SMARTe: Slot-based Method for Accountable Relational Triple extraction

Anonymous ACL submission

Abstract

Relational Triple Extraction (RTE) is a fundamental task in Natural Language Processing (NLP). However, prior research has primarily focused on optimizing model performance, with limited efforts to understand the internal mechanisms driving these models. Many existing methods rely on complex preprocessing to induce specific interactions, often resulting in opaque systems that may not fully align with their theoretical foundations. To address these limitations, we propose SMARTe: a Slot-based Method for Accountable Relational Triple extraction. SMARTe introduces intrinsic interpretability through a slot attention mechanism and frames the task as a set prediction problem. Slot attention consolidates relevant information into distinct slots, ensuring all predictions can be explicitly traced to learned slot representations and the tokens contributing to each predicted relational triple. While emphasizing interpretability, SMARTe achieves performance comparable to state-of-the-art models. Evaluations on the NYT and WebNLG datasets demonstrate that adding interpretability does not compromise performance. Furthermore, we conducted qualitative assessments to showcase the explanations provided by SMARTe, using attention heatmaps that map to their respective tokens. We conclude with a discussion of our findings and propose directions for future research.

1 Introduction

Relational Triple Extraction (RTE) is a well-established and widely studied task in Natural Language Processing (NLP). Its primary objective is to automatically extract structured information such as names, dates, and relationships from unstructured text, thereby enhancing data organization and accessibility (Nayak et al., 2021). These extracted relationships are represented as relational triples consisting of `(Subject, Relation, Object)`. For example, the sentence *"Barack Obama is the President of America"* can be expressed as the triple `(Barack Obama, Head of, America)`. Such structured representations underpin a range of downstream applications, including knowledge graph construction, question answering, and information retrieval. RTE is often confused with Joint Entity and Relation Extraction (JERE) (Zhang et al., 2017; Gupta et al., 2016; Miwa and Sasaki, 2014), or the terms are used interchangeably by some scholars (Sui et al., 2023; Li et al., 2021). However, there is a key distinction: JERE addresses both entity identification and relation extraction, which require annotation of all entities in a text, regardless whether they participate in a relationship. In contrast, RTE focuses exclusively on entities that are part of a relationship, specifically a *set of annotated relational triples*. In this paper, we focus **solely on RTE**.

Traditional approaches to RTE often prioritize performance metrics such as precision, recall, and F1 score, with limited consideration for interpretability. According to a comprehensive survey by Zhao et al. (2024), we found that many existing methods introduce novel architectures, claiming that these designs facilitate specific interactions that enhance model performance. However, such claims are frequently supported solely by empirical results, with minimal analysis or explanation of the underlying mechanisms, leaving the validity of these assertions largely unexplored. Moreover, we observed that none of the reviewed methods explicitly address explainability, which highlights a significant gap in current research. In this study, we introduce **SMARTe**: a **S**lot-based **M**ethod for **A**ccountable **R**elational **T**riple Extraction, a transparent architecture designed to address interpretability in RTE. SMARTe incorporates intrinsic interpretability through the use of a slot attention mechanism. This approach ensures that every prediction from SMARTe can be explicitly traced back to its learned slot representations and the specific

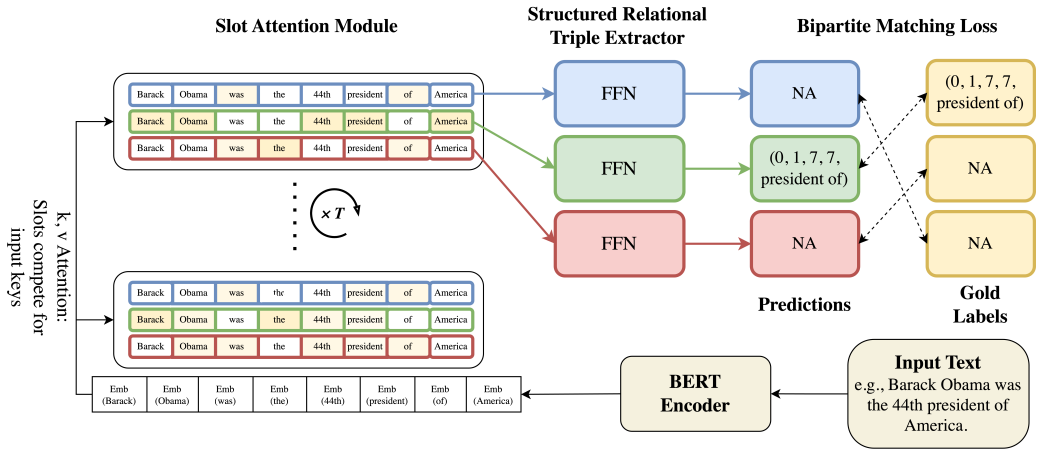

Figure 1: Architecture Diagram of our SMARTe model. The Slot Attention module iteratively refines the slots over T iterations to obtain a final representation, from which relational triples are extracted.

tokens contributing to each predicted relational triple. This results in informed and transparent outputs, embodying the ***Accountability, which directly equates to explainability***, in our model. Since the learned slots do not follow any inherent order, the RTE task is framed as a *set prediction* problem.

Our goal is to demonstrate how slot attention can be effectively applied to NLP tasks to generate meaningful explanations. By clustering relevant information, this method facilitates informed and transparent predictions. Although we illustrate our approach in the context of RTE, it is ***broadly applicable to other NLP tasks characterized by set-based target structures***. Interpretability becomes especially valuable in high-stakes scenarios where transparency and trust are paramount (Danilevsky et al., 2020). As stakeholders increasingly demand clear and interpretable explanations for AI-driven decisions, our approach effectively addresses these critical requirements. We introduce **SMARTe**, a **S**lot-based **M**ethod for **A**ccountable **R**elational **T**riple **e**xtraction. Our contributions are as follows:

- To the best of our knowledge, the proposed SMARTe framework is the first to introduce interpretability to relational triple extraction tasks, marking a significant contribution to the field of Explainable AI (XAI) in Natural Language Processing (NLP). This is achieved by adapting the slot attention mechanism, originally developed for unsupervised learning in computer vision.
- We conduct extensive experiments on two widely used datasets and demonstrate that our model achieves performance comparable to current state-of-the-art systems, all while offering interpretability.
- We provide a qualitative assessment and demonstrate how slot attention facilitates explanations which allow users to understand the model's reasoning behind its predictions.

2 Related Work

Recent advancements in relation triplet extraction (RTE) have investigated various architectures to better capture interactions between entities and relations. We group these approaches into three main categories: sequence-to-sequence (seq2seq) Methods, Tagging-Based Methods, and Pairwise-Based Methods.

Seq2seq methods treat triples as token sequences, leveraging an encoder-decoder framework akin to machine translation. CopyRE (Zeng et al., 2018) uses a copy mechanism to generate relations and entities but struggles with multi-token entities. CopyMTL (Zeng et al., 2020) addresses this by employing a multi-task learning framework. CGT (Ye et al., 2021) introduces a generative transformer with contrastive learning to enhance long-term dependency and faithfulness. R-BPtrNet (Chen et al., 2021) uses a binary pointer network to extract explicit and implicit triples, while SPN (Sui et al., 2023) reframes relational triple extraction as a set prediction problem, utilizing a non-autoregressive decoder with iterative refinement for improved contextual representation.

Tagging-Based Methods, also known as sequence labeling methods, utilize binary tagging sequences to identify the start and end positions of

entities and, in some cases, to determine relations. Early approaches, such as NovelTagging (Zheng et al., 2017), introduced a tagging-based framework that reformulates joint extraction as a tagging problem, enabling direct extraction of entities and their relations. CasRel (Wei et al., 2019) improves on this by first identifying all potential head entities and then applying relation-specific sequence taggers to detect corresponding tail entities. Recent methods, however, are not entirely sequence-based. For instance, BiRTE employs tagging only for entities. It employs a pipeline strategy in which entities are tagged as either subjects or objects, after which all possible subject-object pairs are generated and classified using a biaffine scorer. Similarly, PRGC (Zheng et al., 2021) includes a component to predict potential relations, constraining subsequent entity recognition to the predicted relation subset. PRGC's strong performance stems from its use of a global correspondence table, which effectively captures interactions between token pairs.

Pairwise-Based Methods focus on enhancing token-pair interaction representations to improve relation classification. These methods eliminate the need to explicitly predict head or tail entities, as token pairs classified as `NA` (no relationship) are directly discarded. Notably, approaches in this category have achieved state-of-the-art (SOTA) F1 performance. Early works often framed these methods as table-filling approaches. For example, GraphRel (Fu et al., 2019) models entity-relation interactions through a relation-weighted Graph Convolutional Network. TPLinker (Wang et al., 2020) redefines triple extraction as a token-pair linking task, utilizing a relation-specific handshaking tagging scheme to align boundary tokens of entity pairs. Similarly, PFN (Yan et al., 2021) introduces a partition filter network that integrates task-specific feature generation to simultaneously model entity recognition and relation classification. Modern SOTA techniques further refine these approaches. UniRel (Tang et al., 2022) employs a unified interaction map to effectively capture token-relation interactions and also incorporating relation-specific token information into the prediction process. Similarly, DirectRel (Shang et al., 2022) reformulates relational triple extraction (RTE) as a bipartite graph linking task, focusing on generating head-tail entity pairs.

Our approach builds on the seq2seq paradigm, eliminating the need to permute token pairs. Slot attention also disentangles relational triplets into distinct slots, delivering superior performance over the earlier seq2seq and tagging-based models. It is also competitive to pairwise-based methods while maintaining computational efficiency. Most importantly, the RTE field has given limited attention to explainability. **Existing research primarily focuses on optimizing performance metrics such as F1 scores and surpassing benchmarks, often achieving only marginal gains (typically less than 1%, as seen in newer research)**. Since benchmark datasets already exhibit performance exceeding 90%, this raises questions about their practical relevance. The latest SOTA models (usually based on pairwise methods) lack interpretability, which stems from their prediction mechanisms that classify relationships by evaluating token pairs in a pairwise manner. Since these models rely on token representations derived from fine-tuned BERT models, the surrounding context is already embedded within these token pairs, making it **difficult to trace the specific context each pair has "absorbed."**. For example, given the sentence *"Barack Obama is the President of America,"* these models predict the triple `(Barack Obama, Head of, America)` by relying solely on the representations of `Barack Obama` and `America` to classify the relationship `Head of`. The token `president` does not contribute to the prediction, as its information is absorbed into the representations of the paired tokens. In contrast, our model employs slot attention to directly learn relational triple representations. This method allows us to explicitly identify the tokens contributing to each slot, which corresponds to a relational triple prediction. By focusing on slot-level representations rather than pairwise token interactions, our approach enhances interpretability, as it provides a clearer understanding of how individual tokens influence the final predictions from the slot attention mechanism as shown in Appendix A.

3 Approach

The architecture of our SMARTe model is depicted in Figure 1, consisting of three primary components: an encoder, a slot attention module and a structured relational triple extractor.

3.1 Encoder

The encoder transforms the input text into dense, contextual representations that capture the semantic and syntactic information necessary for downstream processing. In our implementation, we utilize pre-trained transformer-based models BERT-

Base-Cased (Devlin et al., 2019). The process begins with tokenizing the input text into subword units, which are then fed into the transformer. The output is a sequence of contextualized embeddings, each corresponding to a token in the input text. Given an input text sequence $X = x_1, x_2, \ldots, x_n$, where x_i represents the i-th token, the encoder transforms this sequence into a sequence of contextualized embeddings:

$$\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\} = \text{Encoder}(X). \quad (1)$$

Here, $\mathbf{H} \in \mathbb{R}^{n \times d}$, where n is the length of a sequence of contextualized embeddings (including `[CLS]` and `[SEP]`, two special start and end markers), with each $\mathbf{h}_i$ corresponding to a token in the input text, and d is the embedding size produced by the encoder for each token.

3.2 Slot Attention Module

Slot Attention is a neural network module designed for object-centric representation learning (Locatello et al., 2020), enabling a model to decompose a scene into distinct entities or objects. It uses iterative attention mechanisms to map input data (like image features) to a fixed number of slots, which are learnable feature vectors representing objects or parts of the input. This approach is highly effective in unsupervised learning settings, as it disentangles objects without requiring explicit annotations. The general form of Slot Attention is as follow:

$$\mathbf{Q}^{(l)} = \mathbf{Z}^{(l)}\mathbf{W}_Q, \mathbf{K} = \mathbf{X}\mathbf{W}_K, \mathbf{V} = \mathbf{X}\mathbf{W}_V \quad (2)$$

$$\mathbf{A}^{(l)} = \text{normalize}(\text{softmax}(\mathbf{Q}^{(l)}\mathbf{K}^\top)) \quad (3)$$

$$\mathbf{Z}^{(l+1)} = \text{GRU}(\mathbf{Z}^{(l)}, \mathbf{A}^{(l)}\mathbf{V}). \quad (4)$$

At iteration l, the current slots $\mathbf{Z}^{(l)}$ are transformed into query vectors $\mathbf{Q}^{(l)}$ using a learnable weight matrix $\mathbf{W}_Q$, while the input features $\mathbf{X}$, derived from the sequence of contextualized embeddings from the encoder $\mathbf{H}$, are projected into key $\mathbf{K}$ and value $\mathbf{V}$ vectors using $\mathbf{W}_K$ and $\mathbf{W}_V$, respectively. Attention weights $\mathbf{A}^{(l)}$ are computed as the dot product between $\mathbf{Q}^{(l)}$ and $\mathbf{K}^\top$, followed by softmax to produce a probability distribution that ensures each slot attends to specific parts of the input. The slots are updated using a Gated Recurrent Unit (GRU) (Chung et al., 2014), combining the previous slot representation $\mathbf{Z}^{(l)}$ with the weighted aggregation of values $\mathbf{A}^{(l)}\mathbf{V}$. This iterative process refines the slots, allowing the model to disentangle objects or entities in the input data. After completing the iterative process, the output is a refined slots $\mathbf{Z} \in \mathbb{R}^{k \times d}$, where k is the number of slots and d is the dimensionality of each slot.

However, the softmax function in slot attention can be too restrictive for relational triple extraction tasks, particularly when certain tokens, such as "Barack Obama," are involved in multiple triples and need to be associated with multiple slots. For instance, in the triples `(Barack Obama, Born In, Hawaii)`, `(Barack Obama, President Of, United States)`, and `(Barack Obama, Married To, Michelle Obama)`, the token `Barack Obama` serves as the subject in each case. Softmax enforces a normalization constraint where attention scores must sum to 1, causing the token's contribution to be distributed across slots. This can dilute its impact and make it difficult for the model to maintain consistent associations across triples. To address this limitation, our SMARTe model employ the optimal transport variant outlined in Zhang et al. (2023), which is more relaxed and provides a flexible framework for assigning tokens to slots while preserving their relevance across multiple contexts. Specifically, this involves replacing equation 3 with the optimal transport algorithm (Villani et al., 2009), as follows:

$$\mathbf{C}' = \frac{\mathbf{Q} \cdot \mathbf{K}}{\|\mathbf{Q}\|\|\mathbf{K}\|} \quad (5)$$

$$\text{MESH}(\mathbf{C}) = \underset{\mathbf{C}' \in \mathcal{V}(\mathbf{C})}{\arg\min}\, H(\text{sinkhorn}(\mathbf{C}')) \quad (6)$$

$$\mathbf{A} = \text{sinkhorn}(\text{MESH}(\mathbf{C})). \quad (7)$$

In equation 5, the initial transport cost $\mathbf{C}'$ is computed as the cosine similarity between $\mathbf{Q}$ and $\mathbf{K}$. Next, in equation 6, $\text{MESH}(\mathbf{C})$ selects, from the set $\mathcal{V}(\mathbf{C})$, the candidate matrix $\mathbf{C}'$ that minimizes the entropy of the initial cost function, $H\big(\text{sinkhorn}(\mathbf{C}')\big)$. Here, $\text{sinkhorn}(\cdot)$ iteratively normalizes the row and column sums to produce a doubly stochastic matrix. $\text{MESH}(\cdot)$ corresponds to Minimizing the Entropy of Sinkhorn. Finally, in equation 7, the chosen matrix $\text{MESH}(\mathbf{C})$ is passed once more through the Sinkhorn operator to yield the final attention matrix $\mathbf{A}$. This "transport plan" preserves row- and column-stochastic constraints, providing a refined set of attention weights. For more details, we refer readers to the work of Zhang et al. (2023). We have also provided the results and analysis for the softmax variant for reference.

3.3 Structured Relational Triple Extractor

In the original Slot Attention paper (Locatello et al., 2020), the approach was designed for fixed-size images. However, when applied to text, variable input lengths make direct coordinate prediction more challenging. To address this, we reformulate coordinate prediction as a *one-hot sequence labeling task* to identify the start and end positions. We perform matrix multiplication between the slot attention outputs and the sequence tokens to encode information from the slot attention representation into each token. A feedforward neural network with linear layers predicts the indices for entity head-tail pairs, along with their corresponding relationships, in a unified manner. Specifically, it identifies five key components: `[subject-start (ss), subject-end (se), object-start (os), object-end (oe), and relationship (rs)]`, which collectively form a structured relational triple:

$$\mathbf{P}_i^{\text{ss}} = \sigma(\mathbf{V}_1^\top \tanh(\mathbf{W}_1\mathbf{Z}_i + \mathbf{W}_2\mathbf{H})) \quad (8)$$

$$\mathbf{P}_i^{\text{se}} = \sigma(\mathbf{V}_2^\top \tanh(\mathbf{W}_3\mathbf{Z}_i + \mathbf{W}_4\mathbf{H})) \quad (9)$$

$$\mathbf{P}_i^{\text{os}} = \sigma(\mathbf{V}_3^\top \tanh(\mathbf{W}_5\mathbf{Z}_i + \mathbf{W}_6\mathbf{H})) \quad (10)$$

$$\mathbf{P}_i^{\text{oe}} = \sigma(\mathbf{V}_4^\top \tanh(\mathbf{W}_7\mathbf{Z}_i + \mathbf{W}_8\mathbf{H})) \quad (11)$$

$$\mathbf{P}_i^{\text{rs}} = \sigma(\mathbf{W}_t\mathbf{Z}_i). \quad (12)$$

$\mathbf{P}_i^*$ refers to the prediction for each component at the ith slot, with $\mathbf{Z}_i$ representing the ith slot embedding; the matrices $\mathbf{W}_t \in \mathbb{R}^{t\times d}$, $[\mathbf{W}_i]_{i=1}^8 \in \mathbb{R}^{d\times d}$, where t represents the total number of relation types in the RTE task and $[\mathbf{V}_i]_{i=1}^4 \in \mathbb{R}^d$ are learnable parameters. The function σ refers to the `softmax` operation and $\mathbf{H}$ represents the sequence of contextualized embeddings.

Since the slots inherently lack ordering, directly comparing predictions with the ground truth during model training presents a challenge. To overcome this, the predictions must first be aligned with the ground truth. This alignment is achieved using the Hungarian matching algorithm (Kuhn, 1955). After optimally matching the predictions to the ground truth, cross-entropy loss is applied to each component to train the model:

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{k} \{ & -\log \mathbf{P}_{\pi^*}^{\text{rs}}(\mathbf{T}_i^{\text{rs}}) + \mathbb{1}_{\{\mathbf{T}_i^{\text{rs}} \neq \texttt{NA}\}} \\ & [-\log \mathbf{P}_{\pi^*}^{\text{ss}}(\mathbf{T}_i^{\text{ss}}) - \log \mathbf{P}_{\pi^*}^{\text{se}}(\mathbf{T}_i^{\text{se}}) \\ & - \log \mathbf{P}_{\pi^*}^{\text{os}}(\mathbf{T}_i^{\text{os}}) - \log \mathbf{P}_{\pi^*}^{\text{oe}}(\mathbf{T}_i^{\text{oe}})]\} \end{aligned} \quad (13)$$

where π^* represents the optimal assignment that minimizes the total pairwise matching cost and $\mathbf{P}_{\pi(i)}^*$ denotes the prediction, $\mathbf{T}_i^*$ represents the corresponding ground truth target at the ith slot and the indicator function $\mathbb{1}_{\{\mathbf{T}_i^{\text{rs}} \neq \texttt{NA}\}}$ ensuring only valid relational triples contribute to the loss.

4 Experimental Setup

We evaluate our model using two widely recognized benchmark datasets frequently employed in the existing literature on relational triple extraction: the New York Times (NYT) dataset (Riedel et al., 2010) and the Web Natural Language Generation (WebNLG) dataset (Gardent et al., 2017). Both datasets include two variants of annotation: Partial Matching and Exact Matching. In Partial Matching, only the head words of the ground truth entities are annotated, whereas in Exact Matching, the entire span of the entities is annotated. Therefore, Exact Matching offers a more precise and comprehensive representation of the task. It is worth mentioning that most prior studies primarily report results based on Partial Matching, often neglecting Exact Matching. In our study, we present **results for both matching strategies** to facilitate comparisons in future research. Detailed statistics for these datasets are provided in Table 5 and 6. Please note that these are the only two benchmark datasets specifically designed for this task. Our experiments were conducted on a GeForce RTX 2080 Ti GPU (11GB VRAM) and ran with 12 random seeds (11, 29, 33, 39, 42, 53, 57, 62, 65, 73, 96, 98). Training time per epoch is approximately 7 minutes for NYT and 1 minute for WebNLG. Please refer to Appendix C for a detailed overview of the key hyperparameters used in our experiments.

5 Experimental Results

We present our experimental results for the NYT dataset in Table 1 and the WebNLG dataset in Table 2, evaluating on both partial and exact matching criteria. Additionally, Table 3 and Table 4 provide an analysis of our model's performance on overlapping patterns and varying numbers of triples for the NYT and WebNLG datasets, respectively.

Our results are benchmarked against the latest state-of-the-art (SOTA) methods from 2020 onwards, as we find comparisons with outdated models serve limited purpose beyond formality. To ensure a fair comparison, all models including ours, use **BERT-Base-Cased** as encoder, with all com-

Table 1: Precision (%), Recall (%), and F1-score (%) of SMARTe and baselines on the NYT dataset (* indicates partial matching, while exact matching is indicated without it). † represents the best-performing seed in terms of overall F1 score for reference. ± represents the standard deviation of the results across 12 runs.

Model	NYT*			NYT		
	Prec.	Rec.	F1	Prec.	Rec.	F1
CasRel (Wei et al., 2019)	89.7	89.5	89.6	90.1	88.5	89.3
TPLinker (Wang et al., 2020)	91.3	92.5	91.9	91.4	92.6	92.0
CGT (Ye et al., 2021)	**94.7**	84.2	89.1	-	-	-
PRGC (Zheng et al., 2021)	93.3	91.9	92.6	93.5	91.9	92.7
R-BPtrNet (Chen et al., 2021)	92.7	92.5	92.6	-	-	-
BiRTE (Ren et al., 2022)	92.2	93.8	93.0	91.9	**93.7**	92.8
DirectRel (Shang et al., 2022)	93.7	92.8	93.2	**93.6**	92.2	**92.9**
UniRel (Tang et al., 2022)	93.5	**94.0**	**93.7**	-	-	-
SPN (Sui et al., 2023)	93.3	91.7	92.5	92.5	92.2	92.3
SMARTe (Softmax)	$92.2_{\pm 0.3}$	$91.2_{\pm 0.4}$	$91.7_{\pm 0.3}$	$92.1_{\pm 0.2}$	$91.7_{\pm 0.2}$	$91.9_{\pm 0.2}$
SMARTe (Opt Transport)	$92.4_{\pm 0.3}$	$92.9_{\pm 0.2}$	$92.6_{\pm 0.2}$	$92.5_{\pm 0.2}$	$93.0_{\pm 0.1}$	$92.7_{\pm 0.1}$
SMARTe† (Opt Transport)	92.5	93.3	92.9	92.7	93.1	**92.9**

Table 2: Precision (%), Recall (%), and F1-score (%) of SMARTe and baselines on the WebNLG dataset (* indicates partial matching, while exact matching is indicated without it). † represents the best-performing seed in terms of overall F1 score for reference. ± represents the standard deviation of the results across 12 runs.

Model	WebNLG*			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
CasRel (Wei et al., 2019)	93.4	90.1	91.8	-	-	-
TPLinker (Wang et al., 2020)	91.7	92.0	91.9	88.9	84.5	86.7
CGT (Ye et al., 2021)	92.9	75.6	83.4	-	-	-
PRGC (Zheng et al., 2021)	94.0	92.1	93.0	89.9	87.2	88.5
R-BPtrNet (Chen et al., 2021)	93.7	92.8	93.3	-	-	-
BiRTE (Ren et al., 2022)	93.2	94.0	93.6	89.0	**89.5**	89.3
DirectRel (Shang et al., 2022)	94.1	94.1	94.1	**91.0**	89.0	**90.0**
UniRel (Tang et al., 2022)	**94.8**	**94.6**	**94.7**	-	-	-
SPN (Sui et al., 2023)	93.1	93.6	93.4	-	-	-
SMARTe (Softmax)	$92.8_{\pm 0.4}$	$92.5_{\pm 0.4}$	$92.7_{\pm 0.3}$	$85.2_{\pm 0.2}$	$83.8_{\pm 0.4}$	$84.5_{\pm 0.2}$
SMARTe (Opt Transport)	$93.4_{\pm 0.4}$	$93.4_{\pm 0.3}$	$93.4_{\pm 0.3}$	$86.0_{\pm 0.4}$	$85.6_{\pm 0.4}$	$85.8_{\pm 0.3}$
SMARTe† (Opt Transport)	93.7	94.0	93.9	86.6	86.0	86.3

parative results sourced directly from their original publications. For our model, we report the **mean and standard deviation across multiple runs** with our provided seeds in our previous section. Moreover, we also include the **best-performing seed in terms of overall F1 score for reference (indicated with †)**, recognizing that many related works do not explicitly state whether their results are derived from experiments using multiple random seeds or from optimizing the F1 score with a single specific seed. Furthermore, we include results for the softmax variant as a comparison to the optimal transport approach (labeled as "Opt Transport"). Our SMARTe model achieves highly competitive performance, **coming within 1% of the top-performing methods and outperforming most previous attempts, while offering the added advantage of interpretability**. We also acknowledge that our model may exhibit a slightly higher dependency on larger datasets compared to SOTA methods as this became evident in the WebNLG exact matching benchmark, where our model's performance was comparatively lower. We attribute this to the presence of numerous poorly

Table 3: F1-score (%) on sentences with different overlapping patterns and triple numbers for NYT* test set. † represents the best-performing seed in terms of overall F1 score for reference. ± represents the standard deviation of the results across 12 runs.

Model	Normal	EPO	SEO	N=1	N=2	N=3	N=4	N≥5
CasRel (Wei et al., 2019)	87.3	92.0	91.4	88.2	90.3	91.9	94.2	83.7
TPLinker (Wang et al., 2020)	90.1	94.0	93.4	90.0	92.8	93.1	96.1	90.0
PRGC (Zheng et al., 2021)	91.0	94.5	94.0	91.1	93.0	93.5	95.5	93.0
BiRTE (Ren et al., 2022)	91.4	94.2	94.7	91.5	93.7	93.9	95.8	92.1
DirectRel (Shang et al., 2022)	**91.7**	94.8	94.6	**91.7**	94.1	93.5	96.3	92.7
UniRel (Tang et al., 2022)	91.6	**95.2**	**95.3**	91.5	**94.3**	**94.5**	**96.6**	**94.2**
SPN (Sui et al., 2023)	90.8	94.1	94.0	90.9	93.4	94.2	95.5	90.6
SMARTe (Softmax)	$90.3_{\pm 0.3}$	$93.4_{\pm 0.4}$	$92.9_{\pm 0.4}$	$90.3_{\pm 0.3}$	$93.1_{\pm 0.4}$	$93.3_{\pm 0.4}$	$95.5_{\pm 0.5}$	$85.3_{\pm 1.3}$
SMARTe (Opt Transport)	$90.7_{\pm 0.4}$	$94.5_{\pm 0.3}$	$94.3_{\pm 0.3}$	$90.6_{\pm 0.4}$	$93.6_{\pm 0.3}$	$94.4_{\pm 0.5}$	$96.2_{\pm 0.4}$	$90.7_{\pm 0.8}$
SMARTe† (Opt Transport)	90.7	95.1	94.7	90.7	94.2	94.3	96.5	91.4

Table 4: F1-score (%) on sentences with different overlapping patterns and triple numbers for WebNLG* test set. † represents the best-performing seed in terms of overall F1 score for reference. ± represents the standard deviation of the results across 12 runs. *Note: UniRel results are not available for this dataset.

Model	Normal	EPO	SEO	N=1	N=2	N=3	N=4	N≥5
CasRel (Wei et al., 2019)	89.4	94.7	92.2	89.3	90.8	94.2	92.4	90.9
TPLinker (Wang et al., 2020)	87.9	95.3	92.5	88.0	90.1	94.6	93.3	91.6
PRGC (Zheng et al., 2021)	90.4	95.9	93.6	89.9	91.6	95.0	94.8	92.8
BiRTE (Ren et al., 2022)	90.1	94.3	**95.9**	90.2	**92.9**	95.7	94.6	92.0
DirectRel (Shang et al., 2022)	**92.0**	**97.1**	94.5	**91.6**	92.2	96.0	**95.0**	**94.9**
SPN (Sui et al., 2023)	-	-	-	89.5	91.3	96.4	94.7	93.8
SMARTe (Softmax)	$89.3_{\pm 0.5}$	$90.2_{\pm 1.2}$	$93.3_{\pm 0.3}$	$89.1_{\pm 0.5}$	$91.3_{\pm 0.8}$	$95.1_{\pm 0.6}$	$93.9_{\pm 0.5}$	$92.9_{\pm 0.7}$
SMARTe (Opt Transport)	$90.5_{\pm 0.9}$	$90.1_{\pm 1.0}$	$94.0_{\pm 0.3}$	$90.1_{\pm 0.8}$	$91.8_{\pm 0.7}$	$96.0_{\pm 0.5}$	$94.8_{\pm 0.5}$	$93.2_{\pm 0.7}$
SMARTe† (Opt Transport)	90.6	90.6	94.5	90.0	92.4	**96.9**	**95.0**	93.8

defined relationships in the dataset, as illustrated in Figure 11, many of which have fewer than 10 training examples. Additionally, it is also important to note that many previous models were not benchmarked against this variant, making the comparison less conclusive. Furthermore, our analysis shows that the optimal transport algorithm consistently outperforms the softmax approach, particularly in handling overlapping patterns, where the softmax approach shows significantly weaker performance. These findings justify the integration of optimal transport into our model for enhanced performance.

6 Qualitative Analysis of Explanations

The qualitative analysis of our SMARTe model demonstrates its capability to accurately predict relational triples while providing interpretable explanations. In Figure 2, Slot 7 successfully identifies the relational triple (`Weinstein`, `/business/person/company`, `Films`) by focusing on semantically relevant tokens, such as the subject-object pair and `founders`, which aligns directly with the relationship `/business/person/company`. Unlike other slots that generate random attention patterns and fail to predict valid relationships, Slot 7 exhibits a clear and focused attention map. This highlights its ability to isolate critical tokens necessary for prediction, validating the model's decision-making process while offering valuable insights into its reasoning.

Importantly, this reasoning extends beyond merely predicting (`Weinstein`, `founders`, `Films`). The model demonstrates a deeper understanding by inferring that `founders` is semantically tied to the relationship `/business/person/company`, capturing the

relational context effectively. Such interpretability is not an isolated occurrence; similar patterns of focused attention and explainable reasoning have been consistently observed across diverse examples. For instance, entities such as {coach, salary, chef, president, executive} are systematically grouped under the /business/person/company relationship, as shown in Figure 3.

Beyond analyzing successful predictions, we are also interested in examining explanations behind incorrect predictions. For example, in NYT* test sentence 8: *"Mary L. Schapiro, who earlier this year became the new head of NASD, was more amenable to fashioning a deal to the New York Exchange's liking than her predecessor, Robert R. Glauber."* The ground truth is (Glauber, /business/person/company, NASD), but our model predicted (Schapiro, /business/person/company, NASD). This prediction is not entirely unreasonable, as Schapiro is correctly identified as the head of NASD, making the prediction logically valid. The model's reasoning is illustrated in Figure 4, where Schapiro's association with NASD is highlighted. Hence, this demonstrates that explainability can provide valuable insights into model predictions and help identify potential ambiguities or gaps in the dataset, ultimately supporting efforts to improve data quality and annotation consistency.

For location-related relationships, such as /location/location/contains, the attention mechanism predominantly focuses on the object location, as it appears to encapsulate all the necessary information needed to predict the relational triple, as shown in Figure 5 & 6. In these instances, the model's task shifts from deriving relational meaning from the sentence to recognizing pre-existing factual associations. Since such location-based relationships typically represent fixed world knowledge, the annotations often reflect factual truths rather than information inferred from the sentence context. This underscores a key limitation in providing meaningful explanations for certain types of relationships, as they depend more on external knowledge or the training data than on contextual cues within the text.

For relationship types that are not well-trained due to the limited number of instances, such as /people/person/ethnicity and /people/ethnicity/people, as shown in the training data statistics in Figure 8, the model struggles to generate reliable explanations. Although the model correctly predicts the relationship, the generated explanations are incoherent and fail to provide meaningful insights into the model's reasoning process as seen in Figure 7.

By providing clear and interpretable explanations, our SMARTe model establishes a new benchmark in understanding and visualizing relational reasoning, particularly in areas where many existing models fail. Notably, the quality of explanations can vary based on the complexity and nature of the relationship. This is especially evident when dealing with less straightforward or sparsely represented relationships, where limited training data can hinder the model's ability to produce coherent explanations. Despite these challenges, the model's capacity to deliver meaningful insights across a wide range of relationships demonstrates its potential for improving interpretability and fostering a deeper understanding of relational extraction tasks.

7 Conclusion

In this paper, we introduce SMARTe: a Slot-based Method for Accountable Relational Triple extraction, addressing the critical need for interpretability in relational triple extraction models. By leveraging a slot attention mechanism and framing the task as a set prediction problem, SMARTe ensures that predictions are explicitly traceable to their learned representations, providing intrinsic transparency. Our experimental results on the NYT and WebNLG datasets demonstrate that SMARTe achieves performance comparable to state-of-the-art models (within 1% range), while simultaneously offering meaningful explanations through attention heatmaps. These findings underscore the feasibility of combining interpretability with effectiveness, addressing the "black-box" limitations inherent in prior approaches. The pursuit of interpretability in machine learning models is not merely an academic exercise but a pressing necessity. As NLP systems are increasingly deployed in high-stakes domains requiring accountability, such as healthcare (Loh et al., 2022), finance (Chen et al., 2023), and law (Górski and Ramakrishna, 2021), ensuring models are both effective and interpretable becomes paramount.

8 Limitations

For limitations, we acknowledge that our model's performance slightly lags behind leading models

due to its seq2seq design, which does not incorporate combinatorial token interactions like bipartite graphs or interaction tables. Future work will focus on improving token interactions to boost performance while preserving interpretability. Additionally, explanations for complex interactions remain indirect and less intuitive, highlighting differences between model and human text interpretation. To address this, we plan to explore more user-friendly explanation methods to enable users to derive clearer and more actionable insights from predictions.

References

Xun-Qi Chen, Chao-Qun Ma, Yi-Shuai Ren, Yu-Tian Lei, Ngoc Quang Anh Huynh, and Seema Narayan. 2023. Explainable artificial intelligence in finance: A bibliometric review. *Finance Research Letters*, page 104145.

Yubo Chen, Yunqi Zhang, Changran Hu, and Yongfeng Huang. 2021. Jointly extracting explicit and implicit relational triples with reasoning pattern enhanced binary pointer network. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5694–5703.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable ai for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1409–1418.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planning. In *55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 179–188. Association for Computational Linguistics (ACL).

Łukasz Górski and Shashishekar Ramakrishna. 2021. Explainable artificial intelligence, lawyer's perspective. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 60–68.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Xianming Li, Xiaotian Luo, Chenghao Dong, Daichuan Yang, Beidi Luan, and Zhen He. 2021. Tdeer: An efficient translating decoding schema for joint extraction of entities and relations. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 8055–8064.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. 2020. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538.

Hui Wen Loh, Chui Ping Ooi, Silvia Seoni, Prabal Datta Barua, Filippo Molinari, and U Rajendra Acharya. 2022. Application of explainable artificial intelligence for healthcare: A systematic review of the last decade (2011–2022). *Computer Methods and Programs in Biomedicine*, 226:107161.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1858–1869.

Tapas Nayak, Navonil Majumder, Pawan Goyal, and Soujanya Poria. 2021. Deep neural approaches to relation triplets extraction: A comprehensive survey. *Cognitive Computation*, 13(5):1215–1232.

Feiliang Ren, Longhui Zhang, Xiaofeng Zhao, Shujuan Yin, Shilei Liu, and Bochao Li. 2022. A simple but effective bidirectional framework for relational triple extraction. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 824–832.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III 21*, pages 148–163. Springer.

Yu-Ming Shang, Heyan Huang, Xin Sun, Wei Wei, and Xian-Ling Mao. 2022. Relational triple extraction: One step is enough. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4360–4366. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Dianbo Sui, Xiangrong Zeng, Yubo Chen, Kang Liu, and Jun Zhao. 2023. Joint entity and relation extraction with set prediction networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Wei Tang, Benfeng Xu, Yuyue Zhao, Zhendong Mao, Yifeng Liu, Yong Liao, and Haiyong Xie. 2022. Unirel: Unified representation and interaction for joint relational triple extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7087–7099.

Cédric Villani et al. 2009. *Optimal transport: old and new*, volume 338. Springer.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. Tplinker: Single-stage joint extraction of entities and relations through token pair linking. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1572–1582.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2019. A novel cascade binary tagging framework for relational triple extraction. *arXiv preprint arXiv:1909.03227*.

Zhiheng Yan, Chong Zhang, Jinlan Fu, Qi Zhang, and Zhongyu Wei. 2021. A partition filter network for joint entity and relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 185–197.

Hongbin Ye, Ningyu Zhang, Shumin Deng, Mosha Chen, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Contrastive triple extraction with generative transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14257–14265.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9507–9514.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 1730–1740.

Yan Zhang, David W. Zhang, Simon Lacoste-Julien, Gertjan J. Burghouts, and Cees G. M. Snoek. 2023. Unlocking slot attention by changing optimal transport costs. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 41931–41951. PMLR.

Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. 2024. A comprehensive survey on relation extraction: Recent advances and new frontiers. *ACM Computing Surveys*, 56(11):1–39.

Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Ming Xu, and Yefeng Zheng. 2021. Prgc: Potential relation and global correspondence based joint relational triple extraction. *arXiv preprint arXiv:2106.09895*.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236.

A Analysis of our SMARTe model Explanations

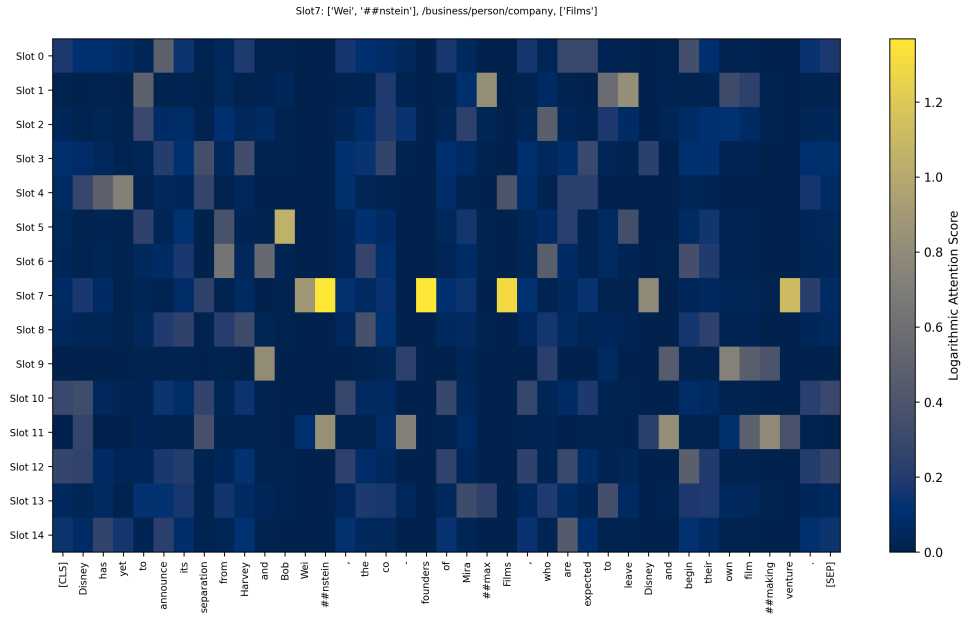

Figure 2: Visualization of the logarithmic attention scores for each token across all slots for the prediction in NYT partial matching test set sentence 429. Slot 7 successfully predicts the relational triple (`Weinstein`, `/business/person/company`, `Films`), while other slots yield no valid predictions (classified as NAs). The contributing tokens are highlighted, with the model assigning high attention scores to tokens such as its subject / object pair and `founders` which aligns with the relationship `/business/person/company`.

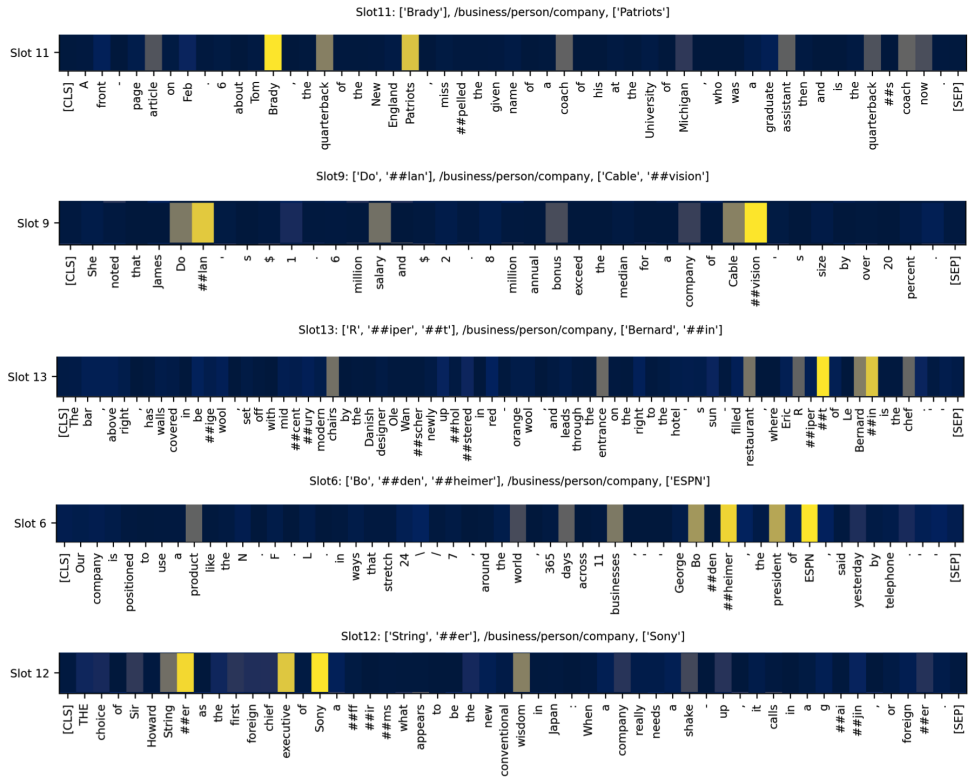

Figure 3: Snippets of explanation in NYT partial matching test set for various sentences: 188,196, 263, 349, 371 for `/business/person/company` relational triple.

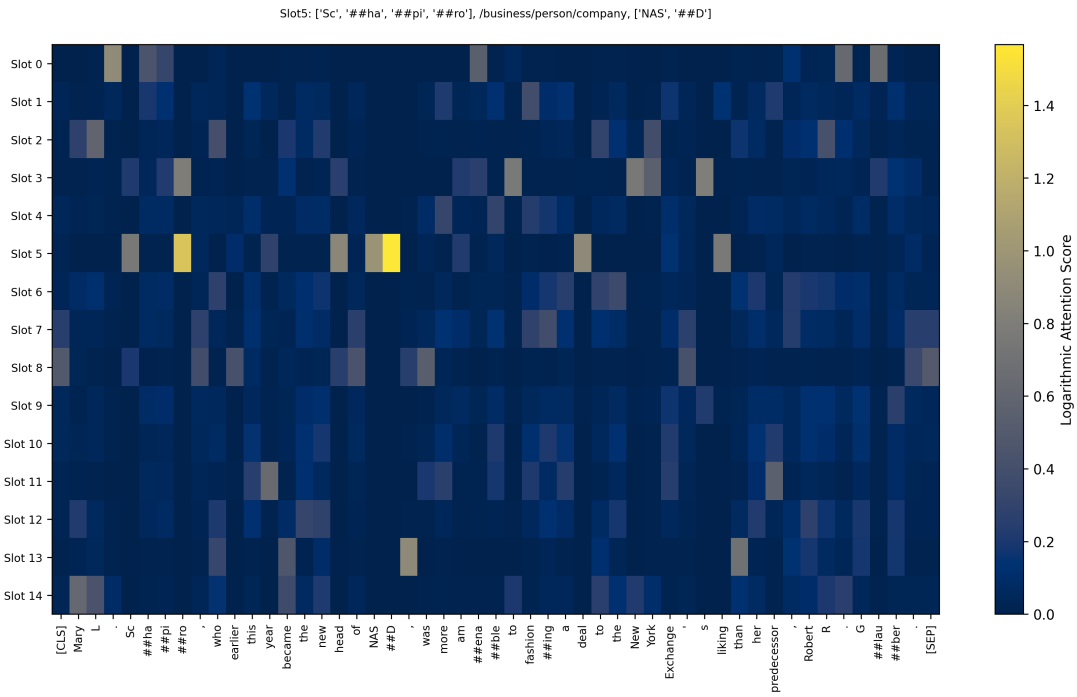

Figure 4: Visualization of incorrectly predicted relational triple for NYT partial matching test sentence 8. The golden triple for the sentence is `(Glauber, /business/person/company, NASD)`. However, our model predicted `(Schapiro, /business/person/company, NASD)`. This misprediction occurred because the model identified Schapiro as being associated with NASD in a valid context, as Schapiro held a leadership (`head`) position in the organization. Although this reasoning is correct, the prediction does not match the ground truth, leading to an incorrect result under the evaluation criteria.

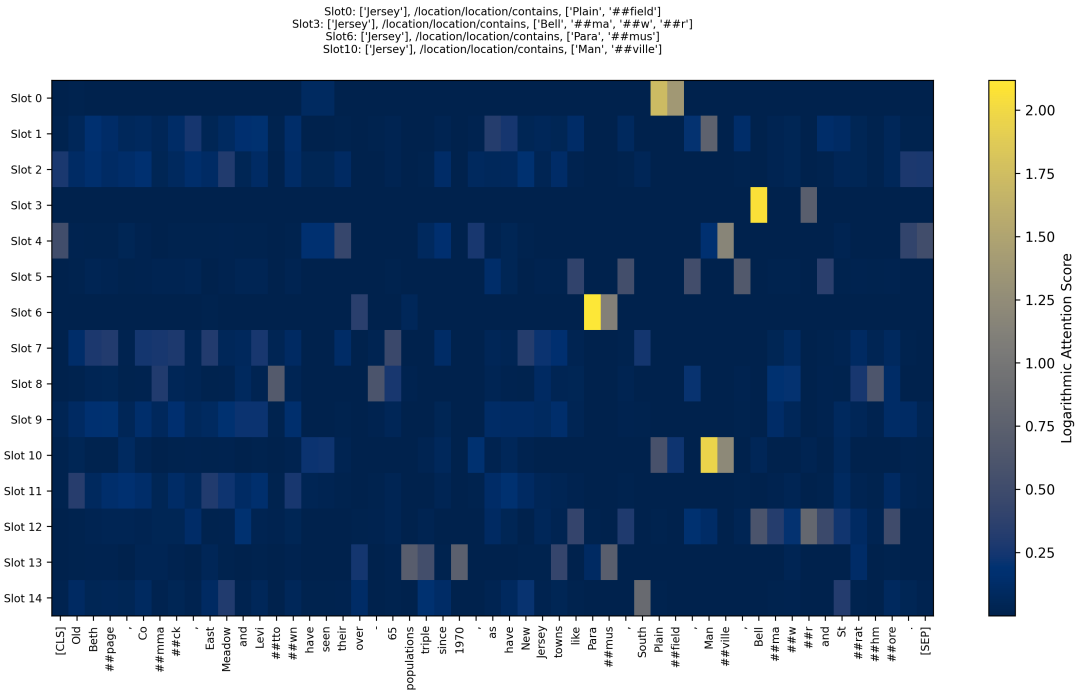

Figure 5: Explanations for NYT partial matching test sentence 279 show that the attentions primarily highlight the object location for the relationship `/location/location/contains`, where information from the object location alone is sufficient to predict the relational triple. In this sentence, Jersey contains Plainfield, Bellmawr, Paramus, and Manville.

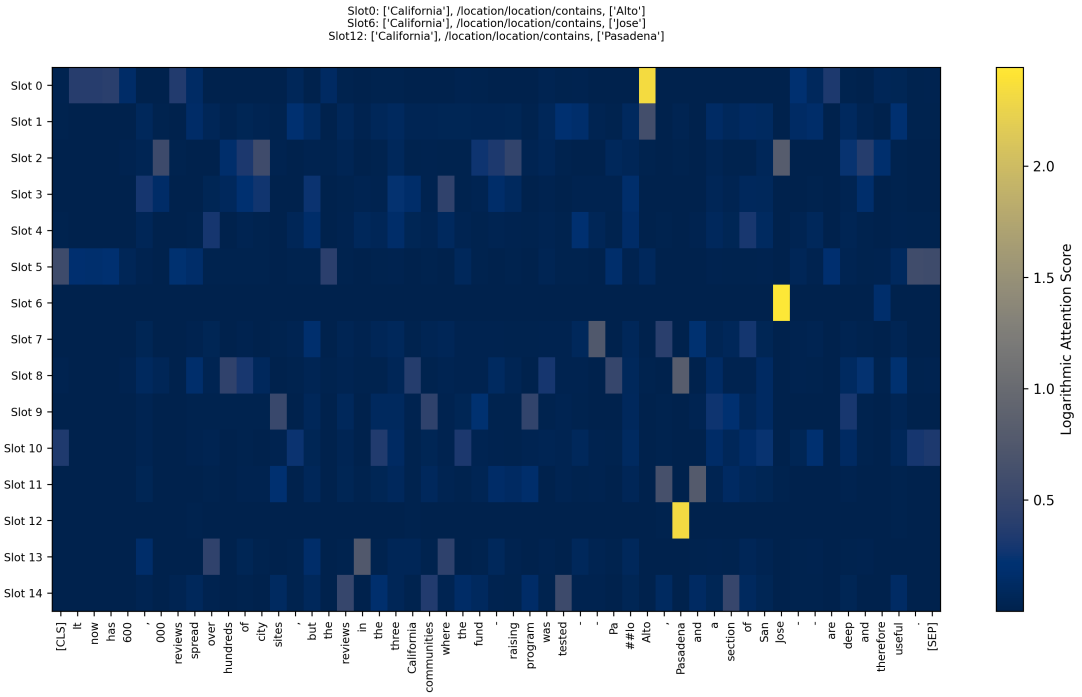

Figure 6: Explanations for NYT partial matching test sentence 325 show that the attentions primarily highlight the object location for the relationship `/location/location/contains`, where information from the object location alone is sufficient to predict the relational triple. In this sentence, California contains Alto, Jose, Pasadena.

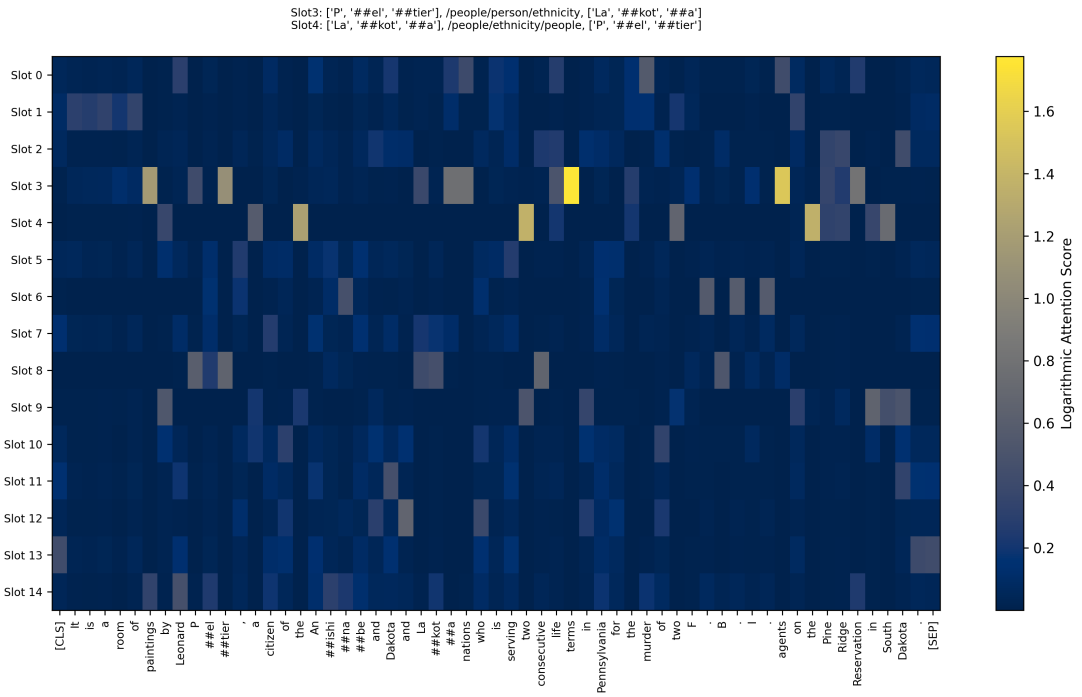

Figure 7: Explanations for NYT partial matching test sentence 411: While the model correctly predicts the relationship, the generated explanations are incoherent and lack meaningful insights into the model's reasoning process, likely due to the scarcity of training instances for these specific relationships.

B Statistics of NYT and WebNLG datasets

Table 5: Statistics of the datasets used in the study. The symbol (*) denotes datasets involving partial matching, whereas datasets without this symbol correspond to exact matching. ‡ relations not in train dataset are removed.

Category	Dataset						
	Train		Valid		Test		Relations
	Sents	Triples	Sents	Triples	Sents	Triples	
NYT*	56195	88253	4999	7976	5000	8110	24
NYT	56196	88366	5000	7985	5000	8120	24
WebNLG*	5019	11776	500	1117	703	1591	170‡
WebNLG	5019	11313	500	1224	703	1607	211‡

Table 6: Statistics of the overlapping patterns (Zeng et al., 2018) across Train, Valid, and Test sets, following prior work. Please note that this form of evaluation is only applicable to partial matching datasets.

Category	Train Set			Valid Set			Test Set		
	Normal	SEO	EPO	Normal	SEO	EPO	Normal	SEO	EPO
NYT*	37013	14735	9782	3306	1350	849	3266	1297	978
WebNLG*	1600	3402	227	182	318	16	246	457	26

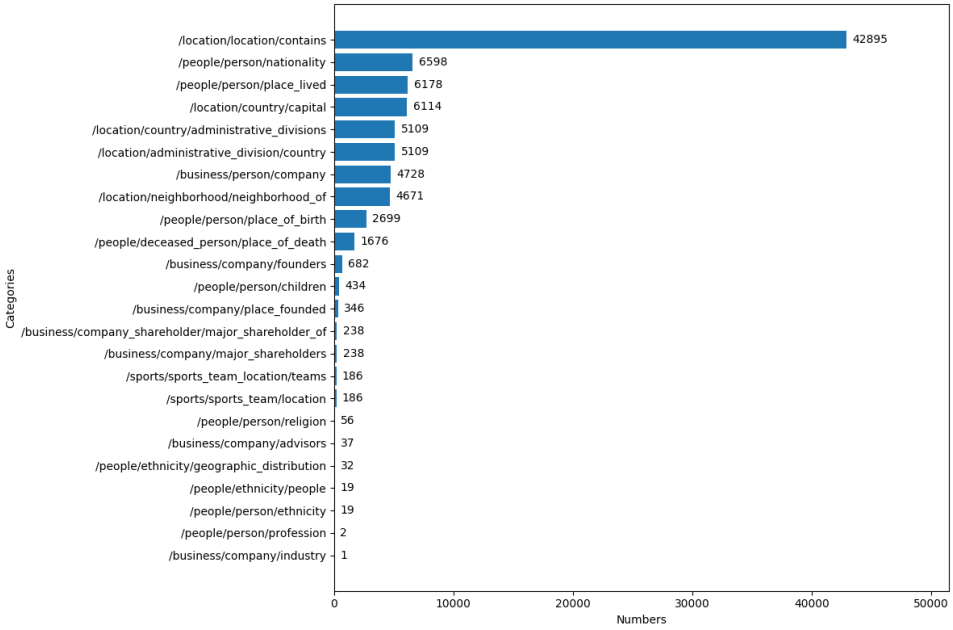

Figure 8: NYT partial matching train dataset relationship statistics

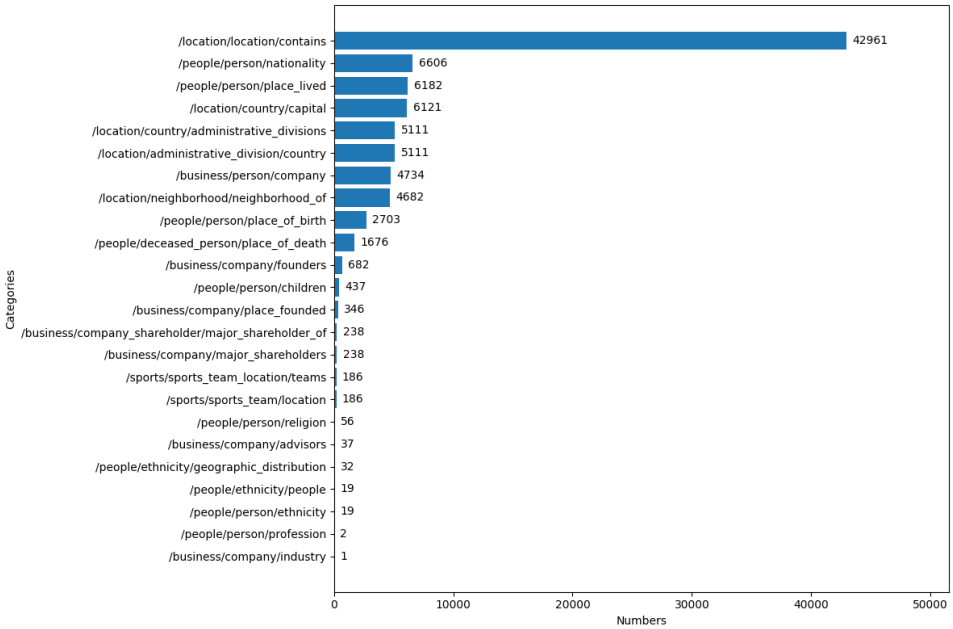

Figure 9: NYT exact matching train dataset relationship statistics

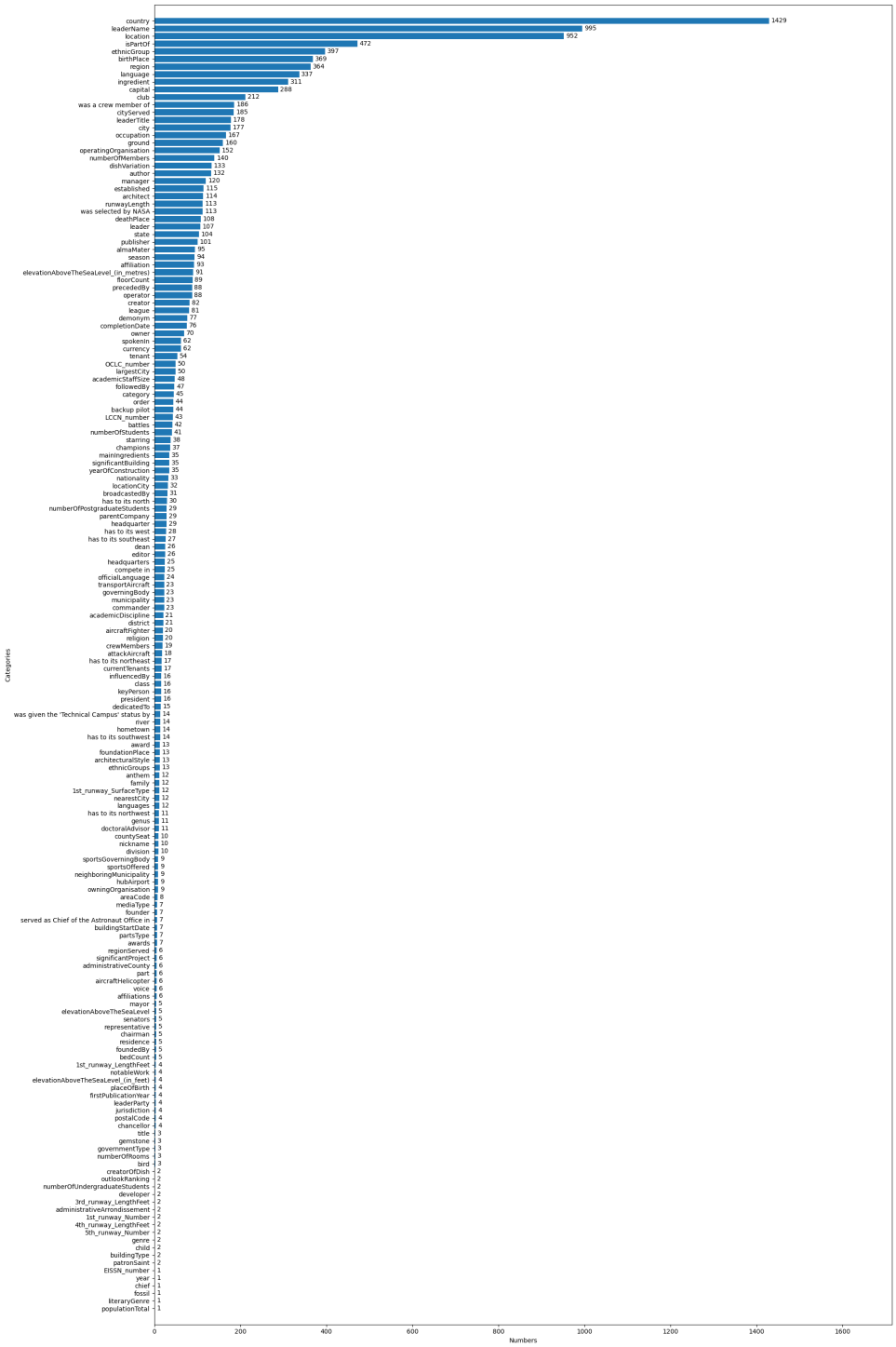

Figure 10: WebNLG partial matching train dataset relationship statistics

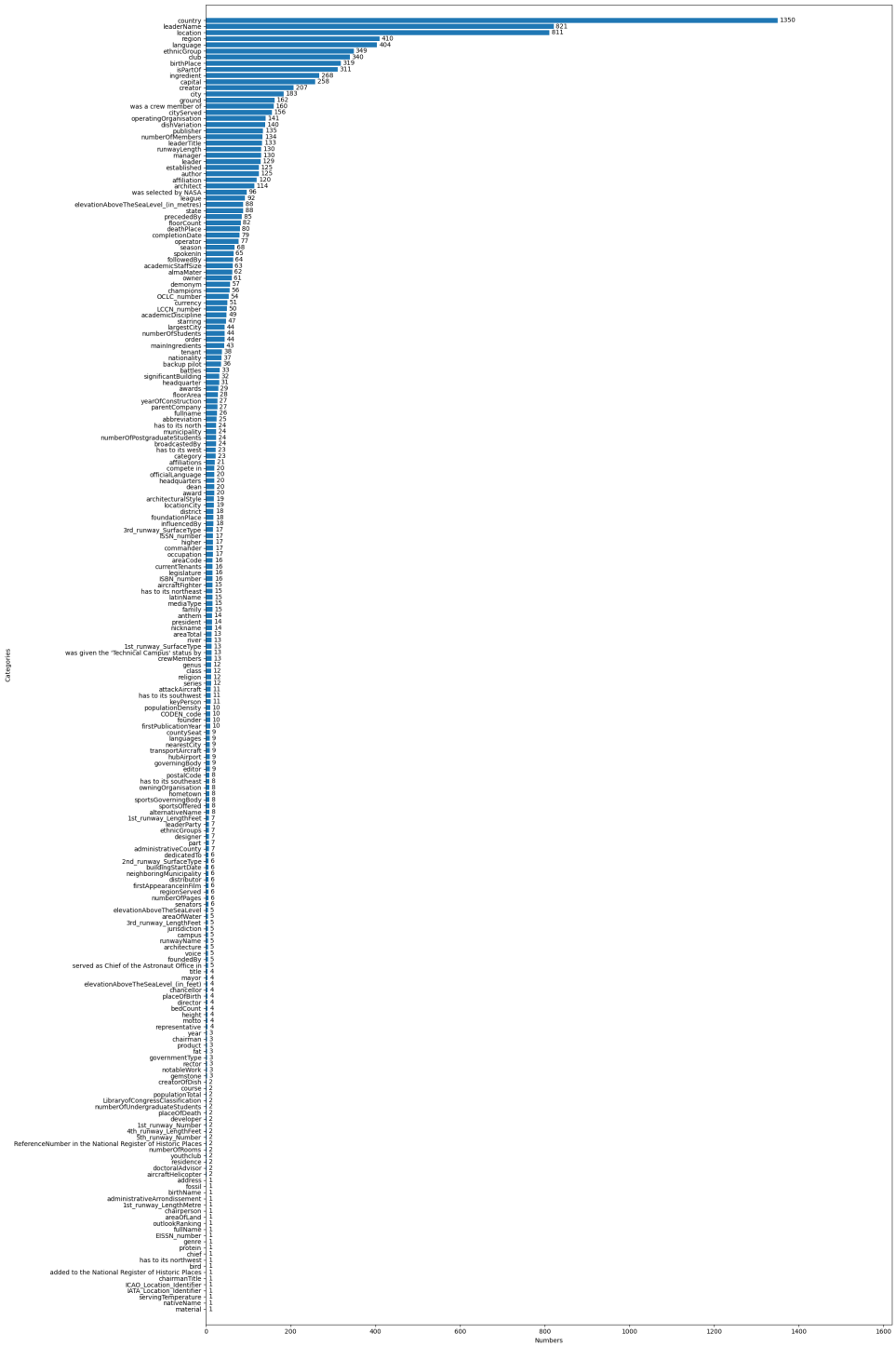

Figure 11: WebNLG exact matching train dataset relationship statistics

C Experiment Hyperparameter Configuration

Parameter	NYT*	NYT	WebNLG*	WebNLG
batch_size	8	8	8	8
epochs	80	100	340	340
num_classes	25	25	170	211
num_generated_triples	15	15	15	15
optimizer	AdamW	AdamW	AdamW	AdamW
encoder_lr	2.00E-05	1.00E-05	2.00E-05	2.00E-05
decoder_lr	8.00E-05	6.00E-05	6.00E-05	6.00E-05
mesh_lr	6	6	6	6
n_mesh_iters	4	4	4	4
num_iterations	6	3	3	3
slot_dropout	0.2	0.2	0.2	0.2
max_grad_norm	2.5	2.5	2.5	2.5
weight_decay	1.00E-05	1.00E-05	1.00E-05	1.00E-05
lr_decay	0.01	0.01	0.01	0.01
warm-up_rate	0.1	0.1	0.1	0.1

Table 7: Parameter settings for NYT and WebNLG experiments

D Licensing and Terms of Use

D.1 Dataset

We utilize the NYT dataset, which originates from the New York Times corpus and is distributed under the Linguistic Data Consortium (LDC) license. Additionally, we employ the WebNLG dataset, which is publicly available under the GNU General Public License v3.0 (GPL-3.0). Both datasets serve as standard benchmarks for evaluating relational extraction tasks and are also accessible through existing research repositories.

D.2 Model and Code Release

We will make our SMARTe implementation publicly available upon acceptance. The code will be released under the MIT License.

D.3 Software Dependencies and Implementation

The model is implemented using PyTorch[1]. We utilize the BERT-Base Cased transformer model from the Hugging Face library[2]. We use Weights & Biases (WandB)[3] for experiment logging. We adopt the relational triple extraction evaluation metrics and preprocessing steps from an existing implementation[4] to ensure consistency with prior work.

D.4 Ethical and Legal Considerations

No personally identifiable information (PII) is contained in the datasets we use. We adhere to the terms of service for all datasets and do not scrape or collect additional proprietary data.

D.5 Information About Use Of AI Assistants

We used ChatGPT[5] for minor writing refinements and code debugging but ensured all final content was reviewed and verified by the authors.

[1] https://pytorch.org/
[2] https://huggingface.co/google-bert/bert-base-cased
[3] https://wandb.ai/
[4] https://github.com/DianboWork/SPN4RE
[5] https://chatgpt.com/