# Document-Level Knowledge Graph to Text Generation

**Anonymous ACL submission**

## Abstract

Knowledge graph-to-text generation has traditionally been evaluated on sentence-level benchmarks, where recent LLMs achieve strong reference-based scores. However, these evaluations provide limited evidence about whether long-form generations remain faithful to larger input graphs. We introduce LAGRANGE$_{\text{DOC}}$ and WikiDocKG, two document-level KG-to-text benchmarks designed to evaluate grounded long-form generation. Across prompting, fact-by-fact verbalization, factuality-guided reranking, and QLoRA fine-tuning, we find a systematic divergence between reference similarity and graph-grounded factual consistency. Fine-tuning improves BLEU, BERTScore, and BLEURT, but can reduce factual consistency when references contain information not supported by the input graph. In contrast, fact-by-fact generation and factuality-guided candidate selection improve graph faithfulness, often at the cost of reference similarity. These findings motivate evaluation protocols that jointly measure reference adequacy and graph-grounded factual consistency.[1]

## 1 Introduction

Knowledge graph-to-text (KG-to-text) generation aims to produce natural language descriptions that accurately express the information contained in an input knowledge graph. Previous work has primarily studied sentence-level generation, where the input consists of a small set of triples and the output is one or a few sentences. In this setting, systems are commonly evaluated using reference-based metrics, including lexical overlap and learned semantic similarity metrics such as BERTScore and BLEURT (Zhao et al., 2024; Yuan and Färber, 2024; Zhao et al., 2023; Ke et al., 2021). However, these metrics compare a system output to a reference text and provide limited evidence that the generation is factually supported by the input graph. This limitation becomes more prominent in document-level KG-to-text generation, which requires selecting relevant facts from a larger graph, ordering them across multiple sentences, aggregating related information, and maintaining factual consistency throughout the document. These requirements increase the risk of hallucinated claims and omitted input facts. At the same time, fluent long-form LLM outputs may receive high reference-similarity scores even when they are only weakly grounded in the KG.

In this paper, we study document-level KG-to-text generation through a systematic evaluation of both reference similarity and factual consistency with respect to the input graph. Our experiments cover four Wikipedia-derived common knowledge datasets of varying scale and complexity. To enable evaluation in this setting, we introduce two document-level KG-to-text datasets, LAGRANGE$_{\text{DOC}}$ and WIKIDOCKG, which challenge models to generate long-form text from large knowledge graph inputs.

We evaluate several LLM-based generation strategies, including direct and chain-of-thought prompting over the full graph, fact-by-fact verbalization followed by document composition, and QLoRA-based parameter-efficient fine-tuning. We further introduce FactSpotter-Guided Consistency (FGC), an inference-time candidate selection method that uses factual consistency estimates to prefer outputs better supported by the input KG.

Our results show that reference-based evaluation can misrepresent factual consistency in document-level KG-to-text generation. Improvements in reference similarity sometimes coincide with degraded graph alignment, particularly when references are noisy or incomplete. In contrast, fact-by-fact verbalization and FGC produce outputs that are more faithful to the input KG, but not always more similar to the references. These results highlight the

[1] Our data and code will be available at [anonymized].

need for evaluation protocols that jointly account for reference matching and graph-grounded factual consistency.

The main contributions of this work are:

- We introduce LAGRANGE$_{\text{DOC}}$ and WIKIDOCKG, two document-level KG-to-text datasets with large KG inputs.
- We provide a systematic comparison of prompting, fact-by-fact generation, factuality-guided reranking, and QLoRA fine-tuning across sentence- and document-level KG-to-text datasets.
- We show that reference-based metrics can favor less graph-faithful outputs, highlighting the need for balanced multi-perspective evaluation.

## 2 Related Work

### 2.1 Benchmarks

Current KG-to-text benchmarks primarily target short-form realization or restricted graph domains. Common knowledge benchmars, such as WebNLG (Gardent et al., 2017a) and DART (Nan et al., 2021), offer small-scale annotations pairing compact KGs and corresponding sentences. Data for scientific abstract generation, such as AGENDA (Koncel-Kedziorski et al., 2019), employ a limited set of predicates and hence often suffer from incomplete coverage of the target paragraph. WikiGraphs (Wang et al., 2021) provides a large-scale alignment between Freebase (Bollacker et al., 2008) and Wikipedia articles, but does not include explicit triple-to-text alignments. Event-Narrative (Colas et al., 2021) offers another large KG-to-text resource, but restricts triples to event entities mentioned in the associated text, limiting its use for general-purpose graph-to-text generation. More recently, Mousavi et al. (2024) introduced a large graph–text aligned LAGRANGE based on cyclic evaluation and alignment heuristics; however, its annotations remain sentence-level.

These resources leave a gap for document-level benchmarks that pair large, general-purpose KGs with long-form textual targets. Our work addresses this gap introducing Wikipedia-derived datasets for grounded document-level KG-to-text generation.

### 2.2 Methods and Evaluation

KG-to-text generation have evolved from sequence-to-sequence models over linearized triples (Domingo et al., 2020; Gardent et al., 2017b) and graph-aware encoders (Koncel-Kedziorski et al., 2019; Marcheggiani and Perez-Beltrachini, 2018) to commonly pre-trained and graph-structured language models (Chen et al., 2020; Ke et al., 2021; Colas et al., 2022). Recent work further studies low-resource, zero-shot, and multilingual settings (Song and Gardent, 2025; Song et al., 2025; He et al., 2025).

Evaluation commonly relies on reference-based metrics such as BLEU, METEOR, chrF++, TER, and BERTScore (Gardent et al., 2017a; Castro Ferreira et al., 2020). However, reference similarity may not capture whether generated text is faithful to the input graph. Recent work therefore emphasizes semantic adequacy, factual correctness, data coverage, and faithfulness in addition to fluency (Dušek and Kasner, 2020; Thomson and Reiter, 2020; Iarosh et al., 2025). This concern is especially relevant for LLM-based KG-to-text systems, which may produce fluent but weakly grounded outputs (He et al., 2025).

In this work, we evaluate LLM-based KG-to-text generation across multiple prompting and adaptation regimes, including direct prompting, chain-of-thought prompting, and QLoRA-based fine-tuning. We conduct experiments on sentence-level and document-level Wikidata-derived benchmarks assessing both reference similarity and factual consistency with the input graph.

## 3 Data

We evaluate document-level general knowledge graph-to-text generation on both standard benchmarks and newly constructed datasets. Existing resources remain largely sentence-centric: WebNLG (Gardent et al., 2017a) contains 1–7 triples per graph, with an average of 2.96, while LAGRANGE (Mousavi et al., 2024) contains 4.02 triples per instance on average. To study generation from larger, document-level graphs, we introduce LAGRANGE$_{\text{DOC}}$, compiled from LAGRANGE, with 4–20 triples per graph, and WIKIDOCKG, a new dataset with substantially larger graphs containing 10–120 triples. These datasets support the study of discourse planning, long-context modeling, and evidence grounding.

**WebNLG** WebNLG is a standard KG-to-text and text-to-KG benchmark consisting of RDF triples, primarily extracted from DBpedia, paired with human-authored verbalizations. It contains approx-

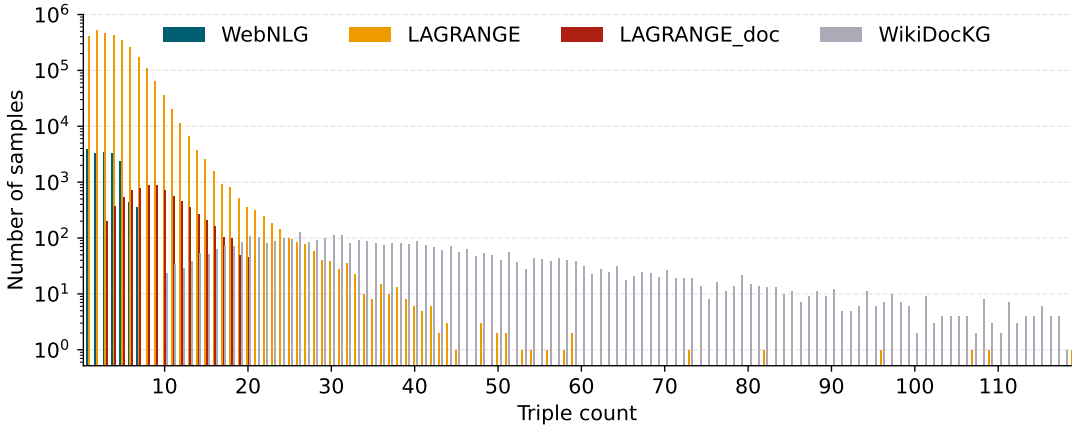



Figure 1: Distribution of graph sizes across KG-to-text datasets.

imately 25K data–text pairs across diverse domains. Although WebNLG provides high-quality alignments, its small graph size and sentence-level scope limit its applicability to document-level generation.

**LAGRANGE** The recent LAGRANGE dataset was introduced to substantially expand annotated data for this task. It aligns WikiData triples with Wikipedia-derived sentences using heuristic filtering to improve semantic consistency across modalities. Compared with earlier automatically constructed large-scale Wikipedia-to-RDF datasets, such as TeKGen (Agarwal et al., 2021) and T-REx (Elsahar et al., 2018), LAGRANGE provides stronger data–text alignment. We use it as a larger-scale but noisier counterpart to WebNLG, comprising over 3 million sentence-level instances.

Figure 1 shows the distribution of graph sizes across datasets. The construction of the document-level datasets is described below.

### 3.1 Constructing LAGRANGE$_{\text{DOC}}$

To study document-level KG-to-text generation, we construct LAGRANGE$_{\text{DOC}}$ from the full LAGRANGE dataset. LAGRANGE often contains multiple sentence-level instances from the opening paragraph of the same Wikipedia article; we merge these instances into document-level examples, each pairing a knowledge graph with the ordered sequence of sentences aligned to its triples.

Construction begins by resolving source titles to Wikipedia pages. Since LAGRANGE triples do not include explicit Wikidata identifiers, we retrieve candidate pages using title-based Wikipedia search and rank them by sentence-level string similarity. We preserve the original train/test partition when forming the document-level dataset.

The original LAGRANGE annotations preserve source sentences but resolve anaphora to make each sentence self-contained. To recover coherent discourse, we align annotated sentences to their corresponding Wikipedia sentences and restore their original order. We first align sentences using edit distance,[2] and then apply semantic validation with an off-the-shelf NLI model[3] to filter lexically similar but unsupported matches.[4] The first sentence and all unsupported matches are retained in their original LAGRANGE-annotated form.

We retain matched pages with at least three annotated sentences, comprising approximately 12% of the original LAGRANGE dataset. We then apply graph-size filtering, retaining only documents whose merged graphs contain 4–20 edges to exclude both under-annotated examples and outliers. The resulting dataset contains 6,882 documents aligned with 24,255 sentences and 68,447 knowledge triples, with a median of 9 triples per document.

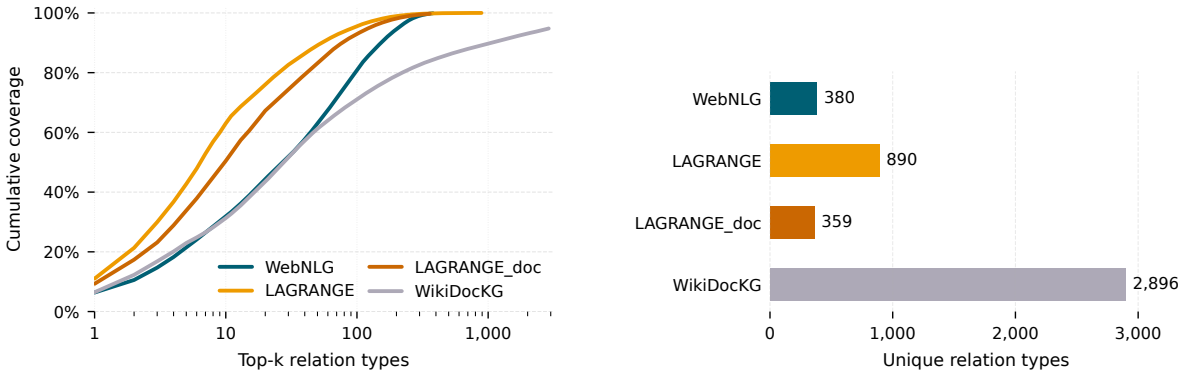
### 3.2 Constructing WIKIDOCKG

To move beyond the first-paragraph scope of LAGRANGE, we construct WIKIDOCKG, a document-level benchmark pairing Wikipedia articles with Wikidata-aligned knowledge graphs. We sample 4,200 Wikipedia pages containing at least four textual paragraphs from the March 2026 dump. To reduce overlap with possible pretraining data, we restrict the corpus to pages created after February 1, 2026, according to Wikipedia page creation

[2] As implemented in the `rapidfuzz` library.

[3] `MoritzLaurer/mDeBERTa-v3-base-xnli-multilingual-nli-2mil7`.

[4] We set a normalized edit distance threshold of 15% and an NLI contradiction threshold of 30%.




(a) Relation concentration.

(b) Relation vocabulary size.

Figure 2: Distribution of the predicates appearing at least three times in the corresponding dataset.

logs.

We extract document-level knowledge graphs using the state-of-the-art KG extraction framework (Chepurova et al., 2026).[5] After initial filtering, the dataset contains 4,096 documents with 10–120 triples per graph (median: 35) and 3–50 sentences per document (median: 15).

We further improve text–graph consistency through constrained sentence filtering. Sentences are first ranked by lexical overlap with triple predicates and objects. We then greedily test sentence removals using a FactSpotter entailment reranker. A sentence is removed only if its deletion preserves the entailment mask of the associated triples. This process removes redundant or weakly grounded content while maintaining coverage of the extracted graph. The resulting corpus contains 4,091 annotated documents with 3–49 sentences per document (median: 14). We randomly select 500 samples for the test set.[6]

Figure 2 illustrates predicate distributions across datasets. Unlike WebNLG, LAGRANGE, and LAGRANGE$_{\text{DOC}}$, which are dominated by a relatively small set of frequent predicates, WIKIDOCKG exhibits a broader and more long-tailed relation vocabulary. Models evaluated on WIKIDOCKG must therefore generalize across a larger and less concentrated set of relations.

## 4 Methods

We first study whether off-the-shelf LLMs can solve the task using prompting alone, without task-specific fine-tuning. We evaluate **Direct** prompting and Chain-of-Thought (**CoT**) prompting (Wei et al., 2022) in both zero-shot and few-shot regimes. Since factual consistency depends critically on grounding generated claims in evidence, we further propose two inference-time evidence-aware baselines: FactSpotter-Guided Consistency (FGC) and Fact-by-Fact generation (FBF). Both methods are designed to enforce alignment between model outputs and the provided evidence.

**FactSpotter-Guided Consistency (FGC)** Given an input graph, we generate $n$ candidate outputs using beam search.[7] For each candidate, we extract the final answer, excluding intermediate reasoning if present, and compute its FactSpotter (Zhang et al., 2023) score against the reference graph. We select the highest-scoring candidate as the final prediction. FGC follows prior work on selecting among multiple generations, including consensus-based selection (Borgeaud and Emerson, 2020), Self-Consistency (Wang et al., 2023), and MBR decoding (Freitag et al., 2022), but uses graph-to-text factual consistency rather than surface similarity, answer frequency, or learned utility.

**FBF** Another intuitive solution for enforcing fact consistency in document generation is to implement is a two-stage direct Fact-by-Fact prompting pipeline. In the first stage, each knowledge graph triplet is independently realized as a simple factual sentence using a constrained prompt that preserves all entities and relations. In the second stage, the resulting sentences, if there are more than one, are aggregated and passed to a language model for discourse naturalization, which produces a single fluent paragraph by reducing redundancy and in-

[5] Further details are provided in Appendix A.

[6] Additional split diagnostics are provided in Appendix A.2.

[7] We set $n = 5$ in our experiments.

troducing appropriate referring expressions while maintaining strict factual fidelity. This decomposition separates content realization from surface fluency and offers greater control over both over- and under-generation of facts. The approach is conceptually related to the two-stage data-to-text generation methods of Moryossef et al. (2019) and Kasner and Dusek (2022), but unlike these prior approaches, which rely on trainable models and explicit content planning, both stages of our FBF baseline are implemented via LLM prompting.

**QLoRA** This setting applies Low-Rank Adaptation (Hu et al., 2021) to 4-bit quantized models to fine-tune on the train part of the dataset. Full hyperparameter details are provided in Appendix E.

**GAP** We additionally evaluate GAP (Colas et al., 2022) as a strong BART-based baseline.

## 5 Experimental Setup

### 5.1 Models

We evaluate a set of language models spanning both open-weight and proprietary across multiple parameter scales: `Qwen3.5-9B`, `Qwen3.5-27B`, `Llama-3.3-70B`, and `GPT-5.1`. This model suite allows us to compare performance under realistic resource-constrained settings as well as higher-capacity generation settings. For open-weight models, we evaluate each configuration over three independent runs with temperature set to 0.7 and report the mean and standard deviation. For the proprietary model, we perform a single evaluation run with temperature 0 in order to minimize decoding variability while keeping API cost manageable.

Document-level KG-to-text generation remains computationally intensive at larger model scales. In our experiments, QLoRA fine-tuning of `Llama-3.3-70B` on WikiDocKG was not reliable on a 2×40GB GPU setup because of memory constraints. We therefore exclude this configuration from the corresponding result tables. Section 6 discusses the observed relations between model size and automatically evaluated KG-to-text generation performance in detail.

### 5.2 Diversity-Preserving Subset Construction

Training and evaluating graph-to-text models at scale is computationally expensive, particularly when using parameter-efficient fine-tuning and proprietary LLM APIs. Recent work has shown that large NLP datasets often exhibit substantial redundancy, suggesting that carefully constructed subsets can retain most of the relevant signal while significantly reducing cost (Li and Xiong, 2025; Wang et al., 2026). While clustering-based selection does not explicitly optimize for model ranking fidelity, prior work shows that redundancy reduction is a key factor enabling accurate evaluation on compressed subsets (Wang et al., 2026). Our approach provides a simple and reproducible approximation to this principle without introducing model-dependent bias. We adopt $k-$means-clustering as a simple, model-agnostic approximation to redundancy-aware subset selection, following the procedure detailed in Appendix B.

For training, we apply this procedure to the LAGRANGE dataset, which is substantially larger than the other benchmarks and would otherwise be prohibitively expensive to use across extensive experimental runs. This yields a subset of 50,000 instances while maintaining diversity in graph size, connectivity, and structural patterns. For the remaining datasets, we retain the full training splits.

For evaluation, we construct a shared subset of 500 examples from each test split using the same clustering-based procedure. All models are evaluated on this identical subset to ensure comparability under a fixed budget. By selecting examples from across clusters, the resulting subset preserves the distribution of graph complexity present in the original test data. We fix all random seeds and release the subset indices to ensure reproducibility.

### 5.3 Metrics

Multiple standard metrics are employed to capture different aspects of document generation performance. Micro-averaged **BLEU** (Papineni et al., 2002) calculates modified n-gram overlap between a generated output and reference text, with a brevity penalty to prevent artificially short outputs from scoring highly. The other metrics are standartly macro-averaged across test samples. **BERTScore** (Zhang et al., 2020) computes cosine similarity between contextual embeddings of tokens in the candidate and reference, producing F1 measure reflecting semantic alignment. **BLEURT** (Sellam et al., 2020) leverages a pre-trained transformer additionally fine-tuned on human ratings and synthetic perturbations to better estimate semantic similarity. **AlignScore**[8] (Zha et al., 2023) is a trained

[8] We employ the checkpoint based on `RoBERTa-base`.

| Method Family | Setting | Lexical | Semantic Similarity | | Factual Consistency | | Avg |
|---|---|---|---|---|---|---|---|
| | | BLEU | BERTScore | BLEURT | AlignScore | FactSpotter | |
| **WebNLG** | | | | | | | |
| GAP | BART-base | 29.38 ± 0.15 | 93.55 ± 0.04 | 73.77 ± 0.12 | 83.44 ± 0.15 | 83.12 ± 0.43 | 74.46 ± 0.15 |
| FBF | 1st stage | 23.46 ± 0.22 | 92.57 ± 0.01 | 73.04 ± 0.03 | **96.41** ± 0.29 | 97.06 ± 0.22 | 78.32 ± 0.09 |
| | 2nd stage | 22.62 ± 0.10 | 92.64 ± 0.03 | 72.95 ± 0.21 | 91.93 ± 0.57 | 96.21 ± 0.37 | 77.17 ± 0.07 |
| Direct | 0-shot | 28.04 ± 0.43 | 93.78 ± 0.02 | 76.48 ± 0.09 | 83.56 ± 0.42 | 97.00 ± 0.13 | 77.25 ± 0.13 |
| | 0-shot + FGC | 27.09 ± 0.11 | 93.66 ± 0.03 | 76.22 ± 0.02 | 83.62 ± 0.38 | 98.73 ± 0.23 | 77.42 ± 0.03 |
| | 5-shot | 29.75 ± 0.49 | 94.00 ± 0.03 | 76.65 ± 0.10 | 89.21 ± 0.52 | 98.25 ± 0.30 | 79.10 ± 0.07 |
| | 5-shot + FGC | 29.09 ± 0.10 | 93.96 ± 0.05 | 76.60 ± 0.12 | 90.74 ± 0.37 | **99.77** ± 0.03 | 79.63 ± 0.09 |
| CoT | 0-shot | 28.25 ± 0.85 | 93.70 ± 0.09 | 75.89 ± 0.16 | 88.19 ± 0.23 | 97.15 ± 0.38 | 78.13 ± 0.12 |
| | 0-shot + FGC | 27.37 ± 0.69 | 93.72 ± 0.11 | 76.12 ± 0.22 | 89.97 ± 0.24 | 99.45 ± 0.23 | 78.85 ± 0.30 |
| | 5-shot | 30.21 ± 0.24 | 94.08 ± 0.06 | 76.61 ± 0.24 | 91.10 ± 0.46 | 98.53 ± 0.29 | 79.61 ± 0.25 |
| | 5-shot + FGC | 28.43 ± 1.34 | 93.93 ± 0.04 | 76.44 ± 0.09 | 92.05 ± 0.43 | 99.77 ± 0.04 | 79.83 ± 0.18 |
| QLoRA | default | 32.64 ± 0.71 | **94.32** ± 0.04 | 77.02 ± 0.17 | 88.12 ± 1.47 | 96.32 ± 0.33 | 79.12 ± 0.30 |
| | default + FGC | **32.58** ± 0.63 | **94.32** ± 0.04 | **77.18** ± 0.18 | 89.44 ± 0.69 | 99.17 ± 0.30 | **79.87** ± 0.06 |
| **LAGRANGE** | | | | | | | |
| GAP | BART-base | 41.09 ± 0.87 | **94.37** ± 0.14 | **69.44** ± 0.30 | 42.97 ± 1.56 | 77.93 ± 0.45 | 68.52 ± 0.28 |
| FBF | 1st stage | 11.18 ± 0.04 | 88.35 ± 0.03 | 58.44 ± 0.04 | **96.84** ± 0.09 | 94.46 ± 0.05 | 72.23 ± 0.03 |
| | 2nd stage | 12.38 ± 0.13 | 88.98 ± 0.03 | 58.35 ± 0.11 | 92.60 ± 0.86 | 91.75 ± 0.51 | 71.54 ± 0.31 |
| Direct | 0-shot | 23.02 ± 0.25 | 91.43 ± 0.02 | 64.08 ± 0.20 | 87.95 ± 0.50 | 85.48 ± 0.20 | 72.63 ± 0.13 |
| | 0-shot + FGC | 21.05 ± 0.34 | 91.13 ± 0.03 | 63.40 ± 0.11 | 88.08 ± 0.29 | 91.87 ± 0.26 | 73.55 ± 0.04 |
| | 5-shot | 26.50 ± 2.62 | 92.07 ± 0.24 | 65.17 ± 0.48 | 86.97 ± 1.49 | 84.68 ± 0.97 | 73.34 ± 0.52 |
| | 5-shot + FGC | 26.72 ± 2.66 | 91.86 ± 0.32 | 64.90 ± 0.55 | 87.44 ± 1.48 | 89.82 ± 1.41 | **74.21** ± 0.43 |
| CoT | 0-shot | 18.21 ± 1.00 | 90.88 ± 0.06 | 62.62 ± 0.23 | 91.47 ± 0.69 | 88.20 ± 0.89 | 72.92 ± 0.31 |
| | 0-shot + FGC | 13.96 ± 0.69 | 90.28 ± 0.09 | 61.52 ± 0.38 | 91.99 ± 0.67 | **94.71** ± 0.14 | 73.57 ± 0.23 |
| | 5-shot | 16.39 ± 3.39 | 91.71 ± 0.24 | 64.40 ± 0.31 | 88.49 ± 0.89 | 84.96 ± 0.26 | 73.32 ± 0.45 |
| | 5-shot + FGC | 8.32 ± 2.05 | 90.54 ± 0.48 | 62.71 ± 0.72 | 85.69 ± 1.46 | 91.09 ± 0.29 | 72.63 ± 0.69 |
| QLoRA | default | 40.30 ± 0.86 | 93.96 ± 0.26 | 68.69 ± 1.33 | 32.32 ± 2.66 | 78.18 ± 1.40 | 65.59 ± 1.01 |
| | default + FGC | **41.35** ± 0.98 | 93.98 ± 0.27 | 68.67 ± 1.25 | 31.77 ± 2.71 | 83.88 ± 1.54 | 66.65 ± 1.00 |

Table 1: Detailed results for the `Qwen-3.5-9B` on sentence-level datasets; mean ± std over 3 runs.

factual consistency metric based on a unified alignment function trained on millions of examples from tasks including NLI, QA, and semantic similarity, enabling evaluation of factual agreement between two texts; scores are probabilities of entailment ranging from 0 to 1. **FactSpotter**[9] (Zhang et al., 2023) focuses on factual faithfulness in graph-to-text tasks by estimating how well generated text reflects the facts encoded in a reference knowledge graph, showing strong agreement with human assessments of data correctness and coverage.

To aggregate performance across heterogeneous metrics, we compute a normalized **Avg** score. For each metric, we apply min–max normalization, mapping raw scores to the $[0, 1]$ range based on the minimum and maximum values observed per example across all experiments.[10] The final score for each system is then obtained by averaging the normalized values across all metrics.

[9] `Inria-CEDAR/FactSpotter-DeBERTaV3-Base`

[10] The normalization statistics are reported in Appendix C.

## 6 Results

We first benchmark all described approaches using a smaller `Qwen-3.5-9B` model (Table 1) to identify the relative strengths and weaknesses of each method across datasets. We then evaluate the full suite of models using the top-performing setting from each family (Table 3).

The evaluation metrics are organized into three major groups: lexical overlap (BLEU), semantic similarity (BERTScore and BLEURT), and factual consistency (AlignScore and FactSpotter). This grouping is important, since the results show that high reference similarity does not necessarily correspond to strong factual grounding.

**Fact-by-fact verbalization yields the strongest factual consistency.** FBF achieves the highest factual consistency by verbalizing input triplets independently. Adding a rewriting step does not consistently improve semantic similarity and may lower factual consistency estimates, especially un-

| Method Family | Setting | Lexical | Semantic Similarity | | Factual Consistency | | Avg |
|---|---|---|---|---|---|---|---|
| | | BLEU | BERTScore | BLEURT | AlignScore | FactSpotter | |
| | | | | **LAGRANGE$_{DOC}$** | | | |
| GAP | BART-base | 23.85 ± 0.30 | 91.36 ± 0.12 | 55.58 ± 0.37 | 37.84 ± 0.44 | 71.12 ± 0.61 | 60.56 ± 0.20 |
| FBF | 1st stage | 12.24 ± 0.01 | 86.76 ± 0.02 | 49.98 ± 0.07 | **95.59** ± 0.03 | **92.17** ± 0.48 | 70.00 ± 0.11 |
| | 2nd stage | 11.87 ± 0.16 | 87.00 ± 0.02 | 49.03 ± 0.10 | 89.83 ± 0.59 | 89.77 ± 0.25 | 68.36 ± 0.12 |
| Direct | 0-shot | 16.54 ± 0.13 | 90.29 ± 0.04 | 55.63 ± 0.08 | 85.02 ± 0.27 | 84.21 ± 0.39 | 69.90 ± 0.13 |
| | 0-shot + FGC | 17.14 ± 0.16 | 90.05 ± 0.02 | 55.21 ± 0.05 | 85.64 ± 0.20 | 91.18 ± 0.05 | 71.27 ± 0.04 |
| | 5-shot | 17.51 ± 0.37 | 90.90 ± 0.03 | 56.65 ± 0.02 | 79.53 ± 1.36 | 82.50 ± 0.31 | 69.30 ± 0.29 |
| | 5-shot + FGC | 18.47 ± 0.46 | 90.73 ± 0.05 | 56.48 ± 0.03 | 81.50 ± 1.21 | 88.73 ± 0.77 | 70.89 ± 0.34 |
| CoT | 0-shot | 14.06 ± 0.26 | 89.94 ± 0.08 | 54.66 ± 0.13 | 88.39 ± 0.43 | 84.29 ± 0.47 | 69.84 ± 0.11 |
| | 0-shot + FGC | 15.50 ± 0.26 | 89.49 ± 0.08 | 53.98 ± 0.08 | 89.07 ± 0.73 | 92.75 ± 0.29 | **71.39** ± 0.17 |
| | 5-shot | 18.03 ± 0.61 | 90.76 ± 0.08 | 56.38 ± 0.09 | 81.54 ± 1.47 | 81.96 ± 0.37 | 69.18 ± 0.01 |
| | 5-shot + FGC | 13.05 ± 1.22 | 90.00 ± 0.14 | 55.34 ± 0.07 | 80.88 ± 1.28 | 89.58 ± 0.14 | 69.95 ± 0.11 |
| QLoRA | default | 28.55 ± 0.74 | 91.72 ± 0.15 | 57.16 ± 0.31 | 28.50 ± 0.61 | 77.75 ± 0.11 | 60.85 ± 0.33 |
| | default + FGC | **29.07** ± 0.48 | **91.78** ± 0.08 | **57.51** ± 0.47 | 28.17 ± 2.29 | 86.48 ± 0.13 | 62.59 ± 0.55 |
| | | | | **WikiDocKG** | | | |
| GAP | BART-base | 15.51 ± 0.52 | 87.84 ± 1.35 | 45.31 ± 3.27 | 25.82 ± 7.89 | 32.45 ± 9.08 | 45.95 ± 4.97 |
| FBF | 1st stage | 18.34 ± 0.02 | 86.66 ± 0.00 | 44.41 ± 0.06 | **59.19** ± 0.07 | 78.62 ± 0.59 | 61.37 ± 0.12 |
| | 2nd stage | 18.42 ± 0.10 | 86.94 ± 0.04 | 45.05 ± 0.18 | 55.20 ± 0.22 | 79.02 ± 0.65 | 60.89 ± 0.12 |
| Direct | 0-shot | 16.96 ± 0.62 | 90.18 ± 0.08 | 51.28 ± 0.25 | 33.37 ± 0.43 | 78.34 ± 0.16 | 59.44 ± 0.13 |
| | 0-shot + FGC | 19.40 ± 0.06 | 90.39 ± 0.02 | 51.80 ± 0.02 | 33.95 ± 0.26 | **85.12** ± 0.08 | 61.45 ± 0.07 |
| | 5-shot | 19.08 ± 0.40 | 90.59 ± 0.05 | 52.29 ± 0.11 | 34.37 ± 0.64 | 77.43 ± 0.33 | 60.20 ± 0.28 |
| | 5-shot + FGC | 20.84 ± 0.08 | 90.72 ± 0.01 | 52.72 ± 0.00 | 34.96 ± 0.64 | 84.42 ± 0.04 | 62.06 ± 0.12 |
| CoT | 0-shot | 6.96 ± 0.46 | 88.78 ± 0.01 | 48.27 ± 0.23 | 30.50 ± 0.33 | 60.11 ± 0.54 | 52.23 ± 0.22 |
| | 0-shot + FGC | 9.67 ± 2.99 | 89.20 ± 0.15 | 49.35 ± 0.47 | 29.81 ± 3.60 | 70.43 ± 3.17 | 54.94 ± 1.88 |
| | 5-shot | 10.66 ± 0.71 | 89.58 ± 0.12 | 50.28 ± 0.30 | 32.47 ± 0.62 | 67.02 ± 0.74 | 55.50 ± 0.42 |
| | 5-shot + FGC | 16.50 ± 1.53 | 90.18 ± 0.07 | 51.56 ± 0.16 | 33.78 ± 0.61 | 79.19 ± 1.15 | 59.52 ± 0.62 |
| QLoRA | default | **23.97** ± 3.22 | 91.06 ± 0.60 | 53.94 ± 1.15 | 35.28 ± 1.78 | 71.38 ± 4.55 | 60.51 ± 2.18 |
| | default + FGC | **26.29** ± 1.44 | **91.39** ± 0.28 | **54.70** ± 0.52 | 36.27 ± 1.01 | 81.35 ± 2.43 | **63.29** ± 1.13 |

Table 2: Detailed results for the `Qwen-3.5-9B` on document-level datasets; mean ± std over 3 runs.

| Dataset | Method | Qwen3.5-9B | Qwen3.5-27B | Llama-3.3-70B | GPT-5.1 |
|---|---|---|---|---|---|
| WebNLG | FBF | 78.32 ± 0.09 | 78.98 ± 0.07 | 77.25 ± 0.03 | 77.04 |
| | Direct | 79.63 ± 0.09 | 79.87 ± 0.14 | 79.99 ± 0.10 | 78.22 |
| | CoT | 79.83 ± 0.18 | 80.02 ± 0.13 | 79.09 ± 0.21 | 79.46 |
| | QLoRA | 79.87 ± 0.06 | 80.27 ± 0.19 | 79.97 ± 0.16 | – |
| LAGRANGE | FBF | 72.23 ± 0.03 | 71.82 ± 0.03 | 71.93 ± 0.10 | 71.30 |
| | Direct | 74.21 ± 0.43 | 74.70 ± 0.07 | 74.55 ± 0.65 | 73.23 |
| | CoT | 73.57 ± 0.23 | 73.57 ± 0.12 | 72.32 ± 0.11 | 74.77 |
| | QLoRA | 66.65 ± 1.00 | 67.48 ± 0.17 | 66.24 ± 0.08 | – |
| LAGRANGE$_{DOC}$ | FBF | 70.00 ± 0.11 | 69.14 ± 0.14 | 69.57 ± 0.09 | 68.89 |
| | Direct | 71.27 ± 0.04 | 71.67 ± 0.10 | 72.31 ± 0.12 | 71.48 |
| | CoT | 71.39 ± 0.17 | 71.58 ± 0.09 | 71.38 ± 0.14 | 71.23 |
| | QLoRA | 62.59 ± 0.55 | 66.18 ± 3.73 | – | – |
| WikiDocKG | FBF | 61.73 ± 0.12 | 60.36 ± 0.20 | 61.73 ± 0.02 | 58.04 |
| | Direct | 62.06 ± 0.12 | 62.14 ± 0.10 | 58.46 ± 0.58 | 63.09 |
| | CoT | 59.52 ± 0.62 | 56.51 ± 0.14 | 52.74 ± 0.32 | 61.17 |
| | QLoRA | 63.29 ± 1.13 | 62.35 ± 1.20 | – | – |

Table 3: Performance across model scales. Scores are normalized averages across all evaluation metrics.

der AlignScore. Comparing the two FBF variants[11] suggests that current factual consistency metrics are sensitive to the elaborate discourse and coreference introduced by fluent rewriting.

**Inference-time grounding improves factual consistency.** Adding FGC generally increases both AlignScore and FactSpotter, wchich indicates that

[11] Discussed in Appendix F.

inference-time evidence grounding may improve alignment without requiring manual textual annotations. At the same time, FGC often reduces BLEU scores, suggesting that lexical overlap metrics alone do not reliably capture factual grounding.

**Fine-tuning improves reference similarity but exposes graph–reference mismatch.** QLoRA achieves the strongest BLEU, BERTScore, and BLEURT scores across datasets, showing that supervised adaptation improves fluency and similarity to reference texts. However, these gains do not necessarily translate into stronger factual consistency. On LAGRANGE and $\text{LAGRANGE}_{\text{DOC}}$, QLoRA substantially reduces AlignScore; for instance, on $\text{LAGRANGE}_{\text{DOC}}$, AlignScore drops from 85.02 with direct 0-shot prompting to 28.50 with QLoRA (Table 2), despite improvements in lexical and semantic similarity. This discrepancy likely stems from a mismatch between the references and the input graphs. In LAGRANGE and $\text{LAGRANGE}_{\text{DOC}}$, reference annotations often contain information beyond the automatically constructed graphs, which encode only a subset of automatically resolved WikiData relations. As a result, models trained or prompted with these examples may learn to generate richer descriptions that include details unsupported by the input evidence. Reference-based metrics therefore improve, while factual consistency metrics degrade.

**Overall trade-off.** No method dominates across all metrics. QLoRA performs best in terms of fluency and reference similarity, whereas FGC and FBF are more effective at preserving factual grounding. This trade-off is most pronounced on complex document-level datasets, where reference texts may contain information that is not fully represented in the input graph.

**Scaling trends across larger models.** Across model scales, the relative behavior of method families remains largely unchanged (Table 3). Larger models yield only moderate gains, suggesting that the task is not strongly scaling-limited: KG-to-text generation mainly tests graph verbalization into coherent text, with limited reliance on external knowledge or complex reasoning. This supports the use of smaller open-weight models in practical applications. At the same time, our analysis shows that current metrics remain sensitive to reference discourse flow and coreference effects, warranting caution when comparing scores on document-level datasets.

### 6.1 Error Analysis

We analyze errors made by the two strongest FGC-guided configurations (`Qwen3.5-9B` and `Qwen3.5-27B` with QLoRA fine-tuning) to identify remaining challenges in document-level KG-to-text generation and evaluation. We sample 50 examples from the WIKIDOCKG test split and manually compare the outputs of both models against the input graphs. Overall, 90% of input triples are grounded in the generated texts, while 3% are omitted and 7% of generated facts are inferred or misinterpreted from the provided graphs. Automatic metrics vary substantially across both factually grounded and graph-deviating examples, further highlighting the need for more stable document-level generation metrics.

## 7 Conclusion

This work advances KG-to-text generation from sentence-level verbalization toward document-level generation from large knowledge graphs. We introduced $\text{LAGRANGE}_{\text{DOC}}$ and WIKIDOCKG, two benchmarks designed to evaluate long-form generation under larger graph inputs, broader relation vocabularies, and stronger grounding requirements. We evaluated prompting, fact-by-fact generation, factuality-guided candidate selection, and QLoRA fine-tuning across several LLMs.

The results demonstrate that reference-based and graph-based evaluation criteria can lead to different conclusions. QLoRA improves lexical and semantic similarity, but its gains may come at the cost of factual consistency when references are only partially supported by the input graph. Conversely, FBF and FGC more effectively preserve graph faithfulness, although they do not always improve reference similarity. Scaling to larger models provides limited additional benefit, indicating that reliable document-level KG-to-text generation remains constrained by grounding and evaluation challenges.

These results motivate a shift toward multi-perspective evaluation that treats graph-grounded factuality as a first-class criterion alongside fluency and reference similarity. that balance semantic fluency, reference adequacy, and factual consistency. We expect $\text{LAGRANGE}_{\text{DOC}}$ and WIKIDOCKG to support further research on faithful long-form generation from structured knowledge.

## Limitations

This study is restricted to general-domain KG-to-text generation with Wikipedia-derived or Wikidata-aligned resources. While this setting is broad and widely used, it may not reflect the constraints of domain-specific applications such as biomedical, legal, or enterprise KG-to-text generation, where graph schemas, terminology, relation semantics, and acceptable discourse structures can differ substantially. The experiments are also limited to English generation and do not address multilingual, cross-lingual, or culturally localized generation.

The proposed FactSpotter-Guided Consistency (FGC) method is based on FactSpotter, and FactSpotter is also used as one of the reported factual consistency metrics. Consequently, FactSpotter gains for FGC should not be interpreted as independent evidence of improved factuality. We mitigate this by reporting multiple metrics, but automatic metrics remain imperfect proxies for human judgments, especially for long-form outputs where errors may involve discourse coherence, omission, aggregation, or unsupported claims spanning multiple sentences. Our manual analysis of 50 WIKIDOCKG examples is intended as diagnostic error analysis rather than a comprehensive human evaluation, and it does not cover all datasets, generation methods, or model scales.

## References


Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Sebastian Borgeaud and Guy Emerson. 2020. Leveraging sentence similarity in natural language generation: Improving beam search using range voting. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 97–109, Online. Association for Computational Linguistics.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020. KGPT: Knowledge-grounded pre-training for data-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8635–8648, Online. Association for Computational Linguistics.

Alla Chepurova, Aydar Bulatov, Mikhail Burtsev, and Yuri Kuratov. 2026. Wikontic: Constructing Wikidata-aligned, ontology-aware knowledge graphs with large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8304–8319, Rabat, Morocco. Association for Computational Linguistics.

Anthony Colas, Mehrdad Alvandipour, and Daisy Zhe Wang. 2022. GAP: A graph-aware language model framework for knowledge graph-to-text generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5755–5769, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Anthony Colas, Ali Sadeghian, Yue Wang, and Daisy Zhe Wang. 2021. EventNarrative: A large-scale event-centric dataset for knowledge graph-to-text generation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Oriol Domingo, David Bergés, Roser Cantenys, Roger Creus, and José A. R. Fonollosa. 2020. Enhancing sequence-to-sequence modelling for RDF triples to natural text. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 40–47, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Ondřej Dušek and Zdeněk Kasner. 2020. Evaluating semantic accuracy of data-to-text generation with natural language inference. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 131–137, Dublin, Ireland. Association for Computational Linguistics.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017a. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017b. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Jie He, Yijun Yang, Wanqiu Long, Deyi Xiong, Victor Gutierrez Basulto, and Jeff Z. Pan. 2025. Evaluating and improving graph to text generation with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10219–10244, Albuquerque, New Mexico. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Dmitrii Iarosh, Alexander Panchenko, and Mikhail Salnikov. 2025. On reducing factual hallucinations in graph-to-text generation using large language models. In *Proceedings of the Workshop on Generative AI and Knowledge Graphs (GenAIK)*, pages 43–53, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Zdeněk Kasner and Ondrej Dusek. 2022. Neural pipeline for zero-shot data-to-text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3914–3932, Dublin, Ireland. Association for Computational Linguistics.

Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. JointGT: Graph-text joint representation learning for text generation from knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2526–2538, Online. Association for Computational Linguistics.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293, Minneapolis, Minnesota. Association for Computational Linguistics.

Guohong Li and Deyi Xiong. 2025. Towards optimal evaluation efficiency for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14176–14183, Suzhou, China. Association for Computational Linguistics.

Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 1–9, Tilburg University, The Netherlands. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2267–2277, Minneapolis, Minnesota. Association for Computational Linguistics.

Ali Mousavi, Xin Zhan, He Bai, Peng Shi, Theodoros Rekatsinas, Benjamin Han, Yunyao Li, Jeffrey Pound, Joshua M. Susskind, Natalie Schluter, Ihab F. Ilyas, and Navdeep Jaitly. 2024. Construction of paired knowledge graph - text datasets informed by cyclic evaluation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3782–3803, Torino, Italia. ELRA and ICCL.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, and 5 others. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of*

the Association for Computational Linguistics, pages 7881–7892, Online. Association for Computational Linguistics.

Yifei Song and Claire Gardent. 2025. MuCAL: Contrastive alignment for preference-driven KG-to-text generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14227–14270, Suzhou, China. Association for Computational Linguistics.

Yifei Song, William Soto Martinez, Anna Nikiforovskaya, Evan Parker Kelly Chapple, and Claire Gardent. 2025. Multilingual verbalisation of knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1111–1162, Suzhou, China. Association for Computational Linguistics.

Craig Thomson and Ehud Reiter. 2020. A gold standard methodology for evaluating accuracy in data-to-text systems. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 158–168, Dublin, Ireland. Association for Computational Linguistics.

Luyu Wang, Yujia Li, Ozlem Aslan, and Oriol Vinyals. 2021. WikiGraphs: A Wikipedia text - knowledge graph paired dataset. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 67–82, Mexico City, Mexico. Association for Computational Linguistics.

Shaobo Wang, Cong Wang, Wenjie Fu, Yue Min, Mingquan Feng, Isabel Guan, Xuming Hu, Conghui He, Cunxiang Wang, Kexin Yang, Xingzhang Ren, Fei Huang, Dayiheng Liu, and Linfeng Zhang. 2026. Rethinking LLM evaluation: Can we evaluate LLMs with 200× less data? In *International Conference on Learning Representations (ICLR)*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Shuzhou Yuan and Michael Färber. 2024. GraSAME: Injecting token-level structural information to pretrained language models via graph-guided self-attention mechanism. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 920–933, Mexico City, Mexico. Association for Computational Linguistics.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Kun Zhang, Oana Balalau, and Ioana Manolescu. 2023. FactSpotter: Evaluating the factual faithfulness of graph-to-text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10025–10042, Singapore. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Feng Zhao, Hongzhi Zou, and Cheng Yan. 2023. Structure-aware knowledge graph-to-text generation with planning selection and similarity distinction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8693–8703, Singapore. Association for Computational Linguistics.

Tianxin Zhao, Yingxin Liu, Xiangdong Su, Jiang Li, and Guanglai Gao. 2024. Exploring the synergy of dual-path encoder and alignment module for better graph-to-text generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6980–6991, Torino, Italia. ELRA and ICCL.

# A WikiDocKG Construction Details

## A.1 Knowledge Graph Construction

We employ Wikontic (Chepurova et al., 2026) to construct a knowledge graph from a Wikipedia article and automatically align it against WikiData, using `GPT-5.1` as an LLM backbone.

To perform document-level KG extraction, we adapted the original system pipeline. First, instead of splitting a long input document into chunks of 3500 characters, we identify sentence boundaries. The document is then segmented according to these boundaries, such that no chunk exceeds 3500 characters.

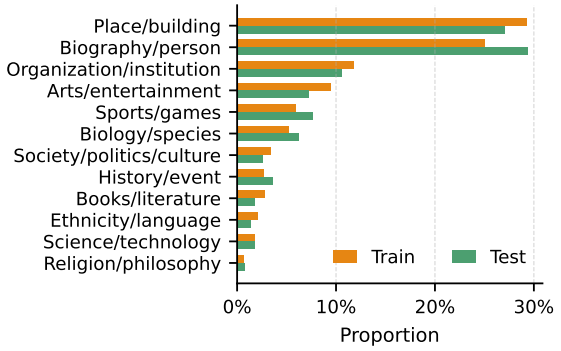
## A.2 Test Split Diagnostics

To verify that the dataset partitions are comparable, we analyze topic, document-length, and graph-size distributions across splits. Document length is measured in number of sentences, and graph size as the number of extracted triples per document.

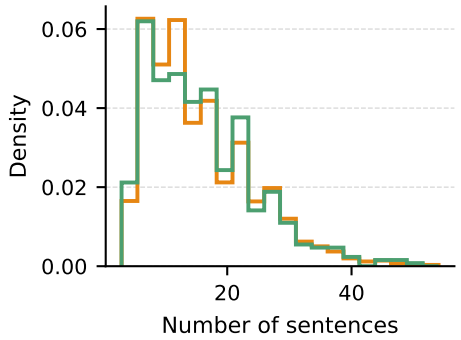
For topical analysis, we assign each document a coarse-grained topic label using a pretrained zero-

shot NLI classifier.[12] We evaluate a fixed set of candidate topic labels using a natural language hypothesis template ("This Wikipedia article is about ...") and assign each document the top-1 label based on the highest entailment score. The classifier is applied to a truncated prefix of each document for efficiency. These labels are used solely to compare topic distributions across splits.

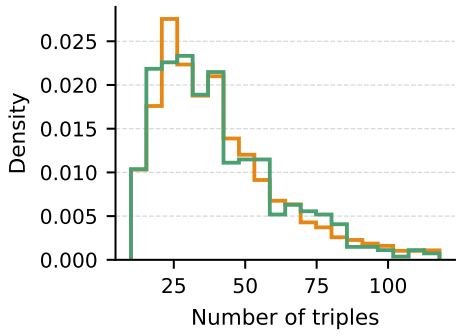
Figure 3 shows the resulting distributions. We observe no substantial differences across splits for any of the considered properties.




(a) Topic distribution




(b) Document length




(c) Graph size

Figure 3: Train/test distributions of WikiDocKG topics, document lengths, and graph sizes.

[12] `MoritzLaurer/mDeBERTa-v3-base-xnli-multilingual-nli-2mil7`

### A.3 Data privacy and offensive content

All datasets used in this work are derived from public Wikipedia text and/or Wikidata triples. These sources may contain names of public figures and other real-world entities, but the datasets are not constructed from private user data and do not intentionally collect personal, sensitive, or confidential information. During dataset construction, we retain only information available in the source resources and do not add any additional personal identifiers. We also do not attempt to infer private attributes about individuals. Because Wikipedia and Wikidata may still biased, or otherwise harmful content, generated outputs may reproduce such content or reflect biases present in the underlying sources. We therefore release the data with provenance information and recommend that downstream users apply task-specific content filtering and privacy review before deployment in user-facing systems.

## B Subset Construction Details

This section provides implementation details for the clustering-based subset construction described in Section 5.2.

**Sample Representation.** Each graph is paired with a natural language verbalization, which we encode using the pretrained sentence embedding model `nomic-ai/nomic-embed-text-v1.5`. These embeddings define the semantic space used for clustering.

**Stratified Clustering.** To preserve the complexity profile of the original split, we first partition examples into buckets defined by the number of graph nodes. We allocate the target subset budget across buckets proportionally to their frequency in the full split. Within each bucket, we apply $k$-means clustering to the corresponding text embeddings, setting the number of clusters equal to the bucket-specific quota. We use $k$-means++ initialization and a fixed maximum number of iterations.

**Subset Sampling.** Within each cluster, representative instances are selected uniformly at random. This preserves the original distribution over graph complexity while reducing redundancy through semantic clustering. We fix all random seeds and release the selected split-relative subset indices for reproducibility.

## C Statistics Used for Metric Averaging

To compute the normalized average score, each metric is first min–max normalized across all experiments:

$$\tilde{x}_{i,m} = \frac{x_{i,m} - \min_j x_{j,m}}{\max_j x_{j,m} - \min_j x_{j,m}}. \quad (1)$$

The aggregate score is then obtained by averaging the normalized values across metrics:

$$\mathrm{Avg}_i = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \tilde{x}_{i,m}. \quad (2)$$

Table 4 reports the minimum and maximum values used for each metric.

Table 4: Values used for normalization of each metric.

| Metric | Min | Max |
|---|---|---|
| BLEU | 0.00 | 1.00 |
| BERTScore | 0.75 | 1.00 |
| BLEURT | 0.15 | 1.10 |
| AlignScore | 0.00 | 1.00 |
| FactSpotter | 0.00 | 1.00 |

## D Prompt Templates

This section lists the prompt templates used in the tested prompt-based approaches. Placeholders such as `{triples}` denote values instantiated at inference time.

### D.1 Direct

The Direct baseline generates the complete output paragraph from the full set of input triples in a single model call.

```
You are a text generator that produces exactly
one concise, fluent, Wikipedia-like paragraph
covering all facts present in the input triples.
Do not output any reasoning, analysis, planning,
or internal thoughts.
Do not describe your process. Output only the
paragraph.
Refer only to items explicitly present in the
graph, using no external information.

Triples:
{triples}

Final:
```

### D.2 FBF

The Fact-by-Fact (FBF) baseline first realizes each triple as an independent factual sentence. When naturalization is enabled, the generated sentences are rewritten into a single fluent paragraph.

#### D.2.1 First Stage: Triple Realization

```
You convert knowledge graph triples into simple
English factual sentences.
Rules:
- Use exactly ONE sentence per triple.
- Do NOT add facts.
- Do NOT remove facts.
- Do NOT infer anything.
- Preserve all names, numbers, and dates
exactly.
- Output only the sentence.

Triple:
Subject: {subject}
Predicate: {predicate}
Object: {object}

Sentence:
```

#### D.2.2 Second Stage: Naturalization

```
You are a careful editor. Rewrite the input
factual sentences into a
single fluent Wikipedia-like paragraph.
Rules:
- Do NOT add facts.
- Do NOT remove facts.
- Do NOT infer anything.
- Preserve all names, numbers, and dates
exactly.
- Avoid repetition by using pronouns and
conjunctions when safe.
- Output only the final paragraph.

Input:
{facts}

Final:
```

### D.3 CoT

The Chain-of-Thought (CoT) baseline asks the model to reason over the graph before producing the final paragraph. During evaluation, only the text after the `Final:` marker is retained.

```
Act as a system that describes all nodes of the
graph connected by edges
as a Wikipedia-like text. Refer only to items
explicitly present in the
graph, using no external information.

Triples:
{triples}

First, reason step by step about what these
triples mean.
Then, on a new line, write 'Final:' followed by
text concisely and fluently summarizing all the
facts.

Answer: Let's think step-by-step:
```

## E Hyperparameters

Table 5 reports the decoding hyperparameters used for prompting-based methods. Values reported in

the form $a/b/c/d$ correspond to WebNLG, LAGRANGE, LAGRANGE$_{\text{DOC}}$, and WikiDocKG, respectively. Generation limits are adjusted per dataset to reflect variation in target output length.

Table 5: Decoding hyperparameters.

| Hyperparameter | Value |
|---|---|
| Maximum new tokens | 128 / 128 / 256 / 512 |
| Temperature | 0.7 |
| Top-$p$ | 0.9 |
| Repetition penalty | 1.4 |

Table 6 reports the hyperparameters used for QLoRA fine-tuning.

Table 6: QLoRA fine-tuning hyperparameters.

| Hyperparameter | Value |
|---|---|
| Epochs | 1 |
| Learning rate | $2 \times 10^{-4}$ |
| Learning-rate scheduler | Constant with warmup |
| Warmup ratio | 0.03 |
| Batch size | 1 |
| Gradient accumulation steps | 4 / 4 / 2 / 1 |
| Maximum sequence length | 256 / 256 / 512 / 1024 |
| Maximum gradient norm | 0.3 |
| LoRA rank ($r$) | 16 |
| LoRA scaling factor ($\alpha$) | 32 |
| LoRA dropout | 0.05 |
| Quantization | 4-bit |
| Generation temperature | 0.7 |
| Generation top-$p$ | 0.9 |

Table 7 reports the hyperparameters used for the GAP baseline.

Table 7: GAP fine-tuning hyperparameters.

| Hyperparameter | Value |
|---|---|
| Backbone model | BART-base |
| Epochs | 40 / 20 / 20 / 20 |
| Early stopping patience | 3 |
| Checkpoint selection | Best validation BLEU |
| Learning rate | $2\times10^{-5}$ |
| Learning-rate scheduler | Linear with warmup |
| Warmup ratio | 0.03 |
| Batch size | 16 / 8 / 8 / 2 |
| Gradient accumulation steps | 1 / 2 / 2 / 8 |
| Effective batch size | 16 |
| Maximum source length | 256 / 512 / 512 / 512 |
| Maximum new tokens | 128 / 128 / 256 / 512 |
| Maximum graph nodes | 50 / 64 / 64 / 512 |
| Relation nodes | Enabled |
| Type encoding | Enabled |
| Graph dropout | 0.1 |
| Beam size | 5 / 5 / 5 / 3 |
| Length penalty | 1.0 / 1.0 / 2.0 / 2.0 |

# F Fact-by-Fact Generation Examples

This section presents an analysis of FBF generations produced by `Qwen-3.5-9B` at both stages, using representative examples from each dataset.

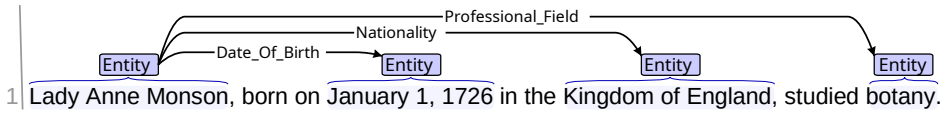
Figure 4 presents a WebNLG instance, including the reference annotation, the first- and second-stage FBF generations, and the corresponding evaluation scores. The source input consists of three triples: `(Lady Anne Monson, dateOfBirth, 1726-01-01)`, `(Lady Anne Monson, nationality, Kingdom of England)`, and `(Lady Anne Monson, professionalField, Botany)`. The first-stage output, shown in Figure 4b, is transparent and semantically well aligned with the source, but it is repetitive and less natural than the reference. The second-stage output, shown in Figure 4c, is more fluent and more elaborated, but is strongly penalized by both AlignScore and FactSpotter. Although it preserves the input facts, it introduces an unsupported characterization by describing `Lady Anne Monson` as a "prominent figure," a claim not entailed by the input triples.

Figure 5 shows an example from the LAGRANGE dataset. The source input consists of three triples: `(Murat Yakin, date of birth, 1974)`, `(Murat, language of work or name, Turkish)`, and `(Murat Yakin, date of birth, 15 September 1974)`.

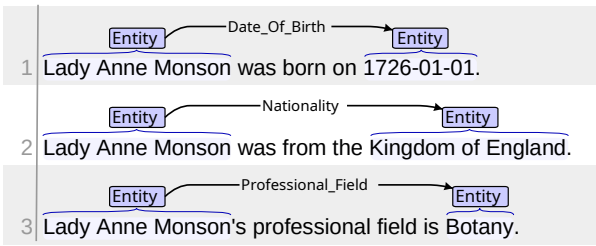
The first-stage output is highly faithful to the input triples, as reflected by its near-perfect AlignScore and FactSpotter values. However, it is fragmented and repetitive, and it renders the predicate `language of work or name` almost literally, producing an unnatural sentence.

The second-stage output is more fluent and compact. Importantly, the textual reference in Figure 5a, which is used for BLEU, BERTScore, BLEURT, and AlignScore evaluation, contains information that is absent from the accompanying graph: it describes `Murat Yakin` as a Swiss football coach and former player. Such graph–reference discrepancies are common in LAGRANGE annotations and can degrade performance when models are given few-shot examples from, or are fine-tuned on, this dataset.

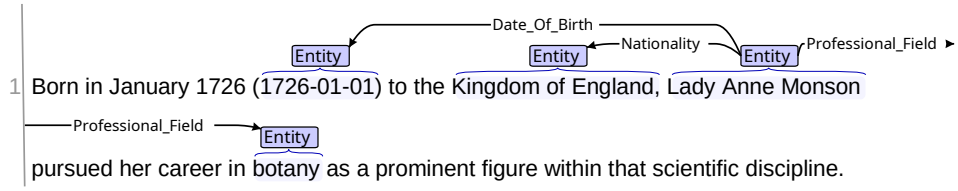
During the second FBF stage, the language model elaborates the graph-based description by introducing additional facts. Some of this elaboration improves lexical overlap with the reference. In particular, describing `Murat Yakin` as a footballer, despite this information not being present




(a) Reference.




(b) First FBF stage. BLEU=13.2, BERTScore=92.3, BLEURT=69.0, AlignScore=99.6, FactSpotter=100, Avg=67.8.




(c) Second FBF stage. BLEU=15.4, BERTScore=90.9, BLEURT=67.8, AlignScore=5.7, FactSpotter=66.7, Avg=41.4.

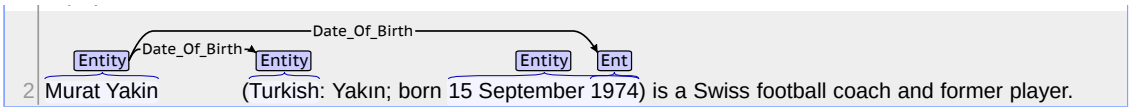
Figure 4: Reference annotation and first- and second-stage generations for a WebNLG example.

in the input graph. At the same time, the model over-interprets the predicate `language of work or name`. These unsupported inferences likely contribute to the sharp drop in AlignScore. Overall, this example shows that second-stage rewriting can improve fluency and reference similarity by adding plausible background information, but that such additions may semantically deviate from the input graph.

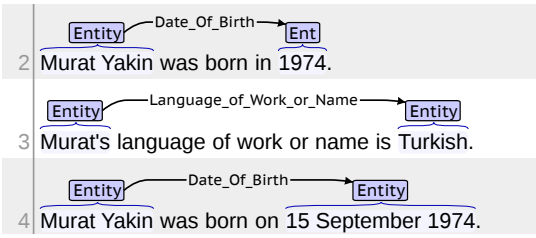
Figure 6 shows a LAGRANGE$_{\text{DOC}}$ example with the following input triples:

1. (`Ernest Bevin`, `member of political party`, `Labour Party`)
2. (`Ernest Bevin`, `member of political party`, `Transport and General Workers' Union`)
3. (`Ernest Bevin`, `date of death`, `1951`)
4. (`Ernest Bevin`, `position held`, `Secretary of State for Foreign and Commonwealth Affairs`)
5. (`Ernest Bevin`, `position held`, `Minister of Labour`)
6. (`Ernest Bevin`, `given name`, `Ernest`)
7. (`Transport and General Workers' Union`, `inception`, `1922`)
8. (`Ernest Bevin`, `date of birth`, `9 March 1881`)
9. (`Ernest Bevin`, `family name`, `Bevin`)
10. (`Ernest Bevin`, `occupation`, `politician`)
11. (`Ernest Bevin`, `date of death`, `14 April 1951`)
12. (`Ernest Bevin`, `name in native language`, `Ernest Bevin`)
13. (`Ernest Bevin`, `date of birth`, `1881`)
14. (`Ernest Bevin`, `position held`, `Transport and General Workers' Union`)

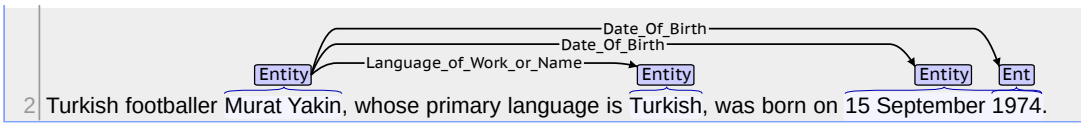
This example illustrates a common pattern: second-stage rewriting improves fluency and aggregation, as reflected by higher BLEU and BERTScore, but can weaken graph faithfulness when dense, partially redundant triples are compressed into a document-level summary. In particular, the first-stage output already introduces relation-level drift by rendering the `position held` triple involving `Transport and General Workers' Union` as Bevin holding the position of




(a) Reference.




(b) First FBF stage. BLEU=8.2, BERTScore=86.1, BLEURT=50.6, AlignScore=99.7, FactSpotter=100, Avg=57.9.




(c) Second FBF stage. BLEU=9.9, BERTScore=88.5, AlignScore=55.3, FactSpotter=11.2, Avg=66.7.

Figure 5: Reference annotation and first- and second-stage generations for a LAGRANGE example.

leader. The second-stage rewrite preserves this weakly supported interpretation while further inferring information such as his age at death. Together with the use of anaphora in long-discourse generation, these deviations explain the sharp decline in factuality-oriented metrics, particularly the low AlignScore of 9.2.

The WikiDocKG triples for the example in Figures 7, 8, and 9 are:

1. (Elliot Dason-Barber, date of birth, 7 December 1969)

2. (Elliot Dason-Barber, country of citizenship, United Kingdom)

3. (Elliot Dason-Barber, occupation, Formula One engineer)

4. (Elliot Dason-Barber, occupation, motorsport engineer)

5. (Elliot Dason-Barber, educated at, University of Huddersfield) ⟨academic degree: mechanical engineering; end time: 1993 ⟩

6. (Elliot Dason-Barber, employer, Jaguar Land Rover) ⟨start time: 1991; position held: sponsored undergraduate ⟩

7. (Elliot Dason-Barber, employer, Jaguar Land Rover) ⟨position held: vehicle dynamics engineer ⟩

8. (Elliot Dason-Barber, employer, Ricardo) ⟨position held: Head of Vehicle Dynamics ⟩

9. (Elliot Dason-Barber, field of work, motorsport industry) ⟨applies to jurisdiction: North America ⟩

10. (Elliot Dason-Barber, employer, Red Bull Racing) ⟨start time: late 2004; position held: Vehicle Dynamics Group Leader ⟩

11. (Elliot Dason-Barber, participant in, design of Red Bull RB1)

12. (Elliot Dason-Barber, participant in, design of Red Bull RB2)

13. (Elliot Dason-Barber, participant in, design of Red Bull RB3)

14. (Elliot Dason-Barber, participant in, design of Red Bull RB4)

15. (Elliot Dason-Barber, participant in, design of Red Bull RB5)

16. (Elliot Dason-Barber, field of work, vehicle dynamics)

17. (Elliot Dason-Barber, field of work, driver-in-the-loop simulation)

18. (Elliot Dason-Barber, employer, Lotus Racing) ⟨start time: 2010; position held: Head of Vehicle Dynamics ⟩

19. (Elliot Dason-Barber, employer, Lotus Racing) ⟨position held: Head of Research and Development ⟩

20. (Lotus Racing, followed by, Caterham F1)

21. (Elliot Dason-Barber, employer, Caterham F1) ⟨position held: Head of Vehicle Dynamics; position held: Head of Research and Development ⟩

22. (Elliot Dason-Barber, employer, Sauber Motorsport) ⟨start time: 2014; position held: Head of Vehicle Performance ⟩

23. (Elliot Dason-Barber, employer, Sauber Motorsport) ⟨position held: Head of Performance Engineering ⟩

24. (Sauber C34, designer, Elliot Dason-Barber) ⟨co-designer: Eric Gandelin; co-designer: Willem Toet ⟩

25. (Sauber C35, designer, Elliot Dason-Barber) ⟨series ordinal: successor of Sauber C34 ⟩

26. (Elliot Dason-Barber, employer, JRM Advanced Engineering) ⟨start time: 2016; position held: Chief Engineer ⟩

27. (Elliot Dason-Barber, employer, JRM Advanced Engineering) ⟨position held: Technical Director ⟩

28. (JRM Advanced Engineering, client, Jaguar Land Rover)

29. (JRM Advanced Engineering, client, Aston Martin)

30. (JRM Advanced Engineering, client, Arctic Trucks)

31. (JRM Advanced Engineering, client, Subaru)

32. (JRM Advanced Engineering, client, Nismo)

33. (Elliot Dason-Barber, employer, Ansible Motion) ⟨position held: Technical Director; start time: 2025 ⟩

34. (Ansible Motion, industry, simulation)

35. (Ansible Motion, product or service, driver-in-the-loop simulation systems)

36. (driver-in-the-loop simulation systems, used by, motorsport teams)

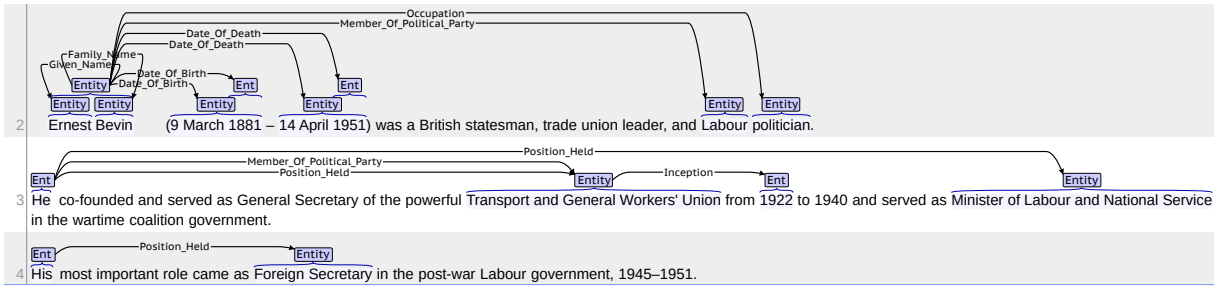
37. (driver-in-the-loop simulation systems, used by, automotive manufacturers)

The WikiDocKG graph annotations include qualifiers more frequently and therefore appear to align more closely with the reference than those in $\text{LAGRANGE}_{\text{DOC}}$.

The second-stage rewrite compresses these triples into a coherent narrative, but in doing so it introduces several temporal and relational inferences. For example, it states that Dason-Barber worked at Sauber "from 2014 until 2016," although the graph only specifies a 2014 start time; the end date is supported by the reference, but not by the input graph itself. The rewrite also reverses the ordering of his Sauber roles by presenting Head of Performance Engineering as prior to Head of Vehicle Performance, even though both roles appear only as qualifiers of separate `employer` predicates and no ordering between them is encoded. It also converts the `successor of Sauber C34` qualifier on `Sauber C35` into the unsupported claim that Dason-Barber "succeed[ed] it with the sole-designated Sauber C35." More broadly, the rewrite relies heavily on anaphora and temporal connective structure ("subsequently," "prior to," "following," "formerly," and "currently"), which improves readability but is not licensed by the input triples. Interestingly, however, the second FBF stage corrects the direction of the `client` relation, which was incorrectly rendered as "JRM Advanced Engineering is a client of [multiple clients]" during the context-loss first FBF stage (Figure 8).

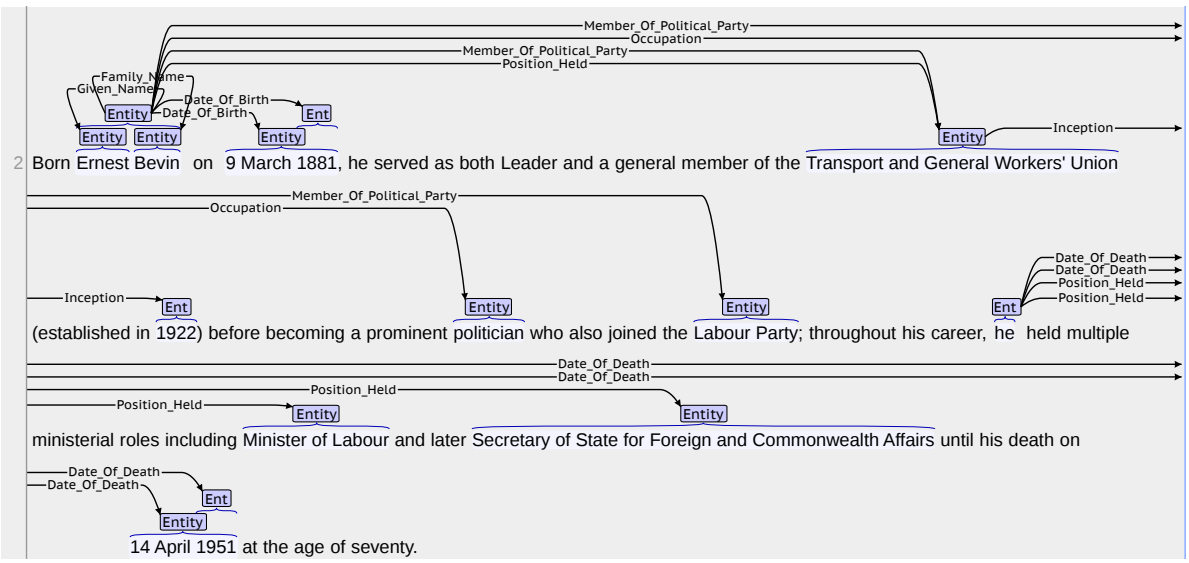
Overall, our analysis shows that discourse generation introduces a fundamental trade-off in document-level KG-to-text generation. Aggregating triple-level statements improves fluency, coherence, and readability by producing more natural

narratives. However, the same discourse operations can introduce temporal or relational implications that are not licensed by the input graph, and coreference can reduce the transparency of entity-level alignment for evaluation metrics. Document-level rewriting therefore improves textual quality while potentially weakening graph faithfulness, especially for inputs with dense qualifiers, partially redundant facts, or underspecified temporal relations.




(a) Reference.

(b) First FBF stage. BLEU=6.9, BERTScore=84.9, BLEURT=51.4, AlignScore=90.3, FactSpotter=92.9, Avg=53.6.




(c) Second FBF stage. BLEU=13.8, BERTScore=88.9, BLEURT=51.8, AlignScore=9.2, FactSpotter=71.4, Avg=37.7.

Figure 6: Reference annotation and first- and second-stage generations for a LAGRANGE$_{\text{DOC}}$ example.

Figure 7: Reference annotation for a WikiDocKG example.

Figure 8: First FBF stage; WikiDocKG example. BLEU=15.8, BERTScore=87.4, BLEURT=45.3, AlignScore=43.5, FactSpotter=100, Avg=48.2.

Figure 9: Second FBF stage; WikiDocKG. BLEU=18.4, BERTScore=89.6, BLEURT=40.7, AlignScore=23.6, FactSpotter=86.5, Avg=42.8.