SSDAU: STRUCTURED SEMANTIC DATA AUGMENTATION FOR JOINT ENTITY AND RELATION EXTRACTION

Anonymous authors

ABSTRACT

Joint Entity and Relation Extraction (JERE) is highly sensitive to training data quality, making data augmentation a natural way to improve generalization. However, existing augmentation methods often weaken entity relevance and disrupt the structured semantics of annotated triples, limiting their effectiveness for JERE. We propose Structured Semantic Data Augmentation (SSDAU), a task-specific augmentation framework that preserves annotation-compatible triple semantics under the original entity–relation schema. SSDAU segments text by entity labels, matches candidate spans using hybrid semantic similarity that combines contextualized embeddings with traditional similarity scores, and reconstructs new instances through structure-preserving replacement. To further ensure semantic faithfulness, it applies consistency-aware filtering with pair scoring and BERTopic-based topic control to remove incompatible augmentations. We evaluate SSDAU on datasets with different annotation types, comparing five representative JERE models against seven popular augmentation baselines. Results show that SSDAU consistently improves JERE performance across benchmark settings, produces structurally and semantically compatible augmented data, and is more robust to ambiguity than non-LLM methods, yielding a substantially smaller average relative F1 drop (8.93% vs. 21.94%). Our code and data are available at `https://anonymous.4open.science/r/SSDAU-B345`.

1 INTRODUCTION

JERE is a fundamental information extraction task that predicts structured relational triples from text and supports applications such as information retrieval (Zhu et al., 2025), question answering (Abdelaziz et al., 2021), and knowledge graph construction (Yang et al., 2023). Unlike sentence-level prediction (Guan et al., 2021), JERE is supervised by triples, whose correctness depends on the joint compatibility of entities, relations, and context. Its performance is therefore highly sensitive to training data quality, making data augmentation a natural way to improve generalization.

However, generic augmentation methods are often ill-suited to JERE. In particular, word substitution (Wei & Zou, 2019), paraphrasing (Tripto et al., 2024), and unconstrained generation (Lewis et al., 2020) may increase surface diversity while weakening entity relevance, distorting relation cues, or breaking the structured semantics of annotated triples. As a result, even fluent augmented sentences can harm JERE when they are incompatible with the original annotation structure. Therefore, effective augmentation for JERE requires preserving the structural validity of annotated triples rather than merely increasing lexical diversity. To this end, we propose **SSDAU**, a task-specific framework that preserves annotation-compatible triple semantics under the original entity–relation schema.

SSDAU follows a three-stage design. It first segments each labeled sentence into triple-aligned semantic units using entity labels. It then matches candidate spans with a hybrid similarity that combines contextualized embeddings and traditional similarity scores, and reconstructs new instances through structure-preserving replacement. Finally, it applies consistency-aware filtering with pair scoring and BERTopic-based topic control to remove incompatible augmentations. By preserving structural compatibility with the source annotation, SSDAU improves augmentation quality without relying solely on surface-level text perturbations.

We compare SSDAU with seven popular augmentation baselines under a fixed JERE backbone for the main study, and separately evaluate its transferability across five representative JERE models. We further examine stability under controlled out-of-domain ambiguity and add equal-budget controls,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

where all compared augmentation methods are capped to the same number of retained augmented instances after method-specific generation and filtering. Results show that SSDAU consistently improves JERE performance across benchmark settings, produces more structurally and semantically compatible augmented data, and is more robust to ambiguity than non-LLM methods, with a substantially smaller average relative F1 drop (8.93% vs. 21.94%).

Our contributions are as follows:

- We present SSDAU, a task-specific augmentation-and-filtering framework for JERE that operationalizes annotation-compatible triple preservation through triple-aligned segmentation, constrained replacement, and consistency-aware filtering.
- We provide a controlled empirical study showing that schema-consistent augmentation is often more effective than generic text perturbation for JERE, especially under partial annotation and noisy augmentation settings.
- We analyze the roles of candidate availability, filtering, and augmentation budget through ablations and additional controls, clarifying when the method is effective and what limitations remain in its ability to ensure annotation-compatible augmentation.

2 Related Work

Generic Text Augmentation. Generic augmentation methods, such as synonym replacement (Wei & Zou, 2019), back translation (Xie et al., 2020), noise injection (Ye et al., 2022), and interpolation (Chen et al., 2022), mainly improve surface diversity and local robustness. They are effective for sentence classification, but JERE is supervised by entity–relation triples rather than sentence-level labels (Wang et al., 2020). As a result, surface-level perturbations may weaken entity relevance, distort relation cues, or break annotation compatibility. This limitation motivates SSDAU, which is designed to preserve triple-level structure during augmentation.

Label-Aware and Structure-Aware Augmentation. For information extraction, prior work has explored tag-aware perturbation (Pham et al., 2025), entity substitution (Hu et al., 2023), and label-constrained editing (Liu et al., 2021). These methods are better aligned with structured tasks because they use annotation signals during augmentation. However, most of them still operate at the token or span level and do not explicitly model the joint dependency among head entities, tail entities, and relations. Consequently, locally valid edits can still produce globally incompatible JERE samples. SSDAU addresses this issue by treating the annotated triple as the basic augmentation unit.

Generative Augmentation. Recent generative (Zheng et al., 2023; Wang et al., 2024) and LLM-based (Chen et al., 2024) methods provide stronger fluency and controllability than traditional perturbation methods. However, fluency alone does not guarantee annotation compatibility in JERE, as generated sentences often alter original entity boundaries, distort relation semantics, or break contextual dependencies (Xiong et al., 2023). This problem becomes more serious when generation is only weakly constrained. In contrast, SSDAU avoids unconstrained generation and instead performs structure-preserving replacement under explicit entity–relation constraints.

Filtering and Denoising Methods. Filtering and denoising methods for relation extraction highlight the importance of sample quality, especially in noisy supervision settings (Fang et al., 2024). They remove harmful instances effectively, but filtering alone does not explain how to generate annotation-compatible samples. To address this limitation, SSDAU combines controlled augmentation with consistency-aware filtering to ensure that the generated samples maintain sufficient diversity while remaining valid for JERE training.

3 Method

In this section, we first formalize the JERE augmentation setting and then present the three components of SSDAU: (1) triple-aligned data discretization, which converts each labeled sentence into annotation-aware semantic units; (2) constrained structured augmentation, which reconstructs new instances through schema-compatible replacement; and (3) consistency-aware filtering, which removes candidates that are structurally or semantically incompatible with the source annotation. Figure 1 gives an overview of the framework, and the full procedure is summarized in Algorithm 1.

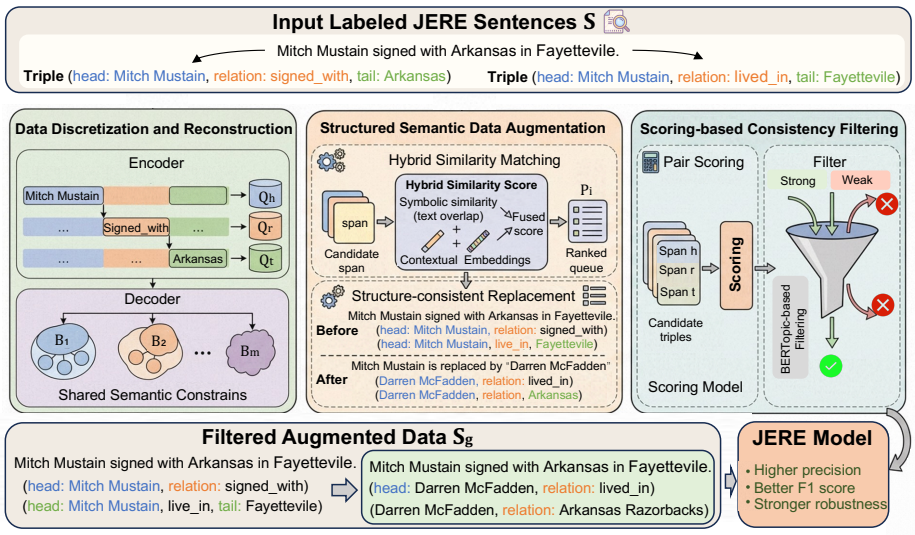


Figure 1: **Overview of SSDAU**. Given labeled JERE sentences S, SSDAU first applies ***Data Discretization and Reconstruction*** to convert each sentence into triple-aligned semantic units under shared entity–relation constraints. It then performs ***Structured Semantic Data Augmentation*** by ranking candidate spans with hybrid similarity and generating new instances through structure-preserving replacement. Finally, ***Scoring-based Consistency Filtering*** verifies whether candidate triples remain faithful to the semantic truth of the source annotation, retaining only semantically compatible augmentations. The filtered data S_g are then used to train a JERE model with higher precision, better F1, and stronger robustness.

At a high level, SSDAU is designed to generate augmented samples that remain *annotation-compatible* with the original entity–relation schema. SSDAU focuses on preserving the structural compatibility between the augmented text and the original supervision format, which is essential for JERE training, rather than verifying real-world factual correctness. The remainder of this section formalizes this objective and shows how it is implemented through triple-aligned discretization, constrained augmentation, and consistency-aware filtering.

3.1 Preliminaries

Let $S = \{s_1, s_2, \ldots, s_N\}$ be a set of labeled sentences, where each sentence contains up to n tokens, and let $R = \{r_1, r_2, \ldots, r_K\}$ denote the predefined relation set. Each sentence is associated with one or more triples $T = \{(h_i, r_i, t_i)\}_{i=1}^{M}$, where h_i and t_i are the head and tail entities, respectively. Following common JERE formulations, triple labels can be represented as a tensor in $\mathbb{R}^{n \times K \times n}$.

Because triples are the core output format of JERE, we treat each annotated triple as the basic unit of augmentation. For each triple-aligned semantic unit, we record a tuple

$$b = (q, l, \rho, p),$$

where q denotes the extracted span, l denotes the local contextual tokens surrounding the span, ρ denotes the corresponding entity or relation label, and p denotes the span boundary in the source sentence. The candidate library is denoted by

$$B = \{B_1, B_2, \ldots, B_m\},$$

where each group contains spans that satisfy the same type and relation compatibility constraints.

3.2 Triple-Aligned Data Discretization

Direct sentence-level perturbation can entangle multiple triples and obscure the dependencies that JERE models must learn. SSDAU therefore begins by converting each labeled sentence into triple-aligned semantic units.

Given a sentence s_i, the encoder identifies the head, relation, and tail spans associated with each annotated triple, records their local context and source positions, and stores the resulting units in three

Algorithm 1: Structured Semantic Data Augmentation with Consistency Filtering

Input: Labeled training set $S = \{(s_i, T_i)\}_{i=1}^{N}$; relation schema R; similarity threshold ε; consistency threshold τ_c; topic threshold τ_t; per-sentence cap $M_{\max}$.
Output: Augmented dataset S_g.

```
Initialize S_g ← ∅;
foreach (s_i, T_i) ∈ S do
    extract triple-aligned units from s_i using T_i;
    store each unit as b = (q, l, ρ, p);
    add head, relation, and tail units to type-constrained collections Q_h, Q_r, and Q_t;
group units into compatibility sets B = {B_1, ..., B_m};
partition each B_j into smaller candidate subsets for efficient matching;
foreach (s_i, T_i) ∈ S do
    retrieve compatible source units for s_i;
    sample candidate subsets from the corresponding compatibility groups;
    foreach source unit b in s_i do
        compute Θ(b, b') against compatible candidates b' in the sampled subsets;
        retain candidates with Θ(b, b') ≥ ε;
        rank retained candidates in descending order;
    generate candidate augmentations by structure-preserving replacement;
    deduplicate generated candidates by exact string match;
    keep at most M_max candidates for s_i;
    foreach candidate sentence s'_i do
        compute consistency score c(s'_i);
        compute topic similarity s_topic(s_i, s'_i);
        if c(s'_i) ≥ τ_c and s_topic(s_i, s'_i) ≥ τ_t then
            add s'_i to S_g;
return S ∪ S_g;
```

collections: Q_h, Q_r, and Q_t. This step ensures that subsequent replacement operates on structured units aligned with the original annotation rather than on unconstrained text fragments.

We then partition these units into compatibility groups according to entity type and relation schema. Each group contains only spans that can be considered as replacement candidates under the same annotation constraints. This grouping step narrows the replacement space and reduces the probability of generating label-incompatible samples.

3.3 Constrained Structured Augmentation

After discretization, SSDAU reconstructs candidate augmentations through constrained replacement.

Hybrid Similarity Matching. A key challenge is to distinguish spans that are lexically or contextually similar from spans that are truly replaceable under the source annotation schema. To address this, we compute a hybrid similarity score between two candidate units b and b':

$$\Theta(b, b') = \alpha \cdot \Theta_{\text{sym}}(b, b') + (1 - \alpha) \cdot \Theta_{\text{ctx}}(b, b'),$$

where Θ_{ctx} is a contextual similarity score computed from pretrained encoder representations, and $\alpha \in [0, 1]$ balances the two terms. We instantiate the symbolic similarity term as the average lexical-overlap score between the span text and its local context:

$$\Theta_{\text{sym}}(b, b') = \frac{1}{2}\left(\text{Jaccard}(q, q') + \text{Jaccard}(l, l')\right),$$

where q and q' denote the candidate spans, and l and l' denote their local contexts. In all main experiments, we set the hybrid-similarity weight to $\alpha = 0.5$.

For each source unit, we rank compatible candidates by $\Theta(b, b')$ and retain only those whose score exceeds a threshold ε, ensuring strictly bounded compatibility. The retained candidates are inserted into a priority queue sorted in descending order of similarity for efficient retrieval.

Structure-Preserving Replacement. For each retained candidate, we replace the corresponding head, relation, or tail span in the source sentence according to its recorded source boundary, while preserving the remaining sentence context and the original annotation layout. This step generates new training samples that strictly adhere to the original entity–relation schema instead of merely introducing surface-level lexical variations.

To prevent the augmented set from being dominated by highly similar variants, we remove exact-string duplicates after generation and retain at most $M_{\max}$ augmentations for each source sentence, where $M_{\max}$ is selected on the validation split.

3.4 Consistency-Aware Filtering

Even with constrained replacement, some generated candidates remain annotation-incompatible. SSDAU therefore includes a post-hoc filtering stage that combines pair-level consistency scoring with topic-level compatibility.

Pair-Level Consistency Scoring. We train an auxiliary relation classification module on the original training split only. This scorer is trained before augmentation filtering, then frozen, and used only to rank or reject candidate augmented samples. It does not share parameters with the downstream JERE model, and it does not use validation or test annotations for training.

Given contextual representations l_h and l_t for a head–tail pair, we define the pair representation as

$$e_{ht} = \phi\left(W[l_h; l_t]^\top + b\right),$$

where $W \in \mathbb{R}^{d_e \times 2d}$ and $b \in \mathbb{R}^{d_e}$ are trainable parameters, $[;]$ denotes concatenation, and $\phi(\cdot)$ is a ReLU activation function.

We then project the pair representation to relation scores:

$$v_{ht} = R^\top \operatorname{drop}(e_{ht}),$$

where $R \in \mathbb{R}^{d_e \times K}$ is the relation projection matrix. After a softmax layer, the scorer estimates how compatible the candidate pair is with the original relation schema.

For filtering, we compute a sentence-level consistency score $c(s')$ from the scorer outputs and retain the candidate only if it exceeds a validation-selected threshold τ_c.

Topic Compatibility. To further suppress topically incompatible replacements, we compute a topic-level similarity between the source sentence s and its augmented candidate s':

$$s_{\text{topic}}(s, s') = \cos\left(z_{\text{topic}}(s), z_{\text{topic}}(s')\right),$$

where $z_{\text{topic}}(\cdot)$ denotes the BERTopic representation of a sentence. We keep a candidate only if

$$\text{keep}(s') \iff c(s') \geq \tau_c \ \wedge \ s_{\text{topic}}(s, s') \geq \tau_t,$$

where τ_c and τ_t are tuned on the validation split.

What Filtering Guarantees. This filtering stage improves compatibility with the original annotation schema, but does *not* guarantee real-world factual correctness. Therefore, the retained samples serve as merely structurally valid augmentations for model training.

4 Experiment

4.1 Research Questions

RQ1: How does SSDAU compare with representative augmentation baselines in improving JERE performance across benchmark settings?

RQ2: How does SSDAU affect JERE performance across different annotation settings, model architectures, and controlled ambiguity conditions?

RQ3: How do the major components of SSDAU contribute to its effectiveness for annotation-compatible augmentation in JERE?

4.2 Experimental Setup

Datasets. We evaluate SSDAU on two representative English JERE benchmarks, NYT (Sandhaus, 2008) and WebNLG (Gardent et al., 2017). Each dataset has two annotation variants: a fully annotated version (NYT, WebNLG) and a partially annotated version (NYT*, WebNLG*). The superscript * denotes the partial-annotation setting, in which not all valid triples appearing in a sentence are annotated. This yields four benchmark settings for testing whether augmentation behavior changes with annotation completeness.

Baselines. We compare SSDAU with seven widely used augmentation methods: *word substitution* (WS) (Wei & Zou, 2019), *back translation* (BT) (Xie et al., 2020), *noise introduction* (NI) (Ye et al., 2022), *same-tag semantic noise* (SSN) (Yan et al., 2019), *generative models* (GM) (Hou et al., 2021), Mixup (Hu et al., 2019), and ChatIE (Wei et al., 2023). We implement all baselines under the same training and evaluation pipeline and tune their settings on the validation split when applicable. ChatIE and the additional LLM augmentation result are included as reference generation-based baselines under a unified setting rather than as claims of frontier LLM performance.

Models and Protocol. For the main comparison with augmentation baselines, we use a fixed JERE backbone to ensure fair comparison under the same model, decoding procedure, and evaluation script. Here, the decoding procedure refers to the model-specific post-processing step that converts raw predictions into final relational triples for exact-match evaluation. To test whether the benefit of SSDAU transfers across extractor designs, we further evaluate it on five representative JERE models spanning three formulations: Multi-module Multi-Step (PRGC (Zheng et al., 2021), CasRel (Wei et al., 2020)), Multi-module One-Step (TPLinker (Wang et al., 2020), SPN4RE (Sui et al., 2020)), and One-module One-Step (OneRel) (Shang et al., 2022). We report these cross-model results separately from the main baseline comparison.

Metrics. We report Precision (Prec.), Recall (Rec.), and F1 for completeness, and use F1 as the primary metric for overall comparison and model selection. Since IoU is a deterministic transformation of F1 under exact-match triple evaluation, we report it only in the appendix for completeness and do not treat it as an independent source of evidence. Unless otherwise stated, all reported numbers are averaged over five random seeds.

Implementation. We instantiate SSDAU with `bert-base-cased` as the contextual encoder. The auxiliary consistency scorer is trained on the original training split only and then frozen during augmentation filtering. We tune all augmentation-related thresholds on the validation split and fix them for test evaluation. Additional implementation details, including hardware and hyperparameters, are deferred to Appendix A.4.

Controlled ambiguity setting. To evaluate robustness under noisy augmentation conditions, we construct a controlled setting by injecting out-of-domain triples from SciER (Zhang et al., 2024) into the training set at a 10% rate, while keeping validation and test sets unchanged. We refer to this as a controlled ambiguity setting because it introduces semantically related but distributionally shifted structures during training without modifying the evaluation set.

Threshold selection. We select $\varepsilon = 0.7$ for the main experiments because it gives the best validation trade-off between augmentation coverage and annotation compatibility. Lower thresholds admit a larger but noisier candidate pool, whereas higher thresholds reduce replacement diversity and limit the benefit of augmentation. Detailed augmented sample counts under different similarity thresholds are provided in Appendix A.5, Table 7.

4.3 Main Results

4.3.1 RQ1: Comparison with Augmentation Baselines

To answer RQ1, we compare SSDAU with seven representative augmentation baselines under the same training and evaluation pipeline. Table 1 reports the results on four benchmark settings under both clean and controlled ambiguity conditions.

SSDAU achieves the best or near-best F1 on most benchmark settings and improves over the non-augmented baseline more consistently than generic augmentation methods. Although Table 1 also

Table 1: Precision, Recall, and F1 (%) for augmentation baselines under clean and controlled ambiguity settings. Values are averaged over five random seeds. The controlled ambiguity setting mixes semantically related but out-of-domain disturbance into training only. Best results in the Clean setting are highlighted in blue, and best results in the Ambiguous setting are highlighted in red.

Method	Setting	NYT*			WebNLG*			NYT			WebNLG		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Original	clean	90.17	**92.77**	91.45	90.62	89.88	90.25	**92.83**	91.52	92.17	90.66	87.55	89.08
	Ambiguous	77.02	64.46	70.20	68.22	60.73	64.25	78.24	70.35	74.07	76.32	69.77	72.90
WS	Clean	88.82	89.14	88.98	91.47	91.55	91.51	89.91	89.31	89.61	89.66	88.11	88.88
	Ambiguous	75.62	77.01	76.31	77.17	73.39	75.23	78.20	65.69	71.40	75.84	64.17	69.52
BT	Clean	88.97	90.08	89.52	91.77	92.17	91.97	89.10	89.98	89.54	89.46	**90.34**	89.90
	Ambiguous	64.44	76.19	69.82	72.47	66.82	69.53	71.17	62.11	66.33	66.62	59.23	62.71
NI	Clean	89.37	90.46	89.91	92.41	91.91	92.16	88.38	91.06	89.70	88.41	86.88	87.64
	Ambiguous	74.82	75.60	75.21	72.60	67.02	69.70	77.96	62.41	69.32	73.52	58.58	65.21
SSN	Clean	89.03	90.08	89.55	91.89	93.00	92.44	88.25	91.34	89.77	84.77	87.12	85.93
	Ambiguous	73.92	74.70	74.31	70.88	67.86	69.34	76.12	60.16	67.21	72.40	58.48	64.70
GM	Clean	88.30	90.49	89.38	91.84	92.99	92.41	88.60	90.11	89.35	90.82	87.54	89.15
	Ambiguous	72.12	74.96	73.51	76.54	71.28	73.82	75.32	59.06	66.21	72.68	59.00	65.13
Mixup	Clean	90.56	89.57	90.06	91.29	**93.17**	92.22	91.36	88.99	90.16	90.35	86.72	88.50
	Ambiguous	76.82	73.67	75.21	77.30	71.34	74.20	77.64	64.76	70.62	76.42	65.10	70.31
ChatIE	Clean	62.92	61.76	62.33	67.75	56.33	61.51	62.40	62.80	62.60	66.70	63.36	64.99
	Ambiguous	59.36	52.88	55.93	63.48	61.49	62.47	64.42	**76.19**	69.81	69.20	56.61	62.28
SSDAU	Clean	**92.00**	92.10	**92.05**	**92.80**	93.10	**92.95**	91.74	**94.09**	**92.90**	**91.58**	88.36	**89.94**
	Ambiguous	**83.22**	**78.35**	**80.71**	**88.14**	**91.57**	**89.82**	**80.10**	75.82	**77.90**	**84.56**	**88.39**	**86.43**

Table 2: The precision of different models under different datasets. Each cell (A/B) represents the performance of training with the original dataset (A) and the data augmented by SSDAU (B). Values in bold indicate the improvement.

Model	NYT*	WebNLG*	NYT	WebNLG
SPN (Sui et al., 2020)	91.44/**91.95**	93.81/**96.84**	92.67/92.64	90.21/**90.88**
PRGC (Zheng et al., 2021)	93.33/**93.36**	94.00/**94.46**	93.54/**94.40**	89.92/**91.32**
CasRel (Wei et al., 2020)	88.97/**91.47**	91.77/**92.13**	89.10/**91.74**	89.46/**91.58**
OneRel (Shang et al., 2022)	90.17/**92.00**	90.62/**92.80**	92.83/**92.90**	90.66/**91.60**
TPLinker (Wang et al., 2020)	90.23/**92.21**	90.89/**91.34**	91.33/**92.27**	89.12/**89.93**

reports precision and recall, our overall conclusions are based on F1, which balances extraction exactness and coverage under triple-level evaluation. This result suggests that JERE benefits more from augmentation that preserves annotation-compatible triple structure than from augmentation that only increases surface diversity.

Compared with WS, BT, NI, and SSN, SSDAU better preserves the dependency between entities and relations, which reduces the risk of generating annotation-incompatible samples. Compared with GM, Mixup, and ChatIE, SSDAU provides stronger structural control by restricting replacement to triple-aligned units and validating the results with consistency-aware filtering. These advantages translate into higher and more consistent F1 scores across benchmark settings (Table 1), confirming that maintaining schema compatibility contributes more to JERE performance than simply increasing textual diversity. Overall, the results provide a positive answer to RQ1.

4.3.2 RQ2: Generalization Across Settings and Robustness

To answer RQ2, we focus on whether the benefit of SSDAU transfers across annotation settings, model architectures, and the controlled ambiguity probe, rather than re-establishing the baseline comparison in Section 4.3.1.

Compared with the main baseline results, the additional evidence in Table 2 shows that the gain of SSDAU is not tied to a single extractor design: improvements remain visible across multi-step and one-step JERE architectures. We also report additional controls in Table 3, where "Budget" equalizes augmentation size across methods and "Low" evaluates the 10% training-data setting.

Table 3: Additional evaluation results in average F1 (%) across the four benchmark settings. "Main" reports the average over five random seeds, "Budget" uses equal augmentation size, and "Low" uses 10% of the training data.

Method	Main	Budget	Low
Original	90.67	–	75.83
Mixup	90.21	90.04	77.18
GM	90.07	89.95	77.17
LLM Aug.	87.89	87.42	74.63
SSDAU	**91.95**	**91.55**	**79.75**

Under the controlled ambiguity probe, SSDAU shows a substantially smaller relative F1 drop than the compared non-LLM baselines (8.93% vs. 21.94% on average across WS, BT, NI, SSN, GM, and Mixup). Note that the consistency-aware filter in SSDAU applies only to the newly generated augmented candidates; it does not remove the injected out-of-domain triples already present in the original training set. We attribute the observed stability to the increased proportion of schema-compatible triples in the augmented training set. By adding these high-quality samples, SSDAU effectively dilutes the influence of the injected out-of-domain triples, thereby reducing their relative contribution to the gradient updates during training. In contrast, generic augmentation baselines (e.g., WS, NI, SSN) may further amplify surface-level noise without correcting for schema compatibility, compounding rather than diluting the injected disturbance. We emphasize that this probe is intended to test robustness to structured perturbation introduced through the training distribution, rather than to faithfully simulate in-schema label noise or to demonstrate an explicit denoising capability.

4.3.3 RQ3: Component Contribution Analysis

To answer RQ3, we conduct ablation studies on the major components of SSDAU, including triple-aligned preprocessing, structure-preserving replacement, consistency-aware filtering, and scorer initialization. Table 4 summarizes the main ablation results, and Table 5 reports the effect of parameter initialization.

Table 4: Ablation study for SSDAU. "No Split" denotes not splitting the text. "No Label Split" denotes splitting by semantics without semantic tag. "Full Split" denotes complete splitting of the words in the text.

Setting	NYT*	WebNLG*	Avg.
Original	90.17	90.62	90.39
SSDAU	92.00	92.80	92.40
Preprocessing			
No Split	89.32	90.17	89.75
No Label Split	90.33	90.42	90.38
Full Split	88.64	89.76	89.20
Replacement			
(h, t)	64.21	73.83	69.02
(r)	77.42	84.31	80.87
(h, r, t)	90.41	91.13	90.77
Filtering			
No Filtering	89.92	90.84	90.38

Preprocessing. Removing triple-aligned segmentation consistently reduces performance. Both *No Split* and *Full Split* underperform SSDAU, showing that augmentation quality depends on preserving span boundaries together with entity and relation labels. This result indicates that the gain of SSDAU does not come from arbitrary fragmentation, but from annotation-aware discretization.

Replacement. Replacing triple components in a coordinated way is more effective than partial replacement. In particular, replacing only head–tail pairs or only relations leads to much weaker results than replacing (h, r, t) together. This suggests that JERE augmentation benefits from preserving the internal dependency among entities and relations rather than perturbing them independently.

Filtering. Removing consistency-aware filtering reduces performance to near or below the non-augmented baseline. This shows that structure-preserving replacement alone is insufficient to prevent noisy augmentations, and that filtering is necessary to select annotation-compatible samples.

The results in Tables 4 and 5 indicate that the performance improvements of SSDAU rely on the joint application of annotation-aware segmentation, coordinated triple replacement, and consistency-aware filtering. Additionally, using pretrained initialization for the scorer further enhances the filtering quality. All ablation scores are measured after full training on the union dataset returned by Algorithm 1; because filtering, deduplication, and the per-sentence cap jointly affect the realized

Table 5: Precision, Recall, and F1 (%) under different parameter initialization strategies across partially and fully annotated datasets.

Init Type	Partially Annotated						Fully Annotated					
	NYT*			WebNLG*			NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Pretrained	92.00	92.10	92.05	92.80	93.10	92.95	91.74	94.09	92.90	91.58	88.36	89.94
Random	73.92	40.46	52.30	76.54	57.58	65.72	67.35	37.43	48.12	90.26	47.74	62.45
Zero	78.42	64.58	70.83	83.78	40.90	54.97	78.22	64.05	70.43	87.65	36.74	51.78

augmentation budget, performance is not expected to vary linearly with the raw candidate counts reported in the appendix.

4.4 Analysis

We further analyze why SSDAU helps and where it remains limited.

Robustness under controlled ambiguity. SSDAU is more stable under the controlled ambiguity probe primarily because its own generated augmentations are constrained by type and schema compatibility and are further filtered for consistency and topic relevance. Consequently, the augmented samples maintain schema compatibility, which increases the proportion of clean supervision signals within the overall training distribution. We note that this mechanism is best understood as *indirect dilution of pre-existing noise* rather than *active denoising*: SSDAU does not identify or remove the injected out-of-domain triples already present in the original training data. This distinction is important for correctly interpreting the robustness gains reported in Table 1, and we discuss it further as a limitation in Section 4.4. Additional evidence from threshold sensitivity, semantic coherence, and augmented-data distribution is provided in Appendix A.5 (Table 7), Appendix A.6, and Appendix A.7.

Why threshold and initialization matter. The similarity threshold controls a coverage–compatibility trade-off: lower values increase candidate volume but admit more noise, while higher values improve compatibility but reduce diversity. We therefore select $\epsilon = 0.7$ based on validation performance rather than candidate count (Appendix A.5, Table 7). The consistency scorer is also crucial, and pretrained initialization substantially outperforms random and zero initialization, indicating that effective filtering depends on contextual semantic representations.

Current limitations. SSDAU targets annotation-compatible augmentation rather than factual verification, so retained samples should not be interpreted as guaranteed facts (Appendix A.6; Appendix A.10). Its usefulness also depends on candidate availability in the source corpus: when compatible candidates are sparse, the benefit of augmentation becomes limited (Appendix A.5, Table 7; Appendix A.7). In addition, the filtering stage relies on a pretrained auxiliary scorer (Table 4). Finally, our experiments are limited to English benchmarks (Appendix A.1, Table 6), and the controlled ambiguity setting should be viewed as a targeted stability probe rather than a full simulation of open-domain distribution shift.

5 Conclusion

We introduced SSDAU, a task-specific augmentation framework for JERE that preserves compatibility with the original entity–relation annotation schema through triple-aligned discretization, constrained replacement, and consistency-aware filtering. Across NYT and WebNLG, SSDAU yields consistent improvements under different annotation settings, model architectures, and controlled ambiguity conditions, with additional controls indicating that the gains are associated with augmentation quality rather than volume alone. These results demonstrate that preserving annotation-compatible structures is an effective strategy for JERE data augmentation, and this approach may also benefit other structured prediction tasks that rely on complex relational labels. We hope this study encourages broader investigation of augmentation methods that preserve not only surface fluency, but also the mechanistically grounded structural validity required by downstream task supervision paradigms.

AI USE STATEMENT

In this work, we used generative AI tools only for limited manuscript-preparation tasks, including improving readability, polishing English expression, and suggesting minor revisions to wording and organization. We did not use generative AI tools to generate synthetic datasets, clean or reformat datasets, develop the core conceptual framework of the method, formulate mathematical claims, provide ingredients for proofs, write proofs, design the main methodology or experiments, implement the final method used in our evaluation, or interpret the final results and conclusions. These tasks were either completed entirely by the authors or are not applicable to this work. All AI-assisted outputs were carefully reviewed, revised, and verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Ibrahim Abdelaziz, Srinivas Ravishankar, Pavan Kapanipathi, Salim Roukos, and Alexander Gray. A semantic parsing and reasoning-based approach to knowledge base question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 15985–15987, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17988.

Hui Chen, Wei Han, Diyi Yang, and Soujanya Poria. Doublemix: Simple interpolation-based data augmentation for text classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 4622–4632, 2022.

Zhuowei Chen, Lianxi Wang, Yuben Wu, Xinfeng Liao, Yujia Tian, and Junyang Zhong. An effective deployment of diffusion lm for data augmentation in low-resource sentiment classification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1844–1856, 2024.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. Advaug: Robust adversarial augmentation for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5961–5970, 2020. URL https://arxiv.org/abs/2006.11834.

Tianqing Fang, Wenxuan Zhou, Fangyu Liu, Hongming Zhang, Yangqiu Song, and Muhao Chen. On-the-fly denoising for data augmentation in natural language understanding. In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 766–781, 2024.

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

Jian Guan, Xiaoxi Mao, Changjie Fan, Zitao Liu, Wenbiao Ding, and Minlie Huang. Long text generation by modeling sentence-level and discourse-level coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6379–6393, 2021.

Yutai Hou, Sanyuan Chen, Wanxiang Che, Cheng Chen, and Ting Liu. C2c-genda: Cluster-to-cluster generation for data augmentation of slot filling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 13027–13035, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17540.

Xuming Hu, Yong Jiang, Aiwei Liu, Zhongqiang Huang, Pengjun Xie, Fei Huang, Lijie Wen, and Philip S Yu. Entity-to-text based data augmentation for various named entity recognition tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 9072–9087, 2023.

Zhiting Hu, Bowen Tan, Ruslan Salakhutdinov, Tom Mitchell, and Eric P Xing. Learning data manipulation for augmentation and weighting. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 15764–15775, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/671f0311e2754fcdd37f70a8550379bc-Abstract.html.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4163–4174, Online, 11 2020. Association for Computational Lingustics. doi: 10.18653/v1/2020.findings-emnlp.372. URL https://aclanthology.org/2020.findings-emnlp.372.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Linlin Liu, Bosheng Ding, Lidong Bing, Shafiq Joty, Luo Si, and Chunyan Miao. Mulda: A multilingual data augmentation framework for low-resource cross-lingual ner. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5834–5846, 2021.

Jonas Mueller and Aditya Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 2786–2792, 2016. URL https://ojs.aaai.org/index.php/AAAI/article/view/10350.

Nhu Vo Quynh Pham, Uyen Bao Nguyen Phuc, Long Hong Buu Nguyen, and Dinh Dien. Enhancing named entity translation from classical chinese to vietnamese in traditional vietnamese medicine domain: A hybrid masking and dictionary-augmented approach. In *Proceedings of the 18th International Natural Language Generation Conference*, pp. 408–418, 2025.

Adriana Picoral, Shelley Staples, and Randi Reppen. Automated annotation of learner english: An evaluation of software tools. *International Journal of Learner Corpus Research*, 7(1):17–52, 2021.

Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pp. 47–57, 2022.

Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6 (12):e26752, 2008.

Yu-Ming Shang, Heyan Huang, and Xianling Mao. Onerel: Joint entity and relation extraction with one module in one step. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11285–11293, 2022. URL https://ojs.aaai.org/index.php/AAAI/article/view/21379.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, Xiangrong Zeng, and Shengping Liu. Joint entity and relation extraction with set prediction networks. *arXiv preprint arXiv:2011.01675*, 2020. URL https://arxiv.org/abs/2011.01675.

Nafis Irtiza Tripto, Saranya Venkatraman, Dominik Macko, Robert Moro, Ivan Srba, Adaku Uchendu, Thai Le, and Dongwon Lee. A ship of theseus: curious cases of paraphrasing in llm-generated texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6608–6625, 2024.

Tianshi Wang, Qikai Yang, Ruijie Wang, Dachun Sun, Jinyang Li, Yizhuo Chen, Yigong Hu, Chaoqi Yang, Tomoyoshi Kimura, Denizhan Kara, et al. Fine-grained control of generative data augmentation in iot sensing. *Advances in neural information processing systems*, 37:32787–32812, 2024.

William Yang Wang and Diyi Yang. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 2557–2563, 2015. URL https://openreview.net/forum?id=r1NRyfzuWB.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. Tplinker: Single-stage joint extraction of entities and relations through token pair linking. In *Proceedings of the 28th International Conference on Computational Linguistics)*, pp. 1572–1582, Barcelona, Spain (Online), December 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.coling-main.138. URL https://aclanthology.org/2020.coling-main.138.

Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 6382–6388, 2019.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. Chatie: Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*, 2023.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1476–1488, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.136. URL https://aclanthology.org/2020.acl-main.136.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020.

Xiong Xiong, Chen Wang, Yunfei Liu, and Shengyang Li. Enhancing ontology knowledge for domain-specific joint entity and relation extraction. In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics*, pp. 713–725, 2023.

Ge Yan, Yu Li, Shu Zhang, and Zhenyu Chen. Data augmentation for deep learning of judgment documents. In *International Conference on Intelligent Science and Big Data Engineering*, pp. 232–242. Springer, 2019. URL https://link.springer.com/chapter/10.1007/978-3-030-36204-1_19.

Zhe Yang, Yi Huang, and Junlan Feng. Learning to leverage high-order medical knowledge graph for joint entity and relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 9023–9035, 2023.

Fanghua Ye, Yue Feng, and Emine Yilmaz. Assist: Towards label noise-robust dialogue state tracking. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2719–2731, 2022.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. URL https://openreview.net/forum?id=r1Ddp1-Rb&;noteId=r1Ddp1-Rb),.

Qi Zhang, Zhijia Chen, Huitong Pan, Cornelia Caragea, Longin Jan Latecki, and Eduard Dragut. Scier: An entity and relation extraction dataset for datasets, methods, and tasks in scientific documents. *arXiv preprint arXiv:2410.21155*, 2024.

Chenyu Zheng, Guoqiang Wu, and Chongxuan Li. Toward understanding generative data augmentation. *Advances in neural information processing systems*, 36:54046–54060, 2023.

Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming, and Yefeng Zheng. Prgc: Potential relation and global correspondence based joint relational triple extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6225–6235, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.486. URL https://aclanthology.org/2021.acl-long.486.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. Large language models for information retrieval: A survey. *ACM Transactions on Information Systems*, 44(1):1–54, 2025.

A Appendix

A.1 Dataset Statistics

Table 6 summarizes the statistics of the four benchmark settings used in this paper. NYT and WebNLG are each evaluated in a fully annotated and a partially annotated variant, which allows us to test whether the effect of augmentation depends on annotation completeness.

Table 6: Dataset statistics. N denotes the number of triples in a sentence.

Dataset	Train	Valid	Test	Relations	$N=1$	$N=2$	$N=3$	$N=4$	$N\geq 5$
NYT	56195	5000	5000	24	36835	12065	3672	2623	1001
NYT*	56195	4999	5000	24	36868	12058	3663	2618	988
WebNLG	5019	500	703	216	1865	1237	1033	641	243
WebNLG*	5019	500	703	171	1716	1264	1043	648	348

A.2 Baseline Details

We compare SSDAU with representative augmentation baselines that introduce different types of variation. Some primarily increase lexical diversity, while others rely on paraphrasing, noise injection, interpolation, or generation. Unlike SSDAU, these methods do not explicitly optimize for annotation-compatible triple preservation in JERE.

Word Substitution. Word substitution methods replace words with near-synonyms using lexical resources, embeddings, or mask-based models (Mueller & Thyagarajan, 2016; Wang & Yang, 2015; Jiao et al., 2020). They can preserve sentence-level meaning to some extent, but do not explicitly guarantee compatibility with entity–relation annotations.

Back Translation. Back translation generates paraphrases by translating the source text into another language and then translating it back (Xie et al., 2020). This often preserves coarse sentence meaning, but does not explicitly control whether the original entity–relation semantics remain usable for JERE.

Noise Introduction. Noise-based augmentation perturbs text by adding, deleting, or modifying words or phrases (Ye et al., 2022). Although this increases textual diversity, such perturbations can easily damage the structural compatibility required by triple-based supervision.

Same-Tag Semantic Noise. SSN introduces tag-constrained perturbations at the word or sentence level (Yan et al., 2019; Sahu et al., 2022). Relative to unconstrained noise, it better preserves local labels, but it still does not check whether the full triple structure remains valid.

Generative Models. Generative augmentation methods synthesize new training sentences from the original corpus (Hou et al., 2021). They can produce fluent outputs, but the generated text is not inherently constrained to preserve the entity–relation structure required by JERE.

Mixup. Mixup-based augmentation interpolates examples or labels in representation space (Zhang et al., 2017; Cheng et al., 2020). While effective for regularization, it does not explicitly preserve the structured semantics of annotated triples.

A.3 JERE Model Details

We evaluate SSDAU on five representative JERE models to test whether its benefit transfers across extractor designs rather than depending on a single architecture.

SPN. SPN4RE (Sui et al., 2020) is a set-prediction framework that generates relational triples in parallel using bipartite matching.

PRGC. PRGC (Zheng et al., 2021) decomposes JERE into relation prediction, global correspondence, and entity extraction, which improves efficiency and handles overlapping triples.

CasRel. CasRel (Wei et al., 2020) models relations as mappings from subjects to objects and is particularly effective for overlapping relational triples.

Table 7: Augmented sample counts under different similarity thresholds.

Dataset	ε	Head	Relation	Tail	Sum.
NYT^*	$0.5 \sim 0.6$	15,062	243	11,300	26,605
	$0.6 \sim 0.7$	9,439	38	4,631	14,108
	$0.7 \sim 0.8$	1,825	19	1,365	3,209
	$0.8 \sim 0.9$	2,927	0	1,137	4,064
	$0.9 \sim 1.0$	960	0	1,546	2,506
$WebNLG^*$	$0.5 \sim 0.6$	7,082	2,742	8,116	17,940
	$0.6 \sim 0.7$	3,933	1,946	5,342	11,221
	$0.7 \sim 0.8$	2,049	2,162	1,557	5,768
	$0.8 \sim 0.9$	814	2,005	1,021	3,840
	$0.9 \sim 1.0$	5,463	890	2,929	9,282
NYT	$0.5 \sim 0.6$	13,507	234	10,076	23,817
	$0.6 \sim 0.7$	7,721	36	4,063	11,820
	$0.7 \sim 0.8$	4,922	13	1,588	6,523
	$0.8 \sim 0.9$	2,198	0	1,140	3,338
	$0.9 \sim 1.0$	3,700	0	1,051	4,751
$WebNLG$	$0.5 \sim 0.6$	4,023	3,186	6,028	13,237
	$0.6 \sim 0.7$	2,673	2,009	4,445	9,127
	$0.7 \sim 0.8$	968	1,345	1,123	3,436
	$0.8 \sim 0.9$	309	919	923	2,151
	$0.9 \sim 1.0$	3,019	444	6,935	10,398

OneRel. OneRel (Shang et al., 2022) formulates JERE as a single-stage classification problem with unified token interaction modeling.

TPLinker. TPLinker (Wang et al., 2020) uses a handshaking tagging scheme to jointly model entities and relations while reducing exposure bias.

A.4 Implementation Details

All experiments are conducted on a server with an Intel Xeon Gold 6248 2.50GHz CPU, two Tesla V100 SXM2 32GB GPUs, and Ubuntu 18.04.6. We use the pretrained `bert-base-cased` model from HuggingFace as the contextual encoder.

To reduce the cost of large-scale matching, we partition the candidate pool into smaller subsets and perform matching within sampled subsets rather than over the full pool. In the main experiments, we use uniform random subset sampling with a fixed seed and keep the subset setting unchanged across all compared methods. We further apply exact-string deduplication after generation to reduce near-duplicate augmentations.

The consistency scorer is trained only on the original training split and then frozen before augmentation filtering. All thresholds are selected on the validation split and fixed for test evaluation. We use `bert-base-cased` as the contextual encoder, so the encoder hidden size is $d = 768$. In the main experiments, we set $\alpha = 0.5$, $d_e = 256$, $\tau_c = 0.7$, $\tau_t = 0.8$, and $M_{\max} = 3$. The scorer is trained for 5 epochs with AdamW, learning rate 2×10^{-5}, and batch size 16. For BERTopic, we use the default sentence-embedding setting with fixed clustering parameters across datasets.

A.5 Threshold Sensitivity

Table 7 reports the number of augmented samples under different similarity thresholds. We select $\varepsilon = 0.7$ for the main experiments because it provides the best validation trade-off between augmentation coverage and annotation compatibility.

A.6 Supplementary Quality Analysis

Semantic coherence. To further examine whether augmented text remains compatible with the source annotation beyond aggregate metrics, we perform a semantic coherence analysis. We first preserve the semantic tag pattern of the source triple, and then use Biber Tagger (Picoral et al., 2021) to compare the syntactic patterns of source and augmented texts. Table 8 shows two examples with different normalized syntactic coherence scores ν, where larger values indicate better agreement with the source pattern.

Table 8: Supplementary semantic-coherence verification for annotation-compatible augmentation. We compare augmented texts that preserve the same triple schema but differ in their agreement with the syntactic pattern of the source sentence. $\nu \in [0, 1]$ denotes normalized syntactic coherence, where larger values indicate stronger structural consistency with the source text.

Source	Text = South Africa, and the rest of Africa. Triple = [[Africa, /location/location/contains, South Africa]] Schema = location contain location
Examples	Text_1 = South Africa is a part of Africa. $\nu = 0.516$ Text_2 = North Africa, and the rest of Africa. $\nu = 0.923$ Triple = [[Africa, /location/location/contains, North Africa]] Schema = location contain location

IoU-based overlap quality. Table 9 reports IoU for the compared augmentation methods, recomputed from Table 1. Together with the main-paper precision, recall, and F1 results, this provides an additional view of prediction overlap quality under both clean and controlled ambiguity settings.

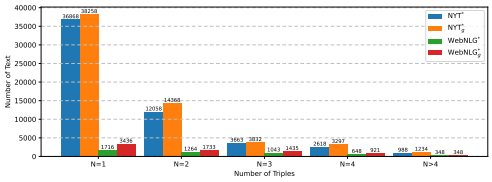
Table 9: IoU (%) of SSDAU versus augmentation baselines.

Method	Setting	NYT*	WebNLG*	NYT	WebNLG	Avg
Original	Clean	84.25	82.24	85.47	80.34	83.08
Original	Ambiguous	54.06	47.34	58.84	57.35	54.40
WS	Clean	80.13	84.30	81.17	80.00	81.40
WS	Ambiguous	61.70	60.29	55.50	53.30	57.70
BT	Clean	81.01	85.11	81.05	81.69	82.21
BT	Ambiguous	53.63	53.28	49.63	45.67	50.55
NI	Clean	81.67	85.47	81.32	77.92	81.60
NI	Ambiguous	60.28	53.50	53.08	48.36	53.80
SSN	Clean	81.08	85.93	81.47	75.35	80.96
SSN	Ambiguous	59.13	53.08	50.59	47.81	52.65
GM	Clean	80.79	85.87	80.72	80.40	81.94
GM	Ambiguous	58.27	58.49	49.42	48.27	53.61
Mixup	Clean	81.86	85.55	82.05	79.23	82.17
Mixup	Ambiguous	60.28	59.00	54.56	54.20	57.01
ChatIE	Clean	45.24	44.42	45.58	48.13	45.84
ChatIE	Ambiguous	32.49	45.42	53.64	45.21	44.19
SSDAU	Clean	85.30	86.77	86.73	81.74	85.14
SSDAU	Ambiguous	67.66	81.53	63.80	76.08	72.27

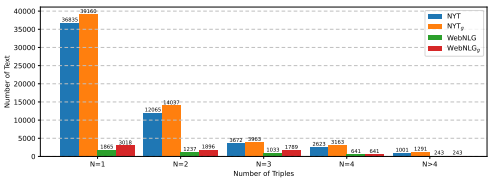
A.7 Distribution of Augmented Samples

Figure 2 compares the original and augmented training sets across triplet-count groups. SSDAU supplements most groups and is especially helpful in sparse regions, while some high triplet-count groups show little or no increase because structurally compatible candidates are scarce. This suggests that the method enriches informative parts of the training distribution where augmentation is feasible, rather than uniformly increasing data volume across all groups.

Figure 2: Comparison of the original and augmented training data across triplet-count groups. SSDAU supplements most groups in both partially annotated and fully annotated settings, with larger gains in relatively sparse regions. Some high triplet-count groups in the fully annotated WebNLG setting show little or no increase, reflecting limited candidate availability for structurally constrained augmentation rather than uniform gains across all groups.



(a) Partially annotated datasets



(b) Fully annotated datasets

A.8 Additional Model-Type Results

Table 10 groups the model-level results by JERE architecture type. This provides a supplementary view of the model-agnostic utility of SSDAU.

Table 10: Supplementary comparison across JERE model types. Each cell reports Original/SSDAU.

Module Type	NYT*	WebNLG*	NYT	WebNLG
Multi-module Multi-Step	91.48 / 92.42	92.89 / 93.33	91.32 / 93.07	89.69 / 91.45
Multi-module One-Step	90.81 / 91.98	92.22 / 94.82	92.75 / 92.77	90.44 / 91.24
One-module One-Step	90.17 / 92.00	90.62 / 92.80	92.83 / 92.90	90.66 / 91.60

A.9 LLM Augmentation Prompt

For completeness and reproducibility, we report the prompt template used for the "LLM Aug." reference baseline reported in the main paper. This baseline is included as a unified generation-based reference rather than as a claim of frontier LLM performance.

When a dataset instance contains multiple triples, all triples are provided in the same prompt so that the model can preserve the full annotation context rather than only a single local relation. We use the same prompt style for all datasets to keep the comparison consistent.

Because LLM-based generation may still produce outputs that are fluent but not fully compatible with the original annotation structure, we treat "LLM Aug." as a reference generation baseline rather than as a structure-preserving method. This is also why the main paper compares it separately from SSDAU, whose design explicitly optimizes annotation-compatible triple preservation.

Further details of the baseline setting are described in Appendix A.2, and supplementary qualitative comparisons are provided in Appendix A.10.

Prompt template. For completeness and reproducibility, we report the XML-style prompt template used for the "LLM Aug." reference baseline. The structured format separates instructions, constraints, input fields, and output specification, which helps preserve prompt clarity for sentence-level paraphrase generation.

```
<instruction>
  You are given a sentence annotated for joint entity and relation extraction (JERE).
  Generate exactly one paraphrased sentence that remains usable under the original
    annotation.
</instruction>

<objective>
  Preserve the original annotation-compatible entity--relation structure while
    paraphrasing the sentence.
</objective>

<rules>
  <rule>Preserve the original meaning as much as possible.</rule>
  <rule>Preserve all annotated triples.</rule>
  <rule>Keep all annotated entities explicitly mentioned.</rule>
  <rule>Do not delete, rename, split, merge, or replace any annotated entity.</rule>
  <rule>Do not introduce any new entity, relation, event, or fact.</rule>
  <rule>Do not change the number of triples or which entities participate in each
    relation.</rule>
  <rule>You may rewrite surrounding context, wording, or word order only if all triples
    remain clearly recoverable.</rule>
  <rule>Keep the output natural, grammatical, and concise.</rule>
  <rule>Output exactly one rewritten sentence and nothing else.</rule>
</rules>

<input>
  <sentence>{sentence}</sentence>
  <triples>{triples}</triples>
</input>
```

```
<output_format>
  <sentence>one rewritten sentence only</sentence>
</output_format>
```

A.10 Supplementary Case Study

We provide two supplementary case-study views. First, Table 11 shows representative SSDAU augmentations. Second, Table 12 illustrates how conventional augmentation methods modify the same source example.

Representative SSDAU cases. The examples in Table 11 show how SSDAU augments triple-aligned units under explicit entity and relation constraints. They demonstrate annotation compatibility rather than guarantee factual correctness.

Table 11: Augmented data generated by SSDAU. Black texts are the original examples. Red texts are the discrete text. Blue texts are the preconditions for text segmentation and augmentation. ε denotes the similarity threshold used for candidate matching.

Source	Text: At Arkansas , the freshman Mitch Mustain led the Razorbacks in a 24-23 double-overtime upset of Alabama. Triples: Mitch Mustain(people)|Arkansas(place)|place_lived Razorbacks(group)|Mitch Mustain(people)|contain
Head → Head	Condition: $Tag_h = people, Tag_t = place, Tag_r = place_lived, \Theta_h \geq \varepsilon$. Text: At Arkansas, the freshman Amy Grant led the Razorbacks in a 24-23 double-overtime upset of Alabama. Triples: Amy Grant(people)|Arkansas(place)|place_lived Razorbacks(group)|Amy Grant(people)|contain
Tail → Tail	Condition: $Tag_h = people, Tag_t = place, Tag_r = place_lived, \Theta_t \geq \varepsilon$. Text: At Nashville, the freshman Mitch Mustain led the Razorbacks in a 24-23 double-overtime upset of Alabama. Triples: Mitch Mustain(people)|Nashville(place)|place_lived Razorbacks(group)|Mitch Mustain(people)|contain
Relation → Relation	Condition: $Tag_h = people, Tag_t = place, \Theta_r \geq \varepsilon$. Text: At Arkansas, the freshman Mitch Mustain led the Razorbacks in a 24-23 double-overtime upset of Alabama. Triples: Mitch Mustain(people)|Arkansas(place)|location Razorbacks(group)|Mitch Mustain(people)|contain

Baseline comparison cases. Table 12 shows how baselines modify the same source example. These examples highlight the contrast between surface variation and triple-aware augmentation.

Table 12: Augmented cases by conventional data augmentation methods.

Source	Text: Goodyear officials scurried back to their racing headquarters in Akron, Ohio, and carefully reviewed the tire data from the June 12 race. Triples: Ohio(location)|Akron(location)|contains
WS	Text: Goodyear officials scurried back to their racing headquarters in Akron, Ohio, and carefully took a closer review to the tire data from the June 12 race. Triples: Ohio(location)|Akron(location)|contains
BT	Text: Goodyear officials rushed back to their headquarters in Akron, Ohio, and carefully examined the tire data from the June 12 race. Triples: Ohio(location)|Akron(location)|contains
NI	Text: Goodyear officials scurried rushed back to their racing headquarters in Akron, Ohio, and carefully reviewed the tire data from the June 12 race competition. Triples: Ohio(location)|Akron(location)|contains
SSN	Text: Goodyear officials scurried anxiously back to their racing headquarters in Akron, Ohio, and carefully reviewed the tire data from the June 12. Triples: Ohio(location)|Akron(location)|contains
GM	Text: Officials from Goodyear hurried back to the racing headquarters in Akron, Ohio, to double-check the tire data for the June 12 race. Triples: Ohio(location)|Akron(location)|contains
Mixup	Text: Goodyear officials scurried back to their racing headquarters in Akron, Ohio, city, and carefully reviewed the tissue data from the June 12 race. Triples: Ohio(location)|Akron(location)|contains