SCENT: SELF-CONSISTENT EPISTEMIC NODE TEXTUALIZER

TECHNICAL DOCUMENTATION

ABSTRACT

Entity linking serves as a foundational capability in Natural Language Processing, enabling the precise alignment of unstructured textual mentions with unambiguous entities in a Knowledge Graph to ensure factual grounding. To facilitate robust and structure-aware generation, we propose **Scent**, a constrained generation framework that strictly grounds textual output in structured graph knowledge. Scent leverages an encoder backbone (e.g., RoBERTa) augmented with two specialized, simultaneously trained Low-Rank Adaptation (LoRA) modules: a **Graph-adapter (f_{graph})**, designed to synthesize semantic representations from structured data, and a **Text-adapter (f_{text})**, which manages the integration of linguistic context with these structure-grounded entity slots. By enforcing consistency between textual outputs and graph topology, this framework ensures that generated content remains strictly valid within the defined candidate set.

source code: https://github.com/ra0o0f/scent

1 Illustrative Framework

To demonstrate the Scent mechanism, consider a scenario where the model must generate a specific entity text. This generation is **constrained**: the output must be selected from a pre-defined candidate set of entities $\mathcal{C}$, ensuring the text is grounded in the Knowledge Graph (KG).

Example Scenario: We aim to predict the entity Kyoto within the following context:

> "The breathtaking gardens of **Kyoto** reflect the serene beauty of traditional Japanese culture."

Here, the target entity n_{target} is **Kyoto**.

1.1 Dual-View Representation

In our framework, **Kyoto** is processed through two distinct but aligned views.

The Textual View: The textual view presents the target entity within its linguistic context. We prepare the sequence for prediction by defining the entity slot with entity boundary markers and a fixed buffer of mask tokens (N). Crucially, to allow the model to dynamically learn entity boundaries within this fixed buffer, the entity label is explicitly terminated by a special `<end_of_title>` token.

$$X_{text} = [\dots \text{gardens}, \text{of}, \texttt{<node\_start>}, \underbrace{\texttt{<mask>}, \texttt{<mask>}, \texttt{<mask>}}_{N}, \texttt{<node\_end>}, \text{reflect}, \dots] \tag{1}$$

The Structural View: Simultaneously, Kyoto exists as a node in the KG. We follow the TokenGT [3] formatting to linearize its local neighborhood (k-hop traversal) into a sequence of node and edge semantic units. For example:

- (Kyoto) $\xrightarrow{\text{located in}}$ (Japan)
- (Kyoto) $\xrightarrow{\text{contains}}$ (Kinkaku-ji)

Crucially, to construct the initial feature representation for each entity, rather than employing a static embedding layer, we generate semantic representations directly from the text. We encode the textual labels of nodes and edges using the backbone model (e.g., `<s> Kyoto </s>`) and utilize the hidden state of the initial token (`<s>`) as the entity's embedding vector.

The sequence places the masked target node in its linearized context, as shown in:

$$X_{graph} = [\dots, \mathbf{n}_{Japan}, \mathbf{e}_{located\_in}, \mathbf{n}_{\texttt{<node\_mask>}}, \mathbf{e}_{contains}, \mathbf{n}_{Kinkaku-ji}, \dots] \tag{2}$$

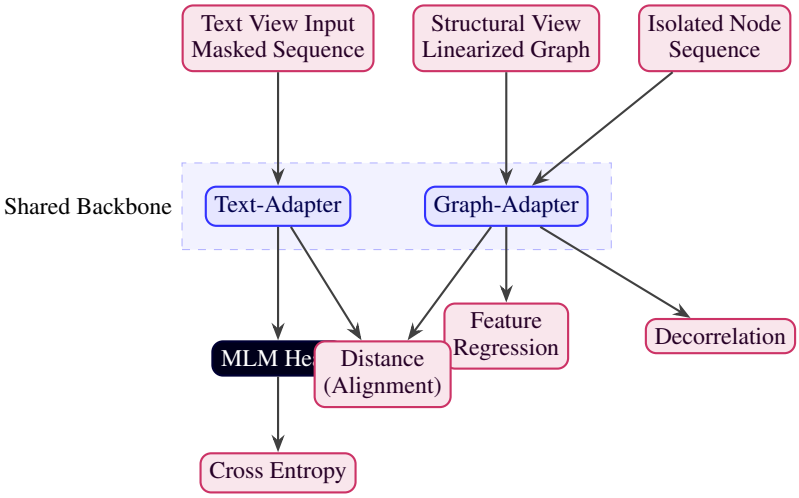

Figure 1: The Scent Architecture. Inputs are processed through shared backbones with specialized LoRA adapters, optimizing for multi-task objectives.

1.2 Training Objectives

Scent employs a multi-task strategy to enforce consistency between the textual output and the underlying graph topology.

1.2.1 Textual Generation

Using the **Text-Adapter**, we perform standard **Masked Language Modeling** on the tokens within the mask buffer. The model must predict the token IDs for "Kyoto" based on the surrounding sentence context.

We minimize the Cross-Entropy loss between the predicted logits and the actual entity label token IDs (Y_{target}) as defined in:

$$\mathcal{L}_{text} = \text{CrossEntropy}\Big(\text{Head}_{MLM}(f_{text}(X_{text})),\ Y_{target}\Big) \tag{3}$$

While the buffer size N is fixed (e.g., 20 masks) to accommodate the longest potential entities, the Cross-Entropy loss is calculated only on the specific subset of mask tokens corresponding to the actual entity length, meaning the remaining padding masks are excluded from the loss update even though they still participate in the self-attention mechanism.

1.2.2 Structure-Aware Feature Regression

Using the **Graph-Adapter**, we apply a **Masked Feature Regression** on the graph sequence. The model must reconstruct the semantic vector of the masked target node solely from its structural context (neighbors and edge relations).

We minimize the distance (e.g., Mean Squared Error or Negative Cosine Similarity) between the predicted vector and the ground truth as follow:

$$\mathcal{L}_{graph} = \text{Distance}\Big(f_{graph}(X_{graph})[\texttt{<node\_mask>}],\ \mathbf{n}_{ref}\Big) \tag{4}$$

1.2.3 Cross-Modal Alignment

To bridge the two views, we enforce a strict consistency constraint on the `<node_start>` token. The hidden state of `<node_start>` generated by the Text-Adapter must align with the graph representation of the target node generated by the Graph-Adapter.

To generate the reference representation for the target, we construct a new, isolated graph sequence containing only the target node:

$$X_{target\_node} = [\mathbf{n}_{Kyoto}] \tag{5}$$

We pass this sequence through the Graph-Adapter to obtain the target hidden state. We then minimize the distance between the text-derived hidden state of `<node_start>` and this graph-derived target representation:

$$\mathcal{L}_{align} = \text{Distance}\Big(f_{text}(X_{text})[\texttt{<node\_start>}],\ f_{graph}(X_{target\_node})\Big) \tag{6}$$

1.2.4 Representation Regularization

We observe that the Graph-Adapter is susceptible to representation collapse, where the model may converge toward trivial solutions despite the backbone being frozen. The precise root cause of this phenomenon, whether arising from the specific geometry of the pre-trained embedding space or the adapter's optimization dynamics, remains under active investigation.

As a provisional measure while we explore a definitive hard constraint, we apply a decorrelation objective(Barlow Twins [6]) to the batch of predicted graph-view representations to penalize redundancy between feature dimensions:

$$\mathcal{L}_{reg} = \text{Decorrelation}(Z_{batch}) \tag{7}$$

This objective encourages distributional diversity and helps destabilize trivial equilibria during the training process.

2 Inference and Constrained Generation

Scent introduces a high-efficiency inference mechanism designed to circumvent the latency bottlenecks typical of constrained generation. By leveraging the encoder-only architecture, we score all potential entities from the candidate set $\mathcal{C}$ simultaneously in a single forward pass. This approach guarantees that the output is strictly valid (i.e., exists within the known candidate set) while maintaining the semantic consistency enforced during training.

2.1 Offline Structural Indexing

Prior to inference, we construct a dense vector index of the Knowledge Graph. This is a one-time, offline process utilizing the Graph-Adapter.

For every unique entity $u \in \mathcal{C}$, we linearize its structural neighborhood (or representative title features) into a sequence X_u and pass it through the encoder. We extract the resulting structural representation to form a static index $\mathcal{M}$ (Equation 8):

$$\mathcal{M}[u] = f_{graph}(X_u) \tag{8}$$

This map provides the "ground truth" semantic vectors against which text generation will be reranked, ensuring that the predicted entity aligns with its structured representation in the graph.

2.2 Parallel Linguistic Evaluation

During inference, the model operates primarily in LoRA-Text mode. The input query is prepared with a mask buffer of fixed length N at the target entity position.

Unlike autoregressive models that generate tokens sequentially, Scent evaluates the likelihood of the entire candidate vocabulary $\mathcal{V}$ across all N mask positions in parallel. We employ a highly optimized tensor "gather" operation to extract specific log-probabilities corresponding to the pre-tokenized sequences of the candidate set.

For a candidate c, we append the special `<end_of_title>` token to its token sequence, resulting in $[t_1, t_2, \ldots, t_k, t_{end}]$. The linguistic score S_{ling} is calculated as the length-normalized sum of log-probabilities for the full sequence, rewarding the model for correctly predicting both the entity content and its termination point:

$$S_{ling}(c) = \frac{1}{k} \sum_{j=1}^{k} \log P(t_j \mid X_{text}) \tag{9}$$

This vectorization allows Scent to evaluate thousands of candidates instantly, with the constraint being that entity token sequences must fit within the mask buffer N.

2.3 Cross-Modal Alignment and Re-ranking

To ensure the generated entity is not only linguistically fluent but also topologically consistent with the knowledge graph, we refine the initial linguistic scores using the graph alignment objective. In the same forward pass used for text generation, the model projects a predicted structural representation $\hat{\mathbf{v}}$ from the hidden state of the `<node_start>` token. This vector represents the model's expectation of the entity's graph position given the textual context.

It is important to note that despite utilizing vector similarity, this stage does not constitute a standard dense vector retrieval task. In a typical dense retrieval setting, the query vector would be compared against the entire index $\mathcal{M}$ (which could contain millions of entities) to find the nearest neighbors.

Instead, Scent employs a **cascade re-ranking strategy**. We utilize the linguistic scores S_{ling} as a high-recall filter to prune the search space, selecting only a subset $\mathcal{C}_{top}$ consisting of the top-k most likely textual candidates (e.g., the top 50 or 100 matches). The structural verification is computed exclusively for this reduced subset.

For each candidate $c \in \mathcal{C}_{top}$, we retrieve its pre-computed embedding $\mathcal{M}[u_c]$ from the index and calculate the final score as a weighted combination of the linguistic probability and the cosine similarity between the predicted and actual graph vectors (Equation 10):

$$S_{total}(c) = S_{ling}(c) + \alpha \cdot \cos(\hat{\mathbf{v}}, \mathcal{M}[u_c]) \tag{10}$$

3 Preliminary Experiments

We present a preliminary evaluation of the Scent framework. The primary objective of these experiments is to verify the hypothesis that grounding the generation in structured graph knowledge alters the semantic distribution of the output. These experiments represent a "proof-of-concept" conducted under constrained compute resources (limited to 3 training epochs) and serve as a baseline for future optimization.

3.1 Experimental Setup

Architecture & Backbone: We utilize RoBERTa[1] as the encoder backbone. To maintain computational efficiency, the backbone parameters are frozen. We train the model using Low-Rank Adaptation (LoRA) [2].

Datasets:

- **Knowledge Graph:** We construct our graph structure using **YAGO4** [4], leveraging its rich taxonomy and rigorous type constraints.
- **Entity Linking:** Evaluation is performed on the **AIDA-YAGO2** [5] dataset. We map the mentions in AIDA-YAGO2 to their corresponding nodes in the YAGO4 graph to create the ground truth targets.

Configurations: We compare two variations of our framework to conduct an ablation study on the impact of the structural grounding:

1. **Scent (Text-Only):** This configuration utilizes the **Text-Adapter** (f_{text}) and the specific fixed-buffer masking strategy required for constrained generation. It is trained solely with the Masked Language Modeling objective ($\mathcal{L}_{text}$). This baseline relies exclusively on the linguistic patterns available in the pre-trained encoder and the textual context.

2. **Scent (Text + Graph):** The full framework, utilizing both the **Text-Adapter** and **Graph-Adapter** (f_{graph}). It is trained with the multi-task objectives: Text Generation ($\mathcal{L}_{text}$), Feature Regression ($\mathcal{L}_{graph}$), and Cross-Modal Alignment ($\mathcal{L}_{align}$).

3.2 Quantitative Results

Table 1 summarizes the performance on the AIDA-YAGO2 test set (9,272 samples).

Table 1: Preliminary Accuracy on AIDA-YAGO2 (3 Epochs)

Model Configuration	Accuracy@1	Accuracy@5
Scent (Text-Only)	**25.55%**	**40.00%**
Scent (Text + Graph)	17.89%	31.74%

Observation: In this preliminary phase, the Scent (Text-Only) configuration outperforms the structure-grounded version in raw accuracy. However, we hypothesize that this gap arises not from a failure of the graph module, but rather from an aggressive "semantic gravity" exerted by the graph embeddings during the alignment process, which we analyze in the qualitative section below.

3.3 Qualitative Analysis: The "Semantic Field" Effect

To understand why the introduction of the graph module reduces exact-match accuracy while preserving potential utility, we examine the prediction dynamics for the query:

"The capital of Italy, `<predict>`, is known for its history." (Target: Rome)

The Top-5 predictions for both configurations are detailed in Table 2.

Table 2: Top-5 Predictions and Scores for Scent variants

Rank	Scent (Text-Only)	Score	Scent (Text + Graph)	Score
1	**Rio** (Rio de Janeiro)	-2.57	**Italian** (Nat. Football Team)	-2.37
2	**Berri** (Nabih Berri)	-2.93	**Italian** (Italy - Country)	-2.37
3	**Rome** (Correct Target)	-3.00	**Italy** (Italy - Country)	-2.39
4	**Porto** (F.C. Porto)	-3.10	**European** (Europe)	-2.81
5	**Para** (Pará)	-3.11	**European** (Champions League)	-2.81

Interpretation: The Scent (Text-Only) model relies on linguistic likelihoods. While it successfully retrieves "Rome" (Rank 3), the surrounding candidates (Rio, Berri, Porto) are semantically disparate; they are proper nouns that fit the syntactic slot but lack specific relevance to "Italy".

Conversely, **Scent (Text + Graph)** displays a distinct **topological bias**. Although it fails to rank "Rome" in the top 5, every retrieved entity is tightly clustered in the Knowledge Graph neighborhood of the context entity "Italy":

- It retrieves "Italy" (self-reference) and the "Italian National Team."
- It retrieves the super-class concept "European."

This suggests that the graph constraints are working *too* effectively. The alignment loss ($\mathcal{L}_{align}$) forces the prediction to reside so close to the semantic centroid of "Italy" that the model currently struggles to distinguish the *part-of* relationship (Rome is part of Italy) from *similarity* relationships (Italy is similar to Italian Team). The model has successfully located the correct "neighborhood" in the vector space but currently lacks the resolution to pinpoint the specific node within that cluster.

3.4 Discussion and Future Work

The experiments demonstrate that the Scent framework successfully injects structured knowledge into the generation process. Even with limited training, the graph module effectively constrains the output to the relevant semantic subgraph, filtering out syntactically plausible but semantically irrelevant hallucinations (e.g., "Nabih Berri" or "Rio").

However, the current optimization landscape leads to a trade-off between semantic consistency and precision. We identify three key areas for the next phase of development:

1. **Refining Cross-Modal Alignment:** The current alignment objective appears overly rigid, often collapsing the representation of a specific entity (Rome) into the representation of its primary context (Italy). We plan to investigate improved alignment strategies that allow for fine-grained entity resolution within the correct semantic neighborhood.
2. **Mitigating Representation Collapse:** As noted in our methodology, the Graph-Adapter is susceptible to representation collapse. Our primary goal is to identify a **hard constraint** (e.g., geometric orthogonality) that physically prevents feature collapse. In the interim, if soft constraints are retained, we must stabilize the decorrelation objective to ensure consistent optimization dynamics.
3. **Extended and Robust Training:** Given the complexity of the multi-task objective, the current 3-epoch regimen is insufficient for the model to resolve the competing gradients of the text and graph heads. We intend to scale the training duration and refine the optimization schedule to ensure full convergence.

References

[1] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR, abs/1907.11692*.

[2] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. *CoRR, abs/2106.09685*.

[3] Kim, J., Nguyen, T. D., Min, S., Cho, S., Lee, M., Lee, H., & Hong, S. (2022). Pure Transformers are Powerful Graph Learners. *arXiv preprint arXiv:2207.02505*.

[4] Pellissier Tanon, T., Weikum, G., & Suchanek, F. (2020). YAGO 4: A Reason-able Knowledge Base. In *The Semantic Web* (LNCS 12123, pp. 583–596). Springer.

[5] Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., & Weikum, G. (2011). Robust Disambiguation of Named Entities in Text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 782–792.

[6] Zbontar, J., Jing, L., Misra, I., LeCun, Y., & Deny, S. (2021). Barlow Twins: Self-Supervised Learning via Redundancy Reduction. *arXiv preprint arXiv:2103.03230*.