical structures, formal typing, and correcting annotation errors to improve overall ontological compliance. While these works focus on prompting models with a static schema to guide extraction, our work addresses the complementary challenge of evaluating standard triple extraction models against a KG whose underlying schema is actively evolving.

Benchmarks and outdated information. We now review the work on creating reliable benchmarks that address challenges that arise from the evolution of data and knowledge. In [27] the authors check how public tabular data repositories are used and show that many benchmark results are unreliable due to poor validation, missing baselines, and data leaks. They propose standardized task definitions and stronger evaluation protocols to make tabular ML benchmarking more reproducible and trustworthy. The authors in [28] introduce the HoH benchmark, which tests how Retrieval Augmented Generation (RAG) systems handle outdated information. They pair current and obsolete facts from sequential Wikipedia versions. The experimental results show that even top models generate false answers with confidence due to outdated content.

Furthermore, the rapid evolution of KGs introduces significant challenges for benchmark maintenance, as the underlying ontologies continuously shift. To address ontology versioning, tools such as DBpedia Archivo [29] have been developed to automatically discover, crawl, and archive ontologies. Archivo highlights the critical need for stable, referenceable identifiers so that applications and research can rely on a specific version of an ontology without unexpected breakages. While DBpedia Archivo provides the necessary infrastructure for tracking these structural changes, our work practically applies this concept to benchmark maintenance, demonstrating how the natural evolution of DBpedia necessitates

the continuous update of ground-truth alignments to prevent the penalization of models that correctly output up-to-date facts.

3. The WebNLG++ Benchmark

3.1. Background and Motivation

WebNLG+ [8] is a dataset of parallel RDF triple sets from DBPedia and short texts. The dataset covers about 450 different DBpedia properties. It is a freely available dataset distributed under the CC BY-NC-SA 4.0 licence and it has been extensively used, among other tasks, for the task of triple extraction. The corpus contains sets consisting of up to seven triplets each, along with one or more reference texts for each set. The test set is split into two parts: seen, containing inputs created for entities and relations belonging to sixteen DBpedia categories (i.e., Airport, Astronaut, Building, City, ComicsCharacter, Food, Monument, SportsTeam, University, WrittenWork, Company, Athlete, Artist, CelestialBody, MeanOfTransportation, and Politician) that are observed in the training data, and unseen, containing inputs extracted for entities and relations belonging to three unseen categories (namely, Film, Scientist, and MusicalWork). This version of the dataset is bilingual, supporting English and Russian. The English version consists of 16,657 triple sets extracted from 45,040 texts. This is an improved relaunch of the dataset introduced in the context of the WebNLG 2017 challenge [4]. In 2023, a version for under-represented languages, including Breton, Irish, Maltese and Welsh, along with the Russian part of the previous dataset, was published in the context of the 2023 WebNLG Shared Task on Low Resource Languages [6]. In [7], the Italian version of the WebNLG corpus is introduced, i.e., they use machine translation, automatic error detection, and manual post-editing to produce high-quality RDF-to-text pairs, while instruction-tuned LLMs are employed to evaluate the resulting alignments. Because our update methodology relies on systematically checking and mapping structural entity and property URIs rather than parsing natural language, this same semi-automatic process can be seamlessly applied to extend WebNLG++ to Russian or other supported languages. Furthermore, it is important to note that because the texts are verbalizations of semantic triples from DBpedia, the benchmark reflects the knowledge embedded in periodic DBpedia releases rather than real-time Wikipedia edits. WebNLG+ relies on an earlier version of DBpedia, and this dependency introduces a notable limitation. Over time, several properties and URIs in DBpedia have been modified, deprecated, or replaced, which can create inconsistencies between the benchmark and the current knowledge base, ultimately leading to potential drops in the accuracy and reliability of automatic evaluation scores. Specifically, we have identified three cases of changes:

(i) change of ontology predicate, but maintaining the same meaning (e.g. “city” → “locationCity”)

(ii) change of subject or object URI, while the entity is the same (e.g. “Otkrytiye_Arena” → “Otkritie_Arena”)

(iii) change of subject or object URI , while the entity is different (e.g. a team now plays in a different league, thus “Campeonato_Brasileiro_Série_C” → “Campeonato_Brasileiro_Série_D”)

Therefore, the primary objective of this work is to develop a harmonized, up-to-date, and internally consistent version of the WebNLG+ test set. This can serve as a benchmark for LLM-related approaches that are constantly emerging in the literature. We refer to this new version as WebNLG++.

It is important to note that for WebNLG++, we deliberately chose to update the RDF-to-text (generation) test split rather than the semantic-parsing split. We utilized the single target sentence provided per set as the extraction input. This split was chosen for its broader coverage: it contains 1,779 triple sets with approximately 50% unseen entities and categories, compared to only 752 sets in the semantic-parsing split. This wider scope more rigorously stresses ontology drift across all 19 categories, which aligns with the core focus of our resource.

3.2. Use Cases

Here, we identify three use cases of the WebNLG++ benchmark: UC1: benchmark relevance for LLMs, UC2: URIs and ontology evolution, and UC3: methods for temporal facts evaluation. For each of these use cases, we provide a separate file in the JSON format. The UCs and the corresponding files are analyzed below:

UC1. Benchmark Relevance for LLMs. By updating the alignments to the current DBpedia, the output of the benchmark can be better aligned with recent world knowledge embedded in LLMs. This can reduce the mismatch between the parametric knowledge of the LLM and benchmark data, ensuring a more reliable test bed for solutions relying on the most recent LLMs.

`File1: webnlg_plus_mod_ground_truth.json`: This file contains the original references (texts) and targets (the sentence from which the user extracts triples) from WebNLG, but contains the ground truth triples from WebNLG++ (i.e., it considers the original texts and all three cases of changes of §3.1).

UC2. Predicate Evolution. Since the latest DBpedia release introduces ontology updates (e.g., new or replaced predicates), we can evaluate model performance on previously unseen predicates.

`File2: webnlg_plus_mod_pred_ground_truth.json`: This file contains the original references and targets from WebNLG+. However, its ground truth triples consist of the WebNLG+ subjects and objects, and the WebNLG++ predicates (i.e., it considers the original texts and only the first case of changes of §3.1).

UC3. Methods for Temporal Facts Evaluation. It enables the study of how models handle temporal changes in factual knowledge by comparing their outputs against time-sensitive updates in entities and relations. It supports the evaluation of model robustness to evolving facts and ontology shifts across time.

`File3: webnlg_plus_plus_ground_truth.json`: This file contains the updated ground truth triples, and the updated reference and target texts. Its references and targets are modified based on the updated subjects and objects (i.e., it considers the modified texts and all three cases of changes of §3.1).

Below, we can see examples of initial and modified triples over files.

1. **No Changes**

```
"initial": "Estádio_Municipal_Coaracy_da_Mata_Fonseca | location | Arapiraca",
"modified": "Estádio_Municipal_Coaracy_da_Mata_Fonseca | location | Arapiraca"
```

2. **Predicate Modifications Only (UC2)**

```
"initial": "MotorSport_Vision | city | Fawkham",
"modified": "MotorSport_Vision |locationCity | Fawkham"
```

3. **URIs Modifications (UC1, UC3)**

```
"initial": "FC_Spartak_Moscow | ground | Otkrytiye_Arena",
"modified": "FC_Spartak_Moscow | ground | Otkritie_Arena"
```

4. **Both URI & Predicate Modifications (UC1, UC3)**

```
"initial": "Adolfo_Suárez_Madrid-Barajas_Airport | operatingOrganisation | ENAIRE",
"modified": "Adolfo_Suárez_Madrid-Barajas_Airport | operator | Aena"
```

5. **Fact Modifications of Texts (UC3)**

```
"initial": Estádio Municipal Coaracy da Mata Fonseca is the name of the ground of
Agremiação Sportiva Arapiraquense in Arapiraca. Agremiação Sportiva Arapiraquense,
nicknamed \"Alvinegro\", lay in the Campeonato Brasileiro Série C league  from Brazil.
"modified": Estádio Municipal Coaracy da Mata Fonseca is the name of the ground of
Agremiação Sportiva Arapiraquense in Arapiraca. Agremiação Sportiva Arapiraquense,
nicknamed \"Alvinegro\", lay in the Campeonato Brasileiro Série D league  from Brazil.
```

3.3. The Creation Process

The creation of WebNLG++ involves the systematic alignment and standardization of the RDF triples and entity references in WebNLG+ with the current version of DBpedia. The process aims to identify and resolve data inconsistencies, update outdated or incomplete entries, and ensure semantic correspondence between each set of triples and its associated natural language verbalization. The process consists of three main steps:

• **Segregation by Entity Category**: The WebNLG+ test set was first divided by the 19 distinct entity categories that the dataset covers. The first 15 *seen* categories come from the original version of 2017 WebNLG: Airport, Artist, Astronaut, Athlete, Building, CelestialBody, City, ComicsCharacter, Food, MeansOfTransportation, Monument, Politician, SportsTeam, University and WrittenWork. The WebNLG+ added one more category: Company, and last 3 *unseen* categories: Film, MusicalWork and Scientist.

• **DBpedia Mapping:** Every element of each RDF triple was systematically located and mapped to its entry in DBpedia via its SPARQL endpoint (https://dbpedia.org/sparql). We developed a Python-based automated script that reads triples from the WebNLG+ JSON files and constructs type-aware SPARQL ASK queries. To ensure high retrieval accuracy, the script automatically formats text into appropriate DBpedia resources, English language literals (@en), and explicit xsd data types (e.g., date, double for measurements, and integer). This automated verification categorizes triples into *existing* and *not existing* sets, isolating the exact subset of data that requires manual intervention.

• **Manual Correction and Human-in-the-Loop Validation:** For the triples that were not automatically found, a manual verification was performed by searching each subject, predicate, and object in DBpedia to identify possible matches. When any component had a different but equivalent form or meaning, the triple was corrected using the appropriate corresponding DBpedia entries. To ensure the highest quality of these corrections, we employed a rigorous two-pass human-in-the-loop validation process. In the first pass, three authors reviewed the 993 manually processed cases and proposed updates. While we did not calculate formal inter-annotator agreement metrics (like Cohen's Kappa) because the validation was conducted as an iterative, consensus-building process, the qualitative agreement was highly consistent. The very few edge cases that provoked disagreement were immediately addressed through joint discussion among all authors to reach a final consensus.

By combining automated DBpedia mapping with this targeted manual validation, the construction process minimizes human effort. This semi-automatic pipeline shifts the workload from rediscovering entities from scratch to merely validating a focused subset of automatically flagged candidates. We detail how this methodology serves as an efficient, long-term maintenance strategy for future benchmark updates in Section 3.5. The project was conducted over the course of about one month (in November 2025) and led to targeted corrections in one or more parts of the triple: i) subject only, ii) subject and predicate, iii) subject and object, iv) predicate only, v) predicate and object, or vi) object only. No subject–predicate–object modification was observed. A summary of the distribution of modified triples across all texts is provided in Table 1.

3.4. WebNLG++: Drawer of Modifications

In the realm of entity alterations, we differentiate between two main kinds of modifications. The first involves *structural changes*, like URIs. The second pertains to *factual modifications* in which the characteristics of an entity change, for instance, a soccer player's team connection or a politician's political affiliation.

An analysis of the dataset reveals that out of an initial 5,639 triples, 2,159 (38.3%) were modified, while the majority (61.7%) remained unchanged. The modifications are distributed across the subject, predicate, and object of the triples, as shown in Table 2. Predicate changes were the most common single-element modification category, accounting for 935 changes (16.5% of the total dataset). However, a more detailed breakdown shows that modifications affecting only the object were the most frequent specific change type, with 1,052 instances (18.66%). This was followed by changes to only the predicate at 738

Table 1
Text Distribution by number of modified triples of WebNLG++.

Number of Modified Triples	Number of Affected Texts
1	369
2	348
3	351
4	305
5	213
6	114
7	79

Table 2
Summary of Change Types Across Triples.

Change Type	Count	Percentage (%)
No changes (Triples remained the same)	3,480	61.73
Only object changed	1,052	18.66
Only predicate changed	738	13.09
Predicate & object changed	182	3.23
Only subject changed	95	1.69
Subject & object changed	77	1.37
Subject & predicate changed	15	0.27
Subject, predicate, & object changed	0	0.00
Total	**5639**	**100**

Table 3
WebNLG++: Top 5 Predicate Modifications by Count.

Category	Initial Predicate	Modified Predicate	Count
MusicalWork	precededBy	prevTitle	77
MusicalWork	followedBy	nextTitle	76
Airport	operatingOrganisation	operator	58
Airport	elevationAboveTheSeaLevel	elevation	37
Scientist	professionalField	fields	30

instances (13.08%). Changes to the subject were the least common, totalling 187 instances (3.3%). Notably, subject modifications that were classified as "Fact changes" (where the entity's attributes change but its URI does not), were 12 instances and URI changes or redirections were 175 instances, with these changes concentrated in the Politician (92) and Company (70) categories. In contrast, modifications to the object were more frequent, with 475 instances (8.4%) for URI changes and 836 instances (14.8%) for fact changes. Complex modifications involving multiple parts of a triple, such as changes to both the subject and object, were rare (77 instances, 1.36%). The analysis of predicate modifications, reveals a total of 935 changes covering various entity types. The modifications were not evenly distributed, the MusicalWork and Airport categories saw the most significant number of changes, with 155 and 153 modifications, respectively. Also, a frequent individual change was *precededBy* becoming *prevTitle* in the MusicalWork category, occurring 77 times. In contrast, the CelestialBody entity had no predicate modifications at all. The nature of these changes predominantly involves mapping to synonymous or more conventional predicate names. For instance, *operatingOrganisation* was changed to *operator* 58 times for Airport entities, and *affiliation* was mapped to *academicAffiliations* 29 times for University entities. In the Scientist category, the predicate *professionalField* was modified to *fields*, *field*, or *academicDiscipline* in numerous instances. This shows a clear effort to normalize the predicates within the dataset. Table 3 shows the most frequent predicate modifications and the corresponding categories.

The most dominant change was a change to only the object (1,052 instances), while changes to the subject were rare (187 instances). The 935 predicate modifications primarily involved normalization to semantically equivelant terms, such as mapping *operatingOrganisation* to *operator*. Following the modification of the WebNLG+ dataset, a substantial alignment with the current version of DBpedia was achieved. From the total of 5,639 triples that were processed, 4,646 were successfully found within DBpedia, i.e., a rate of approximately 82.4%.

3.5. Maintenance Strategy

We provide an updated version of the benchmark for 2025, while noting that the same methodology can be systematically applied for future updates. To support efficient maintenance, our GitHub repository provides the SPARQL-based verification scripts that automatically identify the *delta*, i.e., triples that have become invalid due to changes in a newer ontology version. The scripts are designed for high-throughput periodic checking, featuring type-handling for ambiguous literals and incremental auto-saving to ensure robust execution over large datasets. This automation substantially reduces the required human effort, shifting the task from rediscovering entities from scratch to validating a focused subset of automatically flagged candidates (outputted directly as isolated JSON files). In the 2025 update, this delta comprised

38% of the benchmark, demonstrating that the approach scales well for periodic updates. Since the DBpedia ontology is continuously developed via the DBpedia Mappings Wiki and each change is released as a new version through the DBpedia Databus, benchmark updates should ideally be triggered when the detected delta exceeds a meaningful threshold. Because executing our automated DBpedia verification script over the entire dataset takes only about 5 to 6 hours, this strategy enables highly efficient and sustainable maintenance of the benchmark, which we recommend performing.

Finally, the static release of the WebNLG++ dataset, including the three Use Case splits, is persistently hosted on Zenodo (DOI: 10.5281/zenodo.17695689). The automated SPARQL-based verification scripts, delta-detection pipeline, and reproducibility instructions are maintained on GitHub at https://github.com/baniotis/webnlg_plus_plus_dataset/tree/main.

4. Evaluation Methodology

To account for the strict syntactic constraints of RDF and the semantic flexibility of LLMs, we evaluate triples using a two-steps pipeline. Exact Match requires predicted triples to perfectly match the ground-truth strings. If no exact match is found, we apply owl:sameAs matching. Since Exact Match heavily over-penalizes valid surface-level and URI variations, all empirical results reported in Section 5 utilize the owl:sameAs metric (which inherently includes Exact Matches).

A predicted triple $(s_{pred}, p_{pred}, o_{pred})$ is a soft match to a ground-truth triple (s_{gt}, p_{gt}, o_{gt}) if it satisfies both of the following conditions:

1. Entity Equivalence: Both subjects and objects must align via exact string match, DBpedia Spotlight[30], or `owl:sameAs` equivalence classes:

$$[e] = \{e\} \cup \{e' \mid (e, \texttt{owl:sameAs}, e')\}$$

2. Predicate Equivalence: If entities align, predicates are compared using `all-MiniLM-L6-v2` cosine similarity under a dual-threshold logic. Predicates are equivalent if they satisfy *either* condition:

$$\text{cos_sim}(p_{pred}, p_{gt}) \geq 0.5 \quad \vee \quad \text{cos_sim}(p_{pred} \oplus o_{pred}, p_{gt} \oplus o_{gt}) \geq 0.75$$

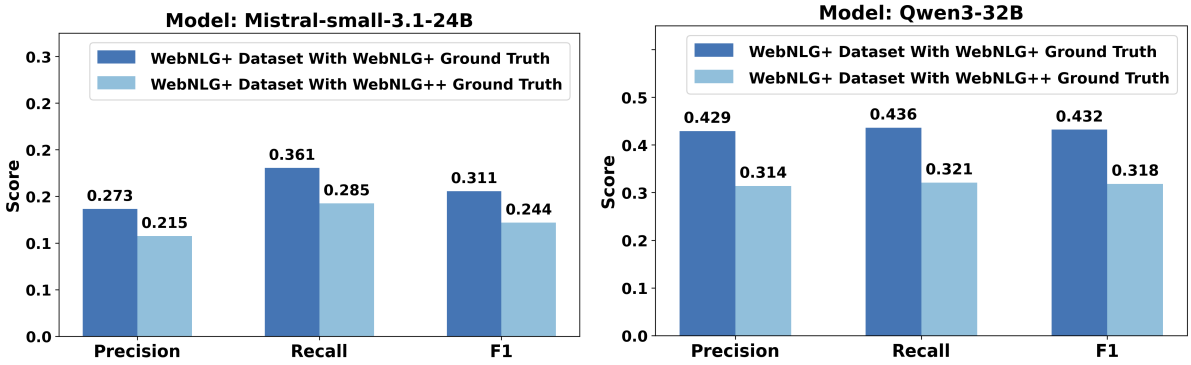
The $\tau \geq 0.5$ threshold captures high semantic overlap in isolated predicates (e.g., safely treating `dbo:author` $\approx$ `dbo:writer` as equivalent), while the stricter $\tau \geq 0.75$ threshold evaluates the concatenated predicate-object context ($p \oplus o$) if the isolated match falls short. To validate these dual thresholds, we relied on a qualitative manual inspection of boundary cases during the alignment process. We did not conduct a large-scale statistical sensitivity analysis, but we systematically reviewed edge-case matches. We observed that lowering the isolated predicate threshold below $\tau = 0.5$ introduced false positives by conflating distinct but related relations, whereas raising the concatenated threshold above $\tau = 0.75$ penalized valid semantic variations, reducing recall. By manually verifying these boundary matches, we ensured our chosen thresholds practically balance strictness with semantic flexibility.

5. Experiments over the WebNLG++ benchmark

5.1. LLM-based Experiments

We conducted a series of experiments to evaluate triple extraction capabilities of LLMs. For this task, we used the 1,779 texts from the benchmark test set and employed two open-weight models: Qwen3 [31] with 32 billion parameters (Qwen) and Mistral-small-3.1 (https://huggingface.co/mistralai/Mistral-Small-3.1-24B-Instruct-2503) (Mistral) with 24 billion parameters. We chose not to run experiments with reduced-parameter versions of these models, as our primary goal was to evaluate their standard, full-scale, publicly-released versions. Each model was instructed with the following prompt:

```
You are an expert in Semantic Web and DBpedia ontology. Your task is to extract as many
valid RDF triples as possible from the given text. Every part of the triple should be a
valid DBpedia property. Your output should  be only the extracted  triples and nothing else.
Example Input: "The Aarhus is the airport of Aarhus, Denmark."
Example Output: "Aarhus_Airport | cityServed | Aarhus, Denmark"
```



(a) Mistral model. The figure compares results obtained using the original WebNLG+ dataset and Use Case 1.

(b) Qwen model. The figure compares results obtained using the original WebNLG+ dataset and Use Case 1.

Figure 1: Comparison of model performance on the original WebNLG+ and WebNLG++ Use Case 1 datasets across Precision, Recall and F1 metrics (1,779 entries).

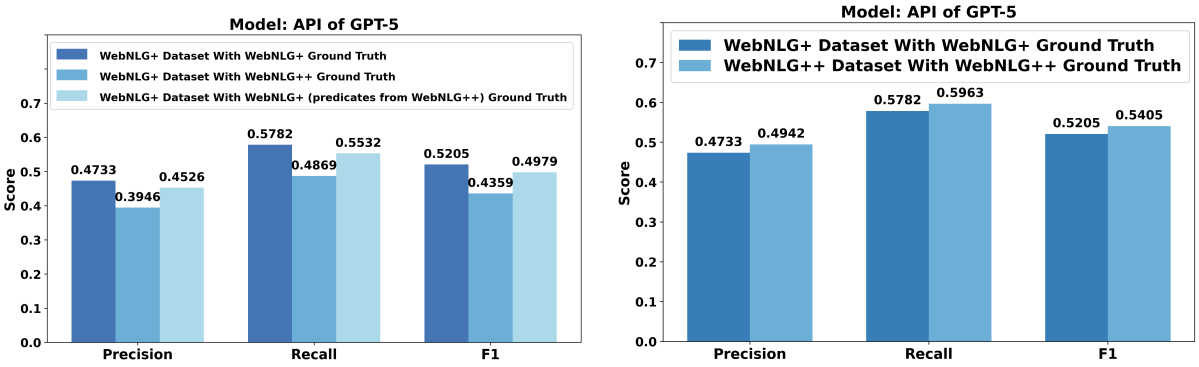
To evaluate the quality of the generated triples, we calculated precision, recall, and F1-score. The evaluation was performed against two references: the original WebNLG+ test set gold standard and our WebNLG++ test set gold standard (Use Case 1). In terms of computational time, a great performance difference was observed between the two models. Executing the triple extraction task on an NVIDIA RTX 3090 GPU, the Mistral model completed the process, significantly fast, in 1 hour and 20 minutes. In contrast, the Qwen model in the same GPU required 23 hours to complete the same task. This difference is large due to the fact that Qwen, in addition to having more parameters, has its thinking mode enabled by default, which naturally increases inference time. Because these models operate using fundamentally different inference processes, our comparison axis is strictly per-model performance across the gold standards (WebNLG+ vs. WebNLG++), rather than a direct cross-model ranking.

In the evaluation with the original WebNLG+ gold standard of the test set, we observed a clear performance gap between the two models. The Qwen model was better than the Mistral model, with F1 scores of 0.4319 for Qwen and 0.3106 for Mistral. We observed similar behaviour in precision, where Qwen achieved 0.4284 while Mistral achieved 0.2727, and also in recall, with scores of 0.4355 and 0.3609, respectively. This performance gap was expected, as Qwen's larger size and default thinking mode enable more robust performance despite longer inference time.

Regarding our WebNLG++ gold standard (Use Case 1) of the test set, again we observed a performance gap between the two models. The Qwen model was better that the Mistral model with an F1-score of 0.3175 to 0.2448, respectively. There was a similar behaviour in precision, where Qwen achieved 0.3143 while Mistral achieved 0.2145, and also in recall, with scores of 0.3207 and 0.2849, respectively. Result metrics of both the original and the improved gold standard can be found in Figure 1. Except from confirming the larger model's (Qwen) better performance, a detailed analysis of the evaluation metrics across the two gold standards reveals the specific nature of the challenge posed by WebNLG++ (Use Case 1). The percentage drop in precision for both models indicates that these models were trained on an older version of DBpedia, which includes URIs of properties and entities that have been changed in newer releases. A manual inspection confirms this observation. Consequently, when comparing their performance with that of a new model that has been trained on the latest version of DBpedia, the new model may appear to perform worse, as it would generate triples containing URIs not present in the ground truth. This underscores the importance of careful benchmark consideration; a fair evaluation should include both benchmark versions.

Experiments using the API of GPT-5. We repeated the triple extraction experiment on *WebNLG+* using the *GPT-5 API* (GPT-5) with the predefined settings and the same conditions. The same 1779 texts from the *WebNLG+* test set were used, and the identical prompt, was employed to ensure comparability. This allowed us to directly contrast the extraction performance of *GPT-5* with the previously tested *Qwen3:32B* (Qwen) and *Mistral-small-3.1-24B* (Mistral) models on the task of RDF triple generation grounded in *DBpedia* ontology.

After inspecting the triples that were produced by *GPT-5*, we observed that the model did not restrict



(a) Performance of GPT-5 model on the WebNLG+ dataset under three configurations: (1) the original WebNLG+, (2) Use Case 1 and (3) Use Case 2.

(b) Performance of GPT-5 model on the WebNLG datasets. The figure compares evaluation results using the original WebNLG+ and Use Case 3.

Figure 2: Comparison of GPT-5 performance across the original WebNLG+ and WebNLG++. Each subfigure shows results over 1,779 test entries.

itself to a single snapshot of *DBpedia*. Instead, it produced triples that incorporated both the properties and the entities of earlier versions as well as newer versions of the *DBpedia* ontology. For example, while *"Expect a Miracle | precededBy | Pools of Light"* and *"Expect a Miracle | prevTitle | Pools of Light"* provide the same information, *precededBy* belongs to an older version of DBpedia and *prevTitle* to a newer one. The behavior indicates that GPT-5 employed a greater historical corpus of DBpedia that, in some cases, improved coverage for entities and relations but introduced inconsistencies since properties from retired schemata were combined with those from the active ontology. To evaluate the quality of the generated triples, we calculated precision, recall, and F1-score. The evaluation was performed against three references: i) the original WebNLG+ test set gold standard, ii) our WebNLG++ test set gold standard (Use Case 1), and iii) the WebNLG+ test set gold standard with the modified predicates of the WebNLG++ test set gold standard (Use Case 2).

In our evaluation, as shown in Figure 2a, across the different models and dataset variants, we find that GPT-5 consistently outperforms Mistral and Qwen in F1, precision, and recall. On the original WebNLG+ dataset, GPT-5 achieves the highest F1 score 0.5205 and recall 0.5782, while Mistral lags behind with an F1 of 0.3110. Qwen performs competitively on WebNLG+ (F1 of 0.4320), but its scores lower when moving to WebNLG++ (Use case 1), which indicates lower robustness to updated DBpedia alignments. In contrast, GPT-5 maintains strong performance across WebNLG++ (Use Case 1), with only a slight reduction (F1 of 0.4359). The WebNLG++ (Predicates Modified Only)(Use Case 2) variant shows that much of the improvement in GPT-5's results comes from predicate alignment (F1 of 0.4979), while entity-level updates introduce additional variability. Overall, GPT-5 demonstrates both higher accuracy and adaptability compared to the other models, while its generation of triples with a mix of old and new DBpedia URIs suggests that evaluation remains dependent on dataset alignment.

Triple Extraction from WebNLG++ using GPT-5. We also repeated the triple extraction experiment (Use Case 3) on WebNLG++ using the GPT-5 API. The 1,779 texts from the WebNLG++ test set were used, and the identical prompt, was employed to ensure comparability. After modifying the WebNLG++ texts and targets, we carried out the evaluation, obtaining a precision of 0.4942, a recall of 0.5963, and an F1 score of 0.5405 (Figure 2b). A complete summary of the results across all tested LLMs and Use Cases is provided in Table 4. We notice a slight improvement in the evaluation results when considering the updated factual knowledge of WebNLG++ (Use Case 3). Finally, there is a broader challenge in one-shot LLM evaluation: models often rely heavily on their internal parametric knowledge rather than strictly adhering to the provided extraction context. Consequently, an LLM may generate a factually correct triple that is unfaithful to the specific input text. This underscores the necessity of designing robust extraction contexts and evaluation frameworks that can distinguish between a model's internal knowledge retrieval and its actual extraction capabilities. To evaluate the models under realistic, out-of-the-box conditions, we conducted single-pass runs using their default generation parameters. While the non-deterministic nature of default LLM generation might cause slight fluctuations in the tight

Table 4
Comparison of results across Mistral, Qwen, and GPT-5 models on the Original WebNLG+(Use Case : -) and the three use cases, sorted by F1 score.

LLM	Dataset	Ground Truth	UC	Precision	Recall	F1
Mistral-small-3.1:24B	WebNLG+	WebNLG++	UC1	0.2150	0.2850	0.2440
Mistral-small-3.1:24B	WebNLG+	WebNLG+	–	0.2730	0.3610	0.3110
Qwen3:32B	WebNLG+	WebNLG++	UC1	0.3140	0.3210	0.3180
Qwen3:32B	WebNLG+	WebNLG+	–	0.4290	0.4360	0.4320
GPT-5	WebNLG+	WebNLG++	UC1	0.3946	0.4869	0.4359
GPT-5	WebNLG+	WebNLG+ with predicates from WebNLG++	UC2	0.4526	0.5532	0.4979
GPT-5	WebNLG+	WebNLG+	–	0.4733	0.5782	0.5205
GPT-5	WebNLG++	WebNLG++	UC3	**0.4942**	**0.5963**	**0.5405**

Table 5
Evaluation results on the WebNLG++ dataset over pretrained models.

Model	Precision	Recall	F1-score
UniRel	0.0890	0.0173	0.0290
PFN	0.1435	0.0488	0.0728
KnowGL	**0.3280**	**0.1275**	**0.1836**

F1 margins across repeated runs, this setup effectively establishes a baseline for how general porpuse LLMs interact with evolving ontologies. Ultimately, these experiments are designed to demonstrate the critical performance delta between the legacy and updated ground truths, rather than to establish an exhaustive cross-model ranking.

5.2. WebNLG++ over Pretrained Large Models (PLMs)

There are many pretrained models that exhibit state-of-the-art results after a preceding training phase that requires a large volume of data. Some notable examples are PFN, UniRel and KnowGL that are used for the evaluation below, i.e., for each of them we use the PLM that was distributed by their creators and we run an evaluation on the WebNLG++.

UniRel [10] offers a unified representation and interaction addressing both the heterogeneity of representations and the absence of interaction dependencies. For the triplets to be extracted, *UniRel* first encodes both relation and entity into meaningful sequence embeddings to construct unified representations. Potential relations are initially transformed into natural language texts and a continuous sequence is created alongside the input statement. Next, it utilizes a PLM [32] to encode the sequence, effectively capturing their informative relationships. Finally, it concurrently represents both entity-entity and entity-relation interactions by utilizing the existing self-attention mechanism found in transformers. When *UniRel* was evaluated on the *WebNLG* dataset, it achieved an F1-score of 0.947, as reported in [32]. However, by using the PLM on WebNLG++, the performance was poor, with an F1-score of 0.029, mainly due to the inability of the model to return triples for many of the input texts. The test set of the WebNLG+ benchmark (and of WebNLG++) contains a much higher number of unseen categories of entities compared to the test set of the original WebNLG.

PFN [33] is a framework designed specifically for joint encoding in the task of entity and relation extraction. This approach can encode task-specific features and ensure effective bidirectional interaction between named entity recognition (NER) and relation extraction (RE). Each step of the encoder follows a partition and filter mechanism. NER and RE are expressed as table-filling challenges, in which every pair of words receives a score for entity spans or relation triples. *PFN* was evaluated on *WebNLG*, and achieved 0.936 F1 score [33]. However, by using it on WebNLG++, the F1-score was quite low, i.e., 0.0728.

KnowGL [34] is a framework that addresses relational triple extraction by treating fact extraction as an autoregressive generation task. Given an input sentence, it generates a structured sequence encoding the subject, relation, and object, enriched with entity labels and types, by using the BART-large model

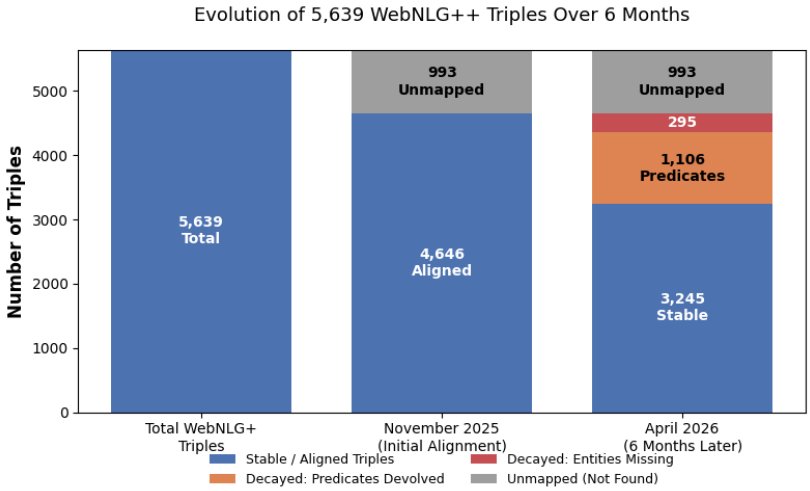


Figure 3: Evolution of the WebNLG++ benchmark from November 2025 to April 2026.

fine-tuned on an extended version of the *REBEL* [35] dataset. The model outputs a ranked list of facts based on their occurrence and confidence across multiple decoding beams. Also, *KnowGL* links generated entities, types, and relations to Wikidata IDs when possible, enabling both KG population and new facts discovery. Evaluated on the *REBEL* test set, *KnowGL* achieved an F1 score of 0.7074, outperforming a traditional IE pipeline (F1 of 0.4250) and the previous novel generative IE model [36] (F1 of 0.6893). While it was not originally tested on WebNLG, WebNLG++ offers an opportunity to evaluate its capabilities more comprehensively. In the evaluation with WebNLG++, it reached an F1 of 0.1836. It is important to mention that KnowGL's grounding in Wikidata and REBEL introduces a knowledge base mismatch confound when evaluated on a DBpedia-based benchmark. Thereby, this low performance illustrates the brittleness of such models when crossing knowledge boundaries, rather than a failure of the benchmark itself. We notice that PLMs excel on benchmark datasets such as WebNLG in in-dataset testing. However, when evaluating these models on new benchmarks, we observe a significant drop in performance, i.e., they cannot easily adapt to linguistic and factual changes.

5.3. Evaluating Benchmark Evolution

The rapid evolution of DBpedia highlights the necessity of our automated maintenance strategy. A follow-up validation in April 2026, just six months after our initial alignment, revealed that 1,401 of 4,646 aligned triples (30.1%) had undergone further structural shifts (see Figure 3). Specifically, 295 entities were no longer present, and 1,106 predicates devolved into generic wikiPageWikiLink properties. This proves that static benchmarks are insufficient and underscores the critical need for benchmark updates.

6. Conclusion

We introduced WebNLG++, an updated WebNLG test set containing 1,779 texts and 5,639 triples, of which 2,159 (38.3%) were modified from the original WebNLG+. The dataset was created via a semi-automatic SPARQL-based pipeline with human validation. Crucially, although focused on DBpedia, our semi-automatic maintenance pipeline is designed with the potential to be adapted to other evolving KGs, though future work is required to empirically verify this beyond DBpedia. Using this benchmark, we evaluated three LLMs (Qwen-3-32B, Mistral-small-3.1-24B, GPT-5) and three pretrained transformer-based encoder-only models (UniRel, PFN, KnowGL). LLMs performed better even in a one-shot prompting setting, while pretrained models underperformed on unseen text types. Results on WebNLG++ differ substantially from those on the older version, underscoring the need for up-to-date benchmarks. A follow-up evaluation in April 2026 revealed that 30.1% of aligned triples decayed within six months, showing that static benchmarks are no longer sufficient. The updated test set enables accurate evaluation of LLM-based approaches without fine-tuning, reflecting real-world deployment scenarios. In future work, we will prepare an updated version of the training set to support fine-tuning.

Declaration of use of Generative AI. The authors used Google Gemini for language polishing and structural formatting. After using this tool, the authors reviewed and edited the content and take full responsibility for the final manuscript.

References

[1] B. J. Gutiérrez, Y. Shu, Y. Gu, M. Yasunaga, Y. Su, Hipporag: neurobiologically inspired long-term memory for large language models, in: Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS '24, Curran Associates Inc., Red Hook, NY, USA, 2024.

[2] S. Li, Y. He, H. Guo, X. Bu, G. Bai, J. Liu, J. Liu, X. Qu, Y. Li, W. Ouyang, W. Su, B. Zheng, GraphReader: Building graph-based agent to enhance long-context abilities of large language models, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 12758–12786. URL: https://aclanthology.org/2024.findings-emnlp.746/. doi:10.18653/v1/2024.findings-emnlp.746.

[3] Z. Shen, C. Diao, P. Vougiouklis, P. Merita, S. Piramanayagam, E. Chen, D. Graux, A. Melo, R. Lai, Z. Jiang, Z. Li, Y. Qi, Y. Ren, D. Tu, J. Z. Pan, GeAR: Graph-enhanced agent for retrieval-augmented generation, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Findings of the Association for Computational Linguistics: ACL 2025, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 12049–12072. URL: https://aclanthology.org/2025.findings-acl.624/. doi:10.18653/v1/2025.findings-acl.624.

[4] 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.

[5] T. Castro Ferreira, C. Gardent, N. Ilinykh, C. van der Lee, S. Mille, D. Moussallem, A. Shimorina, The 2020 bilingual, bi-directional webnlg+ shared task overview and evaluation results (webnlg+ 2020), in: Proceedings of the 3rd WebNLG Workshop on Natural Language Generation from the Semantic Web (WebNLG+ 2020), Association for Computational Linguistics, Dublin, Ireland (Virtual), 2020, pp. 55–76.

[6] L. Cripwell, A. Belz, C. Gardent, A. Gatt, C. Borg, M. Borg, J. Judge, M. Lorandi, A. Nikiforovskaya, W. Soto-Martinez, C. Thomson, The 2023 WebNLG shared task on low resource languages. overview and evaluation results (WebNLG 2023), in: A. Gatt, C. Gardent, L. Cripwell, A. Belz, C. Borg, A. Erdem, E. Erdem (Eds.), Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023), Association for Computational Linguistics, Prague, Czech Republic, 2023, pp. 55–66. URL: https://aclanthology.org/2023.mmnlg-1.6/.

[7] M. Oliverio, P. F. Balestrucci, A. Mazzei, V. Basile, WebNLG-IT: Construction of an aligned RDF-Italian corpus through machine translation techniques, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Findings of the Association for Computational Linguistics: ACL 2025, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 12073–12083. URL: https://aclanthology.org/2025.findings-acl.625/. doi:10.18653/v1/2025.findings-acl.625.

[8] T. Castro Ferreira, C. Gardent, N. Ilinykh, C. van der Lee, S. Mille, D. Moussallem, A. Shimorina, The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020), in: T. Castro Ferreira, C. Gardent, N. Ilinykh, C. van der Lee, S. Mille, D. Moussallem, A. Shimorina (Eds.), Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+), Association for Computational Linguistics, Dublin, Ireland (Virtual), 2020, pp. 55–76. URL: https://aclanthology.org/2020.webnlg-1.7/.

[9] X. Li, X. Luo, C. Dong, D. Yang, B. Luan, Z. He, Tdeer: An efficient translating decoding schema

for joint extraction of entities and relations, in: Proceedings of the 2021 conference on empirical methods in natural language processing, 2021, pp. 8055–8064.

[10] W. Tang, B. Xu, Y. Zhao, Z. Mao, Y. Liu, Y. Liao, H. Xie, Unirel: Unified representation and interaction for joint relational triple extraction, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 7087–7099.

[11] J. Ning, Z. Yang, Y. Sun, Z. Wang, H. Lin, OD-RTE: A One-Stage Object Detection Framework for Relational Triple Extraction, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 11120–11135.

[12] D. Sui, X. Zeng, Y. Chen, K. Liu, J. Zhao, Joint entity and relation extraction with set prediction networks, IEEE transactions on neural networks and learning systems (2023).

[13] Y.-M. Shang, H. Huang, X. Mao, OneRel: Joint Entity and Relation Extraction with One Module in One Step, Proceedings of the AAAI Conference on Artificial Intelligence 36 (2022) 11285–11293. URL: https://ojs.aaai.org/index.php/AAAI/article/view/21379. doi:10.1609/aaai.v36i10.21379.

[14] Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, L. Sun, TPLinker: Single-stage joint extraction of entities and relations through token pair linking, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 1572–1582. URL: https://www.aclweb.org/anthology/2020.coling-main.138.

[15] A. Papaluca, D. Krefl, S. Rodríguez Méndez, A. Lensky, H. Suominen, Zero- and Few-Shots Knowledge Graph Triplet Extraction with Large Language Models, in: R. Biswas, L.-A. Kaffee, O. Agarwal, P. Minervini, S. Singh, G. de Melo (Eds.), Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 12–23. URL: https://aclanthology.org/2024.kallm-1.2/. doi:10.18653/v1/2024.kallm-1.2.

[16] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. URL: https://arxiv.org/abs/2302.13971. arXiv:2302.13971.

[17] S. Riedel, L. Yao, A. McCallum, Modeling relations and their mentions without labeled text, in: J. L. Balcázar, F. Bonchi, A. Gionis, M. Sebag (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 148–163.

[18] Y. Zhang, V. Zhong, D. Chen, G. Angeli, C. D. Manning, Position-aware attention and supervised data improve slot filling, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), 2017, pp. 35–45. URL: https://nlp.stanford.edu/pubs/zhang2017tacred.pdf.

[19] C. Alt, A. Gabryszak, L. Hennig, TACRED revisited: A thorough evaluation of the TACRED relation extraction task, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 1558–1569. URL: https://aclanthology.org/2020.acl-main.142/. doi:10.18653/v1/2020.acl-main.142.

[20] L. Hennig, P. Thomas, S. Möller, MultiTACRED: A multilingual version of the TAC relation extraction dataset, 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. 3785–3801. URL: https://aclanthology.org/2023.acl-long.210/. doi:10.18653/v1/2023.acl-long.210.

[21] X. Han, H. Zhu, P. Yu, Z. Wang, Y. Yao, Z. Liu, M. Sun, Fewrel:a large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation, in: EMNLP, 2018.

[22] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, M. Sun, DocRED: A large-scale document-level relation extraction dataset, 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. 764–777. URL: https://aclanthology.org/P19-1074/. doi:10.18653/v1/P19-1074.

[23] L. Huguet Cabot, S. Tedeschi, A.-C. Ngonga Ngomo, R. Navigli, RED^{fm}: a filtered and multilingual relation extraction dataset, 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. 4326–4343. URL: https://aclanthology.org/2023.acl-long.237/. doi:10.18653/v1/2023.acl-long.237.

[24] H. Elsahar, P. Vougiouklis, A. Remaci, C. Gravier, J. Hare, F. Laforest, E. Simperl, T-REx: A large scale alignment of natural language with knowledge base triples, in: N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, T. Tokunaga (Eds.), Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018. URL: https://aclanthology.org/L18-1544/.

[25] N. Mihindukulasooriya, S. Tiwari, C. F. Enguix, K. Lata, Text2kgbench: A benchmark fornbsp;ontology-driven knowledge graph generation fromnbsp;text, in: The Semantic Web – ISWC 2023: 22nd International Semantic Web Conference, Athens, Greece, November 6–10, 2023, Proceedings, Part II, Springer-Verlag, Berlin, Heidelberg, 2023, p. 247–265. URL: https://doi.org/10.1007/978-3-031-47243-5_14. doi:10.1007/978-3-031-47243-5_14.

[26] J. Plu, O. Moreno Escobar, E. Trouillez, A. Gapin, P. Lisena, T. Ehrhart, R. Troncy, Text2KGBench-LettrIA: A refined benchmark for Text2Graph systems, in: KBC-LM and LM-KBC Challenge (Knowledge Base Construction from Pre-trained Language Models) at ISWC 2025, 24th International Semantic Web Conference, CEUR, Nara, Japan, 2025. URL: https://hal.science/hal-05301914.

[27] A. Tschalzev, L. Purucker, S. Lüdtke, F. Hutter, C. Bartelt, H. Stuckenschmidt, Unreflected use of tabular data repositories can undermine research quality, 2025. URL: https://arxiv.org/abs/2503.09159. arXiv:2503.09159.

[28] J. Ouyang, T. Pan, M. Cheng, R. Yan, Y. Luo, J. Lin, Q. Liu, HoH: A dynamic benchmark for evaluating the impact of outdated information on retrieval-augmented generation, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 6036–6063. URL: https://aclanthology.org/2025.acl-long.301/. doi:10.18653/v1/2025.acl-long.301.

[29] J. Frey, D. Streitmatter, F. Götz, S. Hellmann, N. Arndt, DBpedia Archivo: A Web-Scale Interface for Ontology Archiving Under Consumer-Oriented Aspects, 2020, pp. 19–35. doi:10.1007/978-3-030-59833-4_2.

[30] P. N. Mendes, M. Jakob, A. García-Silva, C. Bizer, Dbpedia spotlight: shedding light on the web of documents, in: Proceedings of the 7th international conference on semantic systems, 2011, pp. 1–8.

[31] Q. Team, Qwen3 technical report, 2025. URL: https://arxiv.org/abs/2505.09388. arXiv:2505.09388.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 6000–6010.

[33] Z. Yan, C. Zhang, J. Fu, Q. Zhang, Z. Wei, A Partition Filter Network for Joint Entity and Relation Extraction, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 185–197.

[34] G. Rossiello, M. F. M. Chowdhury, N. Mihindukulasooriya, O. Cornec, A. M. Gliozzo, Knowgl: Knowledge generation and linking from text, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, 2023, pp. 16476–16478.

[35] P.-L. H. Cabot, R. Navigli, REBEL: Relation extraction by end-to-end language generation, in: Findings of the Association for Computational Linguistics: EMNLP 2021, 2021, pp. 2370–2381.

[36] M. Josifoski, N. De Cao, M. Peyrard, F. Petroni, R. West, GenIE: Generative information extraction, in: M. Carpuat, M.-C. de Marneffe, I. V. Meza Ruiz (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States,

2022, pp. 4626–4643. URL: https://aclanthology.org/2022.naacl-main.342/. doi:10.18653/v1/2022.naacl-main.342.