WebNLG-IT: Construction of an aligned RDF-Italian corpus through Machine Translation techniques

Michael Oliverio[1], Pier Felice Balestrucci[1], Alessandro Mazzei[1], Valerio Basile[1]

[1]University of Turin,

{michael.oliverio, pierfelice.balestrucci, alessandro.mazzei, valerio.basile}@unito.it

Abstract

The main goal of this work is the creation of the Italian version of the WebNLG corpus through the application of Neural Machine Translation (NMT) and post-editing with hand-written rules. To achieve this goal, in a first step, several existing NMT models were analyzed and compared in order to identify the system with the highest performance on the original corpus. In a second step, after using the best NMT system, we semi-automatically designed and applied a number of rules to refine and improve the quality of the produced resource, creating a new corpus named WebNLG-IT. We used this resource for fine-tuning several LLMs for RDF-to-text tasks. In this way, comparing the performance of LLM-based generators on both Italian and English, we have (1) evaluated the quality of WebNLG-IT with respect to the original English version, (2) released the first fine-tuned LLM-based system for generating Italian from semantic web triples and (3) introduced an Italian version of a modular generation pipeline for RDF-to-text.

1 Introduction

WebNLG was originally created in 2017 in the context of the first WebNLG Challenge (Gardent et al., 2017) for generating English texts starting from an RDF set of triples, but it has been used as a reference corpus in several shared task competitions[1]. As a consequence of this quite large use, several works converted WebNLG into different languages, that are primarily German (Ferreira et al., 2018) and Russian (Shimorina et al., 2019), and partially Maltese, Irish, Breton and Welsh (Cripwell et al., 2023).

The present work aims to create the first aligned Italian RDF-to-text corpus, with the objective of enriching Italian resources in the domain of Natural Language Generation (NLG). In order to obtain a high-quality complete Italian version of WebNLG 3.0, we follow the approach of Shimorina et al. (2019): in a first step, we apply a high-quality neural machine translation system on the original English WebNLG 3.0, and in a second step, we improve the quality of the translations by using hand-written rules.

The first step of corpus construction was conducted using the DeepL machine translation system, after a preliminary phase of analysis and comparison with other neural machine translation models.[2] This comparison relied on specific metrics for evaluating machine translations. Additionally, a human evaluation of the DeepL outputs was conducted to assess the adequacy and fluency of the translations.

In the second step, after the automatic translation, a post-editing phase was carried out to improve the quality of the resource, following a similar approach to Shimorina et al. (2019), thus generating the first version of the Italian corpus, called WebNLG-IT.[3] During this phase, a classification of the detected errors was conducted following the taxonomy provided by Blain et al. (2011), in order to further analyze the corpus, resulting in an additional dataset of 198 labeled errors.

In contrast to Shimorina et al. (2019), our work introduces a novel step, in which Italian and multilingual LLMs were fine-tuned for the RDF-to-text task.

Finally, to further enhance the performance of LLM-based RDF-to-text systems, we integrated the fine-tuned models into a modular generation pipeline for the Italian text generation.

In summary, the main contributions we present in this work are:

[1] https://synalp.gitlabpages.inria.fr/webnlg-challenge/

[2] https://www.deepl.com/translator

[3] The data are released under the Creative Commons Attribution Non Commercial 4.0 International license and available at: https://github.com/MichaelOliverio/WebNLG-IT

Findings of the Association for Computational Linguistics: ACL 2025, pages 12073–12083

1. The public release of the WebNLG-IT corpus, a validated Italian version of WebNLG.

2. An analysis of the performance of various LLMs in generating verbalizations of data units across different language versions of WebNLG.

3. An analysis of the Italian version of an existing RDF-to-text generation pipeline.

2 Related work

Since 2017, versions of WebNLG have been developed in other languages. In 2018, the German version was released by Ferreira et al. (2018), created using the University of Edinburgh's Neural MT System (Sennrich et al., 2017). Subsequently, the same approach was adopted in 2019 for Russian, with an additional phase of post-editing aimed at improving the quality of the produced resource. During this phase, sentences containing errors were manually identified and annotated with their corrections. Through the use of the Levenshtein distance (Levenshtein, 1965), a set of rules was automatically extracted to create a rule-based post-editing system, with the aim of correcting the corpus produced.

Over the years, several WebNLG challenges have been held (2017, 2020 and 2023), aimed at developing the best RDF-to-text models based on WebNLG corpora (Castro Ferreira et al., 2020).[4]

In our work, we chose to fine-tune some LLMs using the WebNLG-IT corpus we produced to assess the performance that can be achieved by the models, comparing it with what is achievable using other corpora in different languages.

3 From WebNLG to WebNLG-IT

WebNLG is a linguistic resource composed of data units, each accompanied by one or more verbalizations manually written by expert annotators. The generated verbalization aims to capture the semantics represented by the given starting unit. For example:

Data unit:
`(ABILENE,_TEXAS COUNTRY UNITED_STATES)`
`(ABILENE,_TEXAS ISPARTOF TEXAS)`

verbalization:
Abilene is part of Texas, United States.

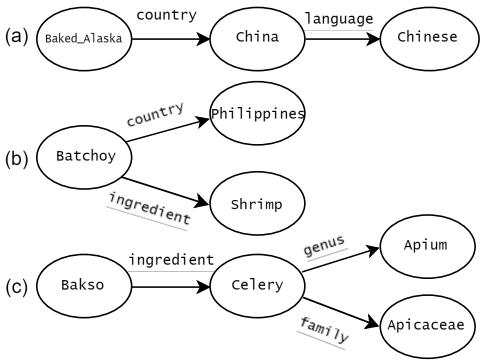

Figure 1: (a) The triples in the data unit are chain-related to each other. As can be seen, `Baked_Alaska` is the subject of the triple `(Baked_Alaska, country, China)`, while `China` is both the object of this triple and the subject of the triple `(China, language, Chinese)`. This allows for constructing verbalizations that maintain coherence in discourse. (b) The relation between triples in the data unit is defined as sibling. Triples share the same subject, `Batchoy`, allowing for constructing verbalizations that maintain coherence in theme. (c) Some triples in the data unit are sibling-related, while others are chain-related, hence referred to as triples in mixed relation. In this case, it is possible to construct verbalizations that maintain both theme and discourse coherence.

A data unit is a set of RDF triples, each containing a subject, predicate, and object, automatically extracted from 15 different categories of DBPedia. The choice of a large number of categories is due to the desire to generate a dataset that covers different topics, in order to obtain a resource with a high variety of data (Perez-Beltrachini et al., 2016).

Among the triples within the data units, there can be various types of relationships, summarised in Figure 1:

- Chain: the object of one triple is the subject of another triple;
- Sibling: the triples share the same subject;
- Mixed: the data unit contains both sibling and chain-related triples.

The extraction of triples with these different types of relations was carried out with the aim of obtaining diverse linguistic constructions.

The version of WebNLG used for the creation of the Italian corpus is 3.0, released during the WebNLG 2020 Challenge. The corpus has been

[4] https://synalp.gitlabpages.inria.fr/webnlg-challenge/challenge_2017/,https://synalp.gitlabpages.inria.fr/webnlg-challenge/challenge_2020/,https://synalp.gitlabpages.inria.fr/webnlg-challenge/challenge_2023/

divided into three parts, namely training, development, and test sets, each containing data units consisting of 1 to 7 RDF triples.

	Train	Dev	Test	Total
Data units	13,211	1,667	1,779	16,657
Verbalizations	35,426	4,464	5,150	45,040
Avg. verb.	2.68	2.67	2.89	2.70

Table 1: The number of data units (i.e. RDF triple set) and their verbalizations, split over the training, development, and test sets in WebNLG 3.0. For each data unit, several verbalizations have been provided by human annotators. "Avg. verb." refers to the average number of verbalizations per data unit.

The construction of WebNLG-IT, similarly to the development of the Russian version of WebNLG, was carried out in two main steps:

1. Machine translation from English to Italian of WebNLG 3.0, that we describe in Section 3.1;

2. Post-editing of the translated corpus, that we describe in Section 3.2.

3.1 Machine Translation

The translation of WebNLG was performed after evaluating several NMT systems to identify the most performant one. Subsequently, we conducted a human evaluation of the chosen system to ensure an accurate assessment of its performance, allowing us to select it.

3.1.1 Automatic Evaluation of the NMT Systems

We evaluated and compared these four systems:

- OpusMT: multilingual model published in 2020 by the University of Helsinki (Tiedemann and Thottingal, 2020);

- M2M-100: model capable of translating any language pair taken from a set of 100 languages, developed by Facebook AI Research (FAIR) in 2020 (Fan et al., 2020);

- NLLB-200: multilingual model build by META AI in 2020, capable of translating 200 languages (NLLBTeam, 2022);

- DeepL: multilingual model developed by the German company DeepL GmbH.[5]

[5] https://www.deepl.com/translator

The evaluation was conducted by manually translating 400 sentences from English to Italian, which were randomly extracted from the WebNLG corpus.[6] We compared the performance of the four MT systems using four widely-used automatic metrics: chrF2, BLEU, TER, and BERTScore (Jurafsky and Martin, 2024).

For the evaluation, we used MATEO by Vanroy et al. (2023), a platform specialised in Machine Translation, which takes as input the reference and the candidate translations generated and provides an evaluation and comparison between the systems.[7] The evaluation showed that DeepL achieved the best performance on every metric (Table 2).

Model	BERTScore ↑	BLEU ↑	chrF2 ↑	TER ↓
DeepL	**0.93***	**0.67***	**0.84***	**0.25***
OpusMT (b)	0.92	0.58	0.78	0.33
M2M-100	0.90*	0.50*	0.71*	0.40*
NLLB-200	0.90*	0.49*	0.67*	0.44*

Table 2: Results of the MT evaluation on a sample of 400 human translated sentences. The results have been computed using the MATEO platform, where the * indicates a statistically significant difference w.r.t. the baseline (paired bootstrap resampling). The best results are in bold. Upward arrows denote *more is better*, while downward arrows indicate *less is better*. The notation (b) denotes the baseline model OpusMT.

3.1.2 Human Evaluation of Machine Translation

To further assess the systems, we performed a manual evaluation. Specifically, we selected the two models with the highest performance from the previous comparison and evaluated them manually. Following Jurafsky and Martin (2024), our evaluation considers two dimensions: *adequacy* and *fluency*. Adequacy refers to how well the translation preserves the meaning of the source text, while fluency concerns the naturalness and readability of the translation. For each dimension, we used a scale from 1 (low adequacy/fluency) to 5 (high adequacy/fluency). The evaluation was made by two of the authors, both native Italian speakers, on 100 of the 400 candidate translations used in the automatic evaluation. We established guidelines that include assessing when the translation of a

[6] Translations were provided by one of the authors, a native Italian speaker. For each English sentence, the author created a single translation, translating named entities into Italian when necessary to maintain the fluency of the text.

[7] https://mateo.ivdnt.org/

named entity is appropriate, determining when it is acceptable to retain an English term in the translation, and paying close attention to the use of verb tenses as well as singular and plural forms.

	Dimension	$\bar{x}$	M	α	κ
D	Adequacy	4.87 ± 0.46	5	0.90	0.90
	Fluency	4.92 ± 0.35	5	0.80	0.80
O	Adequacy	4.78 ± 0.63	5	0.77	0.77
	Fluency	4.56 ± 0.65	5	0.88	0.88

Table 3: The table displays the mean and median scores for adequacy and fluency, along with the values of Krippendorff's α and Cohen's κ, based on the evaluation of 100 candidate translations generated by DeepL and OpusMT. Both dimensions were rated on a scale from 1 to 5, where 1 indicates low adequacy/fluency and 5 indicates high adequacy/fluency. The first column shows the evaluated systems: "D" indicates DeepL and "O" indicates OpusMT. Symbols used: $\bar{x}$ denotes mean, M denotes median, $\boldsymbol{\alpha}$ is Krippendorff's alpha, and $\boldsymbol{\kappa}$ is Cohen's kappa.

The results show that both systems received high scores in both dimensions, providing fluent and adequate translations. This outcome is consistent across annotators, as reflected by the high values of Krippendorff's α and Cohen's κ (see Table 3). Given DeepL's highest scores in both human and automatic evaluations, we selected it to translate the WebNLG verbalizations from English to Italian.

3.2 Semi-automatic Post-Editing

Once we selected the machine translation model, and automatically translated WebNLG into Italian, we performed two post-editing steps to remove potential errors, namely 1) error detection, 2) error annotation and correction.

3.2.1 Error Detection

In order to detect errors, we initially performed automatic extraction followed by manual review, as illustrated in Figure 2. Specifically, for automatic error detection, we employed the Language Tool Python (LTP) library[8], which automatically identified errors in translated Italian sentences. However, many of these detected errors were false positives, often due to the misclassification of named entities. To mitigate the number of false positives, we utilized the `extend` function from SpaCy[9] to extract named entities from the sentences. These extracted entities were then used to filter out false positives, ensuring that correctly recognized named entities were not erroneously flagged as errors due to LTP's limitations.

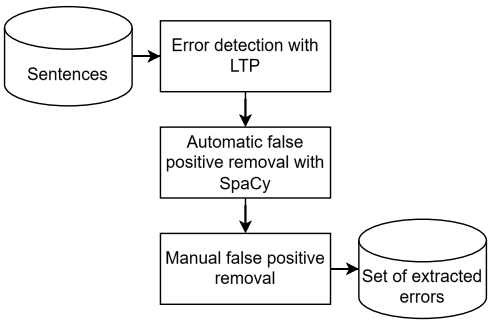

Figure 2: Workflow for error detection in translated Italian sentences.

3.2.2 Error Annotation and Correction

After identifying errors in the translations, they were annotated using a taxonomy specifically developed to label the corrections, referred to as Post-Editing Actions (PEA) (Blain et al., 2011). Within this taxonomy, level 1 indicates the action taken to correct the error, while level 2 provides further details on the nature of the correction, in case level 1 is not self-explanatory. Note that the original taxonomy described in Blain et al. (2011) has several labels not used in our annotation.

The annotations are collected into an XML file within the `<entry>` tag, which contains the error, the PEA, and the level 1 and 2 labels, for example:

```
<entry>
  <error>
    consiglio-amministratore
  </error>
  <pea>
    consiglio amministrativo
  </pea>
  <l1>Noun phrase</l1>
  <l2>Noun stylistic change</l2>
</entry>
```

In total, using this approach, we identified 198 errors, each of which appeared one or more times in the translated corpus. We replaced all these errors in our resource with corresponding corrections, resulting in 3,325 post-editing actions.[10] The majority of these post-editing actions involved modifying noun phrases (3,205 instances), with a stylistic substitution of the noun occurring in 91.28% of the cases involving noun phrases (e.g., *S.r.l.*, *srl*, *Srl*, all

[8] https://pypi.org/project/language-tool-python/

[9] https://spacy.io/

[10] Corrections were provided by one of the authors, a native Italian speaker.

standardized as *S.R.L.*). These edits did not address semantic or grammatical errors, but rather aimed to ensure stylistic consistency across the corpus.

Label	#	%
Noun-phrase	**3,205**	**96.39**
Noun stylistic change	3,035	91.28
Adjective choice	8	0.24
Determiner choice	26	0.78
Case change	2	0.06
Noun meaning choice	134	4.03
Verbal-phrase	**59**	**1.77**
Verb meaning choice	57	1.71
Verb agreement	2	0.06
Preposition change	**17**	**0.51**
Misc style	**35**	**1.05**
Misc	**9**	**0.27**
#Post-editing actions	**3,325**	**100.00**

Table 4: The number of post editing actions performed in the post-editing phase.

4 LLM-based RDF-to-text for WebNLG-IT

In this section, we analyse the results of four distinct experiments. These experiments have been designed with two different goals: (i) providing an evaluation of WebNLG-IT by considering its results for the LLM-based RDF-to-text task, using the original WebNLG as a baseline; (ii) building and releasing the first LLM-based RDF-to-text systems for Italian. With these two aims, we designed four experiments:

1. Fine-tuning of multilingual LLMs on English and Italian versions of WebNLG, with the same subset of data units from the WebNLG 2020 test set (see Section 4.1 and results in Table 5).

2. Fine-tuning of multilingual LLMs on English, Italian, German and Russian WebNLG, using a shared subset of data units for both training and evaluation (see Section 4.2 and results in Table 7).

3. Comparison of LLMs fine-tuned on the WebNLG-IT (see Section 4.3, results in Table 8).

4. Evaluation of a modular generation pipeline for Italian and comparison with fine-tuned LLMs (see Section 4.4, results in Tables 9).

For our experiments, we used an A100 GPU and open-weight LLMs. We utilized both multilingual and Italian LLMs, depending on the experiment. For multilingual models, we employed Llama-3.1-8B-Instruct, Mistral-Nemo-Instruct-2407, and Qwen2.5-7B-Instruct, while for Italian models, we used LLaMAntino-3-ANITA-8B-Inst-DPO-ITA and Minerva-7B-Instruct-v1.0. [11]

All models were fine-tuned for 2 epochs using QLoRa to reduce computational costs. To ensure more robust evaluations, we generated three outputs for each fine-tuned model with a temperature setting of 0.1 on the test sets. For automatic evaluation, we adopted the standard metrics used in the WebNLG challenges (Mille et al., 2024): BLEU (B), chrF++ (C), METEOR (M), and BERTScore F1 with baseline rescaling (BS). Additionally, following the methodology outlined in Dror et al. (2018), we applied the Wilcoxon test to assess statistical significance between models.

4.1 Fine-tuning and Evaluation of Multilingual LLMs on Italian and English WebNLG

In this experiment, we fine-tuned several multilingual LLMs on the English and Italian WebNLG corpora. We used the same training, development, and test set partitioning as in the WebNLG 2020 Challenge (see Table 1).

	Model	B↑	M↑	C↑	BS↑
en	Llama-3.1-8B	0.55*	0.76	0.64	0.77
	Mistral-Nemo	0.54*	0.76	0.64	0.76*
	Qwen2.5-7B	0.52	0.76	0.64	0.76
it	Llama-3.1-8B	0.52*	0.73*	0.62*	0.75*
	Mistral-Nemo	0.54*	0.74*	0.63*	0.76*
	Qwen2.5-7B	0.47	0.71	0.61	0.72

Table 5: Fine-tuning and comparison of multilingual LLMs trained on English and Italian WebNLG. The * denotes a statistically significant difference between an LLM and the subsequent model. See Section 4 for the full names of the LLMs. Evaluation metrics: BLEU (B), METEOR (M), chrF++ (C), and BERTScore F1 with baseline rescaling (BS).

As shown in Table 5, on English WebNLG, the

[11] https://huggingface.co/meta-llama/Llama-3.1-8B,https://huggingface.co/mistralai/Mistral-Nemo-Instruct-2407,https://huggingface.co/Qwen/Qwen2.5-7B-Instruct,https://huggingface.co/swap-uniba/LLaMAntino-3-ANITA-8B-Inst-DPO-ITA,https://huggingface.co/sapienzanlp/Minerva-7B-instruct-v1.0

Llama and Mistral models achieve the closest performance across the metrics, except for the BLEU score, which shows a statistically significant difference between the models. For WebNLG-IT, Mistral emerges as the top-performing model, demonstrating statistically significant differences across all metrics compared to Llama. For each corpus, Qwen has the lowest scores, showing no statistically significant difference only on the METEOR and chrF++ metrics in the fine-tuning on English WebNLG. In conclusion, the models demonstrate good performance across each version of WebNLG. Mistral is the best-performing model among those selected for WebNLG-IT, while both Llama and Mistral achieve similar scores on the English version of the corpus.

4.2 Comparison of the Italian, English, German, and Russian Corpora Using Multilingual LLMs

To assess the quality of the produced resource, we fine-tuned several multilingual LLMs on the Italian, English, German, and Russian WebNLG corpora, followed by a comparison of the resulting automatic evaluation scores. We chose to include the German and Russian corpora due to their silver-standard nature and their construction process, which is similar to that of our resource.

The different dataset cardinalities and the distinct test sets for each corpus made it impossible for us to use the original resource splits to compare the results. First, for each language, we only maintained the common data units. This process involved selecting examples with identical data units. However, we observed that in some cases, data units were similar but differed due to minor variations, such as the presence of an apostrophe or differences in wording for certain elements of the triples. For example:

```
IT: (Aarhus, leader, J_Bundsgaard)
(Aarhus_Airport, cityServed, Aarhus)
RU: (Aarhus, leaderName, J_Bundsgaard)
(Aarhus_Airport, cityServed, Aarhus)
```

We also attempted to merge examples using the key `eid + category + size`. However, in some cases, the data units did not align properly. Given this, we ultimately chose to merge based on exact data unit matches, excluding those that were only similar but not identical.

For each subset of common data units, we allocated 80% to the training set, 10% to the development set, and the remaining 10% to the test set. For each data unit, we examined the number of verbalizations available in the four languages and retained, for each language, the minimum number of occurrences to ensure the same number of verbalizations across the four corpora (Table 6).

	Train	Dev	Test	Total
Data units	3,062	383	383	3,828
Verbalizations	7,998	991	993	9,982
Avg. verb.	2.61	2.58	2.59	2.61

Table 6: This table presents the settings obtained for each corpus.

We used the same multilingual LLMs as in the previous experiment, all of which support Italian, English, German, and Russian. For each corpus, we fine-tuned the three models.

	Model	B↑	M↑	C↑	BS↑
en	Llama-3.1-8B	0.64	0.83	0.71	0.83
	Mistral-Nemo	0.64*	0.83*	0.71*	0.83*
	Qwen2.5-7B	0.60	0.82	0.70	0.80
it	Llama-3.1-8B	0.62	0.80	0.70	0.82
	Mistral-Nemo	0.63*	0.80*	0.70*	0.83*
	Qwen2.5-7B	0.56	0.78	0.68	0.80
ge	Llama-3.1-8B	0.54	0.76	0.64	0.78
	Mistral-Nemo	0.55*	0.76	0.64*	0.78*
	Qwen2.5-7B	0.50	0.75	0.63	0.76
ru	Llama-3.1-8B	0.46	0.71	0.61*	0.77
	Mistral-Nemo	0.48*	0.73*	0.62*	0.78*
	Qwen2.5-7B	0.43	0.70	0.60	0.75

Table 7: Results of multilingual LLMs fine-tuned on Italian, English, German, and Russian WebNLG. An asterisk (*) indicates a significant difference compared to the model listed immediately below it in the table. See Section 4 for the full names of the LLMs. Evaluation metrics: BLEU (B), METEOR (M), chrF++ (C), and BERTScore F1 with baseline rescaling (BS).

Table 7 shows that the Italian and English versions of WebNLG achieve higher scores across all metrics compared to the German and Russian versions. Among the models, Mistral and Llama generally perform best for this task across all languages, while the Qwen model consistently yields lower scores, with a statistically significant difference in most cases. We observe that the Italian corpus achieves the best results on the silver-standard dataset and performs similarly to the English corpus. One possible explanation for these results could be the different NMT models used to create the silver-standard resources. For Italian, we used DeepL, which is based on state-of-the-art neural

architectures, while for German, for instance, a system by Sennrich et al. (2017), based on older neural architectures, was used.

4.3 Comparison between Italian and Multilingual LLMs using WebNLG-IT for Fine-Tuning

This experiment aimed to compare multilingual and Italian LLMs fine-tuned on WebNLG-IT. We adopted the same settings as in the first experiment, using the original training, development, and test set split provided by the WebNLG 2020 Challenge.

Model	B↑	M↑	C↑	BS↑
Mistral-Nemo	0.54*	0.74*	0.63*	0.76*
Llama-3.1-8B	0.52	0.73	0.62	0.75
LLaMAntino-3-8B	0.51*	0.73*	0.62*	0.74*
Qwen2.5-7B	0.47*	0.71*	0.61*	0.72*
Minerva-7B-v1.0	0.44	0.70	0.60	0.71

Table 8: Comparison of multilingual and Italian LLMs fine-tuned on WebNLG-IT. An asterisk (*) indicates a significant difference compared to the following model. See Section 4 for the full names of the LLMs. Evaluation metrics: BLEU (B), METEOR (M), chrF++ (C) and BERTScore F1 with baseline rescaling (BS).

The results in Table 8 show that the Mistral model achieves the highest scores, with statistically significant differences from the other models. This outcome may be attributed to the larger number of parameters in Mistral (Kaplan et al., 2020). Although the Llama model achieves higher scores, it does not show a statistically significant difference compared to the LLaMAntino model. The Qwen and Minerva models have the lowest scores, with Minerva being the least performing model, showing a statistically significant difference compared to the Qwen model. These results are particularly interesting because a multilingual model outperforms models specialized in Italian, as already stated in previous studies (Sarti and Nissim, 2022).

4.4 Assessment of SGA Systems for Italian

As stated in (Kasner and Dušek, 2024), in the RDF-to-text task, long data inputs can present practical challenges, such as the requirement for long-context models and increased GPU memory. Additionally, the generated output may be fluent but inaccurate, potentially containing factual errors or adding/removing pieces of information.

To address this, we implemented an existing modular generation pipeline, the SGA pipeline Oliverio et al. (2024), for Italian, which follows a three-step process: splitting, generation, and aggregation. The method involves dividing longer data units, composed of four or more triples, into smaller units containing three or fewer triples each (e.g., a data unit with five triples is split into two units: one with three triples and the other with two triples), generating verbalizations for each resulting data unit, and aggregating them into more fluent text. The split was made considering the relations between subjects and objects in the data units. For instance, consider the following data unit:

```
Alan_Frew background solo_singer
Alan_Frew genre Rock_music
Rock_music musicFusionGenre
Bhangra_(music)
Rock_music stylisticOrigin Blues
Rock_music stylisticOrigin Folk_music
```

The data unit was split into two sets. The first set includes the first two triples with *Alan_Frew* as the subject, and the second set includes the remaining three triples with *Rock_music* as the subject. For the generation step, we used the fine-tuned models obtained in the experiment described in Section 4.3, while for the sentence aggregation step, we used the base version of these models with the following zero-shot prompt:

```
{"role": "system", "content": "You only
have to paraphrase and aggregate the
given sentences in order to generate a
more fluent Italian text. Do *not* add
any extra information"}
{"role": "user", "content": "Text 1:
..., Text 2: ..., Text 3: ..."}
```

For the evaluation, starting with the classical test set partition provided by the WebNLG 2020 challenge, we considered only the data units with more than three triples, given our focus on long data units, resulting in a test set of 710 data units to verbalize.

Table 9 shows that for each fine-tuned model and its SGA version, the fine-tuned model achieves the best scores on all automatic metrics with statistically significant differences. Especially for BLEU, there is a significant decrease in scores for the SGA systems compared to the fine-tuned ones. Regarding BERTScore, a decrease in scores is also observed, but the gap is smaller compared to the fine-tuned models.

For a more detailed analysis of the results, we manually assessed the generations of Mistral and LLaMAntino to evaluate both a multilingual and an Italian model. In this evaluation, we analyzed the verbalizations produced by the fine-tuned and

Model	B↑	M↑	C↑	BS↑
Mistral-Nemo	0.52*	0.69*	0.61*	0.69*
Mistral-Nemo-SGA	0.39	0.64	0.57	0.59
Llama-3.1-8B	0.50*	0.68*	0.60*	0.68*
Llama-3.1-8B-SGA	0.46	0.67	0.59	0.65
LLaMAntino-3-8B	0.49*	0.68*	0.60*	0.68*
LLaMAntino-3-8B-SGA	0.38	0.62	0.56	0.62
Qwen2.5-7B	0.46*	0.67*	0.60*	0.67*
Qwen2.5-7B-SGA	0.39	0.62	0.56	0.64
Minerva-7B-v1.0	0.43*	0.66*	0.58*	0.65*
Minerva-7B-v1.0-SGA	0.29	0.53	0.48	0.67

Table 9: Comparison of Italian SGA systems with fine-tuned LLMs on the RDF-to-text task. An asterisk (*) indicates a significant difference when compared to the model below. See Section 4 for the full names of the LLMs. Evaluation metrics: BLEU (B), METEOR (M), chrF++ (C), and BERTScore F1 with baseline rescaling (BS).

SGA versions. For each model, we annotated 20 generations.[12] We used the taxonomy from Kasner and Dušek (2024), which classifies errors into four categories: *Incorrect*, where the text contradicts the data; *Not Checkable*, where the information cannot be verified; *Misleading*, where the text is deceptive given the context or the information is missing; and *Other*, for problematic cases not fitting the other categories.

Model	I↓	NC↓	M↓	O↓
LLaMAntino-3-8B	0.10	**0.00**	0.25	0.18
LLaMAntino-3-8B-SGA	**0.00**	0.10	0.20	**0.05**
Mistral-Nemo	0.10	**0.00**	**0.18**	0.10
Mistral-Nemo-SGA	0.05	0.05	0.25	**0.05**

Table 10: The table presents the average scores assigned by annotators for each label and model, along with the average agreement metrics. Symbols used: *I* (Incorrect), *NC* (Not Checkable), *M* (Misleading) and *O* (Other).

The human evaluation shows that, in the evaluated sample, the SGA systems achieve the best scores for the Incorrect and Other labels. Additionally, the use of the SGA pipeline improves the performance of both models across all dimensions, except for the Misleading label in Mistral-Nemo. The average agreement between the two evaluators is 0.96 for Krippendorff's alpha and 0.96 for Cohen's kappa. In these manual assessments, we observed that sometimes the SGA systems produced excessively verbose text, maintaining the correct meaning of the sentence without adding or omitting any information. For example, given this segment generated by the LLaMAntino-based SGA system:

> *"...La contea in cui si trova il monumento, Adams, si distingue da altre come ad esempio la contea di Carroll, situata in Maryland, a sud-est di Adams in Pennsylvania."* (Translation: *"...The county in which the monument is located, Adams, stands out from others such as Carroll County, located in Maryland, southeast of Adams in Pennsylvania."*)

and its reference:

> *"...A sud-est della Contea di Adams, Pennsylvania, si trova la Contea di Carroll, Maryland."* (Translation: *"...Southeast of Adams County, Pennsylvania, lies Carroll County, Maryland."*)

The system's generation is adequate and fluent, but it is excessively verbose. As a result, this generation is penalized (for instance, the BLEU score of the full generation is 0.15). This verbosity arises from the sentence aggregation step. Future work to improve this approach could involve exploring different prompting strategies (e.g., few-shot learning) or fine-tuning the models used in the aggregation step for sentence aggregation tasks, potentially by extracting datasets from the WebNLG-IT corpus. These improvements could enhance the performance of this pipeline and help mitigate the verbosity issue.

5 Conclusion

The main objective of this work was the creation of the WebNLG-IT corpus. To achieve this goal, we used machine translation techniques to translate the corpus from English to Italian, followed by post-editing to enhance the quality of the produced resource. For the machine translation, we chose the DeepL model after a comparison through automatic metrics with other translation models, followed by an automatic evaluation phase to assess the output produced by the chosen model. The post-editing process involved a semi-automatic approach, consisting of the automatic extraction of errors and their manual correction.

To demonstrate the quality of the produced resource, we tested various Large Language Models, showing that our resource enables the models to achieve good performance in RDF-to-text generation. Finally, we implemented a natural language generation pipeline that integrates LLMs, but it did not achieve the desired results. Future work on this generation pipeline could involve the use of fine-tuning for the sentence aggregation step.

[12]The evaluation was conducted by two of the authors, both native Italian speakers.

6 Limitations

We believe that this corpus can represent a useful resource in supporting the development of models and studies for Italian; however, we are also aware that this corpus is a silver resource with synthetic data, which would benefit from human evaluation to further validate and correct any potential errors. Another limitation lies in the experimentation, which could be more extensive by exploring other possibilities not undertaken here due to computational resource constraints. Additionally, the human assessments of the NMT and SGA systems are based on a limited sample of examples, but they were useful in highlighting potential types of errors. Lastly, the experiment described in Section 3.1.1 was conducted using manual translations produced by only one of the authors. As a result, the evaluation is based on a single ground truth, which does not take into account the natural variation that would arise if multiple humans, potentially with different backgrounds or levels of expertise, had carried out the task.

7 Ethical Considerations

The human evaluations, crucial for validating the content of the automatic translation (Section 3.1.2), were carried out by two authors of this paper. Moreover, the release of our corpus permits its use for scientific purposes and it is released under the Creative Commons Attribution Non Commercial 4.0 International license.

References

Frédéric Blain, Jean Senellart, Holger Schwenk, Mirko Plitt, and Johann Roturier. 2011. Qualitative analysis of post-editing for high quality machine translation. In *Asia-Pacific Association for Machine Translation*.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina, editors. 2020. *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*. Association for Computational Linguistics, Dublin, Ireland (Virtual).

Liam Cripwell, Anya Belz, Claire Gardent, Albert Gatt, Claudia Borg, Marthese Borg, John Judge, Michela Lorandi, Anna Nikiforovskaya, and William Soto Martinez. 2023. The 2023 webnlg shared task on low resource languages. overview and evaluation results (webnlg 2023). In *Association for Computational Linguistics*, page 55–66.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. In *Journal of Machine Learning Research*, pages 1–48.

Thiago Castro Ferreira, Diego Moussallem, Emiel Krahmer, and Sander Wubben. 2018. Enriching the webnlg corpus. In *Association for Computational Linguistics*, page 171–176.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from rdf data. In *Association for Computational Linguistics*, pages 124–133.

Daniel Jurafsky and James H. Martin. 2024. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd edition. Online manuscript released August 20, 2024, pages 263–285.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Zdeněk Kasner and Ondřej Dušek. 2024. Beyond reference-based metrics: Analyzing behaviors of open llms on data-to-text generation. *Preprint*, arXiv:2401.10186.

V.I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions and reversals. In *Proceedings of the USSR Academy of Sciences*, page 845–848.

Simon Mille, João Sedoc, Yixin Liu, Elizabeth Clark, Agnes Johanna Axelsson, Miruna Adriana Clinciu, Yufang Hou, Saad Mahamood, Ishmael Nyunya Obonyo, and Lining Zhang. 2024. The 2024 GEM shared task on multilingual data-to-text generation and summarization: Overview and preliminary results. In *Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges*, pages 17–38, Tokyo, Japan. Association for Computational Linguistics.

NLLBTeam. 2022. No language left behind: Scaling human-centered machine translation.

Michael Oliverio, Pier Felice Balestrucci, Alessandro Mazzei, and Valerio Basile. 2024. Dipinfo-unito at the gem'24 data-to-text task: Augmenting llms

with the split-generate-aggregate pipeline. In *In Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges*.

Laura Perez-Beltrachini, Rania Sayed, and Claire Gardent. 2016. Building rdf content for data-to-text generation. In *The COLING 2016 Organizing Committee*, pages 1493–1502.

Gabriele Sarti and Malvina Nissim. 2022. It5: Text-to-text pretraining for italian language understanding and generation.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural mt systems for wmt17. In *Association for Computational Linguistics*, page 389–399.

Anastasia Shimorina, Elena Khasanova, and Claire Gardent. 2019. Creating a corpus for russian data-to-text generation using neural machine translation and post-editing. In *Association for Computational Linguistics*, pages 44–49.

Jörg Tiedemann and Santhosh Thottingal. 2020. Opus-mt – building open translation services for the world. In *European Association for Machine Translation*, pages 479–480.

Bram Vanroy, Arda Tezcan, and Lieve Macken. 2023. Mateo: Machine translation evaluation online. In *European Association for Machine Translation*, pages 499–500.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Experimental Setup

Parameter	Value
QLoRA parameters	
LoRA attention dimension	64
Alpha parameter	16
Dropout probability	0.1
bitsandbytes parameters	
Activate 4-bit precision	True
Compute dtype for 4-bit	float16
Quantization type	nf4
Activate nested quantization	False
TrainingArguments parameters	
Number of training epochs	2
Enable fp16 training	False
Enable bf16 training	True
Batch size per GPU for training	4
Batch size per GPU for evaluation	4
Gradient accumulation steps	1
Maximum gradient norm	0.3
Initial learning rate	2e-4
Weight decay	0.001
Optimizer	p_adamw_32bit
Learning rate schedul	cosine
Warmup ratio	0.03

Table 11: Hyperparameters used in the experiments.

LLMs are trained on an A100 GPU using the Huggingface Transformers library (Wolf et al., 2020). We used multilingual and Italian models, and all these are fine-tuned with the same hyperparameters. Table 11 shows the common hyperparameter configuration used for the fine-tuning of the models. Table 12 shows the number of parameters for each used model. For all the fine-tuning on WebNLG performed in the various experiments, we always used the same prompt:

```
"<s> [INST] Given the following triples in (TRIPLE),
you have to generate the corresponding text in (ANW)
[/INST] [TRIPLE] ... [/TRIPLE] [ANW] ... [/ANW]
</s>"
```

Using these models, we conducted four experiments. The first (Section 4.1) took approximately 25 hours for fine-tuning 6 models and 27 hours for 18 generations of the test set (3 per fine-tuned model). The second experiment (Section 4.2), with fewer examples in the datasets, took 6 hours for fine-tuning (12 fine-tunings) and 9 hours for the generations (36 generations of the test set). The third experiment (Section 4.3) used the fine-tuned multilingual model obtained in the first experiment, so for this experiment, we only fine-tuned two Italian LLMs, which took 8 hours. The generations for these two models took 9 hours (6 generations of the test set, three per LLM). In the final experiment, we used the fine-tuned models obtained from experiments 1 and 3. We only needed to perform the generation for the SGA, which took 8 hours for data-to-text generation (15 generations of the test set) and 8 hours for the sentence aggregation of the obtained generations. In conclusion, we spent approximately 100 hours on fine-tuning and generations. All the results shown in the tables are rounded up.

Model	#param.
Llama-3.1-8B-Instruct	8B
Mistral-Nemo-Instruct-2407	12B
Qwen2.5-7B-Instruct	7B
LLaMAntino-3-ANITA-8B-Inst-DPO-ITA	8B
Minerva-7B-Instruct-v1.0	7B

Table 12: This table shows the number of parameters of the models used in the four experiments.

B Labels used in the error annotation process

In Table 13, we present the labels used in our error annotation process described in Section 3.2.2, derived from the taxonomy of Blain et al. (2011).

Level 1	Level 2
Noun-phrase	Determiner choice Noun meaning choice Case change Noun stylistic change Adjective choice
Verbal-phrase	Verb agreement Verb meaning choice
Preposition change	
Misc style	
Misc	

Table 13: Labels used for classifying the Post Editing Actions (PEA).