SSDAU: Structured Semantic Data Augmentation for Joint Entity and Relation Extraction

Anonymous ACL submission

Abstract

Joint Entity and Relation Extraction (JERE) is highly susceptible to weak generalization due to low-quality training data. Data augmentation is a common strategy to enhance model generalization across different domains. However, existing data augmentation methods often overlook text relevance and may disrupt semantic structures and dependencies, making it difficult to generate effective augmented data for improving model generalization. In this paper, we propose **Structured Semantic Data Augmentation (SSDAU)**, a novel method designed to preserve the semantic structure of text during augmentation. SSDAU segments text based on entity labels and employs an encoder to capture semantic features of entities through context awareness. It then performs entity semantic restructuring to generate augmented data. To mitigate potential topic ambiguity and information loss, we apply the BERTTopic model to filter out irrelevant topics, ensuring topic consistency. We evaluate SSDAU on datasets with different annotation types and compare its performance on five representative JERE models against six popular data augmentation baselines. Extensive experiments demonstrate that SSDAU generates data with a consistent semantic structure, leading to improved JERE model performance and surpassing state-of-the-art baselines.

1 Introduction

Joint Entity and Relation Extraction (JERE) is widely used for representation learning on text data due to its strong performance in applications such as information retrieval (Lin et al., 2020), question answering (Abdelaziz et al., 2021), and text summarization (Zhong et al., 2020). The generalization performance of JERE models heavily depends on the quality and scale of the training data. A common strategy to enhance generalization is data augmentation. Techniques such as MixUp (Cheng et al., 2020) and back-translation (Xie et al., 2020) enable efficient expansion of the training set by generating new data with subtle perturbations derived from the original samples.

However, a key challenge in applying existing techniques to enhance the generalization of JERE models is that introducing noise or perturbations into the original data may weaken entity relevance (Kambhatla et al., 2022). Training on incorrectly generated data can ultimately degrade JERE models' performance. Additionally, entities are often involved in multiple triples with complex semantic relations and dependencies. Existing data augmentation methods can disrupt the structures and dependencies, leading to issues such as overlapping relations and cascading (Liu et al., 2020).

To address this issue, we propose Structured Semantic Data Augmentation (SSDAU) to preserve the semantic structure of text during data augmentation. Instead of directly perturbing text, SSDAU aligns triplet text to maintain semantic integrity. First, we use a feature-based encoder to segment the text, ensuring that each segment retains the semantics of its neighboring regions. Next, we match segments with similar semantic labels using a decoder. To maintain structural consistency, we implement a text matcher based on semantic similarity. Finally, we substitute text with high similarity to reorganize the original text, generating augmented data while preserving semantic coherence. Inspired by recent techniques such as BertTopic (Grootendorst, 2022), we further refine the augmented text by filtering out irrelevant topics using topic modeling based on BERT and c-TF-IDF.

To assess the effectiveness of SSDAU, we compared its performance on four widely used datasets with six baseline methods. We evaluated the performance of different data augmentation techniques on different JERE tasks and models, including Multi-Module Multi-Step (Zheng et al., 2021; Wei et al., 2020), Multi-Module One-Step (Sui et al.,

2020; Wang et al., 2020), and One-Module One-Step (Shang et al., 2022). The experiment results demonstrate that SSDAU outperformed state-of-the-art methods in improving the generalization of JERE models.

2 Related Work

Information Extraction JERE is an NLP task that maps entities and relations to generate a text-to-triplet model, assigning the triple to a new annotation (Fu et al., 2019). Previous JERE models often use joint modeling (Ren et al., 2017) or sequential annotation (Zheng et al., 2017) to extract entities and relations together, focusing on structured learning through manually constructed features or knowledge tables (Miwa and Bansal, 2016). However, these manual features limit their performance across applications. To address this, Zhao et al. (Zhao et al., 2021) decompose the JERE task and modify the classification process for contextual learning. They categorize JERE models into three types: multi-module multi-step (Zheng et al., 2021; Wei et al., 2020), multi-module one-step (Sui et al., 2020; Wang et al., 2020), and one-module one-step (Shang et al., 2022). The accuracy of these models is constrained by the quality of training data, and our structured semantic data augmentation method can generate high-quality data for both basic and downstream JERE applications.

Semantic Match Semantic matching is a subtask of text matching used to retrieve semantically similar texts in search scenarios (Wu et al., 2022). Common approaches include cosine similarity, TF-IDF, and DSSM (Gao et al., 2021). Recent studies show that pre-training semantic classification models can compress large amounts of text and improve the generalization of semantic matching models (Brown et al., 2020). For example, the *Similarities* tool (Zhang Bingyu, 2022) enhances practical applications for text semantic matching, especially in text relation extraction. Based on existing techniques, we improve JERE by incorporating text semantic matching.

Data Augmentation Data augmentation is an effective and efficient method to improve machine learning model performance, especially in data-limited environments (Cashman et al., 2020). Common techniques in NLP include word replacement (Wei and Zou, 2019), word vector replacement (Wang and Yang, 2015), masked language model replacement (Jiao et al., 2020), back translation (Zhang et al., 2020), and adding noise (Min et al., 2020; Yan et al., 2019; Hou et al., 2018). Zhang et al. (Zhang et al., 2015) and Jonas et al. (Mueller and Thyagarajan, 2016) proposed lexical substitution to preserve semantics, but this method is limited by the size of the proxemics list. Unlike simple perturbation (Liu et al., 2020) or extra augmentor models (Hou et al., 2021; Hu et al., 2019), we propose sampling-based augmentation to generate data with the same semantic structure while maintaining the logic of the samples.

3 Method

In this section, we first define the problems. Then, we introduce the three main components of SSDAU: 1) data discretization and reconstruction, 2) structured semantic data augmentation, and 3) scoring-based consistency filtering. Figure 1 depicts the overall framework of SSDAU.

3.1 Preliminaries

Given set of sentences $S = \{s_1, s_2, ..., s_N\}$ containing L token and K predefined relations $R = \{r_1, r_2, ..., r_K\}$, we extract entities and relations to construct triples $T = \{(h_i, r_i, t_i)\}_{i=1}^{M}$ in S, where h_i, t_i are the head and tail entities, respectively, N represents the number of sentences, M represents the number of triples. We store the knowledge as a three-dimensional matrix M^{L*K*L}.

Since triplets are the core output format of JERE, we use the triplet as the basic unit of data augmentation and partition the text according to the triplet to obtain three series of text collections. To preserve the contextual semantics of the segmented text, we keep the contextual token l of each segmented text and record the location of each cut point p.

3.2 Data Discretization and Reconstruction

Encoder We use the triplet as the basic unit of data augmentation to eliminate the noise from textual perturbations. We design a text feature-based encoder E (the structure is shown in Figure 2). The input of the encoder is the sentence text S. For each sentence s_i, we locate the specified text block $(q_{h_i}, q_{r_i}, q_{t_i})$ based on the triplet tags $(\rho_{h_i}, \rho_{r_i}, \rho_{t_i})$, and record the context token $(l_{h_i}, l_{r_i}, l_{t_i})$ and its cut position $(p_{h_i}, p_{r_i}, p_{t_i})$. The encoder processes all the input text and gets three output text collections according to the tag types: head entity collection Q_h, tail entity collection Q_t, and relation entity collection Q_r.

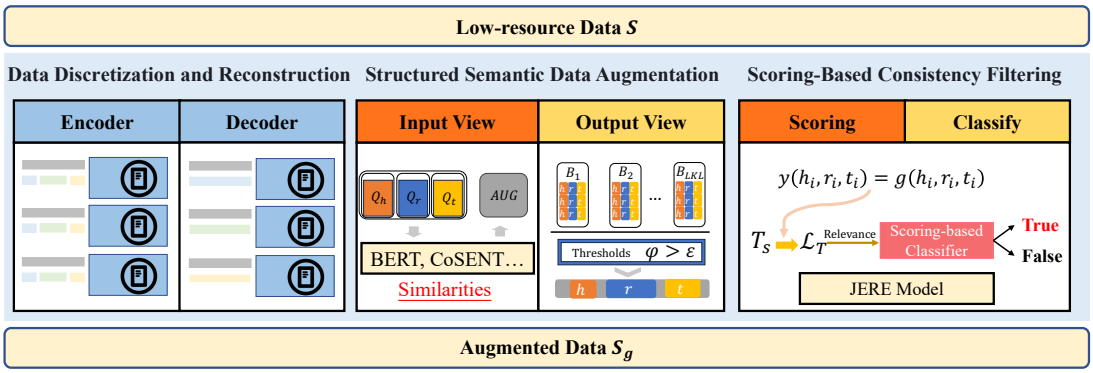


Figure 1: **Overview of SSDAU**. The ***Data Discretization and Reconstruction*** component discretizes the text data S semantically using the *Encoder* and outputs text collections in the form of segmented sets. The *Decoder* then processes these segmented sets to facilitate the ***Structured Semantic Data Augmentation*** component, where the *Input View* is based on similarity matching, while the *Output View* focuses on augmenting the data. Finally, the ***Scoring-based Consistency Filtering*** component uses a structured semantic classifier to filter low-resource data, and the remaining augmented data $\mathcal{L}$ and T are used as augmented data S_g to train a more robust JERE model.

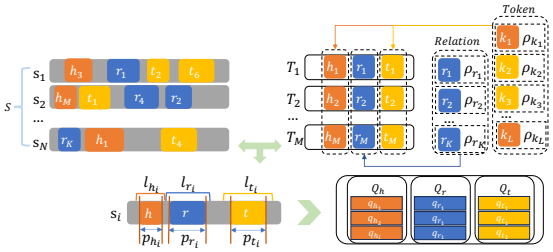


Figure 2: The structure of our feature-based encoder.

Decoder We then design a similarity-based text matching decoder D. The input of decoder D is (Q_h, Q_t, Q_s). The decoder divides the text collections according to the relation types and label types to get $L * K * L$ groups $B = \{B_1, B_2, ..., B_{LKL}\}$, where each group has the same relation type and the same label.

3.3 Structured Semantic Data Augmentation

Discrete Text Matching We designed a text matcher based on the semantic similarity evaluation tool *Similarities* (Zhang Bingyu, 2022) to align the decoder's output. A text block b in an output group $B_i = b_1, b_2, \ldots, b_j$ from the decoder stores the text q, context tokens l, label type ρ, and segmentation position p. We perform matching across all b in different text corpora B_i, incorporating semantic, syntactic, and lexical similarity evaluations, as well as context token similarity assessments. The matching results are normalized to a value between 0 and 1 and inserted into a priority queue sorted in descending order of similarity. Finally, for each B_i, we obtain a similarity-based priority queue P_i.

Data Augmentation After completing the similarity matching, we filter out data in the priority queue $P_i = P_1, P_2, ..., P_{KM}$ with a similarity score lower than the threshold ε. For the remaining data, we replace the text content of the corresponding text blocks based on the recorded segmentation position l in each block's information, thereby generating the augmented data.

3.4 Scoring-based Consistency Filtering

To further improve the quality of the augmented data, we employ a BERTTopic model to identify and retain key terms from topic descriptions. We then filter out augmented data associated with irrelevant topics, ensuring the topic coherence of the generated text.

First, we extract all entities and relations from the text. Then, we encode the tokens using BERT (Kenton and Toutanova, 2019), obtaining the corresponding entity tokens $l_1, l_2, \ldots, l_L$. Next, we combine entities and relations in the form of (l_h, r, l_t) and perform triplet extraction using joint entity and relation extraction (Shang et al., 2022). Finally, we apply a function to compute the correlation between the head and tail entities. The scoring function is defined as:

$$h \star t = \phi(W[l_h; l_t]^T + b) \quad (1)$$

where h and t represent the head and tail, respectively. $\star$ denotes circular correlation ($\mathbb{R} \times \mathbb{R}^d \rightarrow \mathbb{R}^d$). $W \in \mathbb{R}^{d_e \times 2d}$ and b are trainable weights and biases, respectively, where d_e denotes the dimension of the entity. $[;]$ is the concatenation operation and $\phi(\cdot)$ represents the ReLU activation function.

We then incorporate the highly evaluated entity pairs with the relations and use the relational representation function $R \in \mathbb{R}^{d_e \times 4K}$. The vector function is defined as follows:

$$\upsilon_{(l_h,r_k,l_t)}{}_{k=1}^{K} = R^T \phi(drop(W[l_h; l_t]^T + b)) \quad (2)$$

where υ represents the score vector and $drop(\cdot)$ refers to the dropout strategy (Srivastava et al., 2014).

Next, we add the scoring vector υ to the softmax function to predict the corresponding labels. The formulated triples are presented as follows:

$$\zeta_{triple} = -\frac{\sum_{i,j,k} logP(y_{(l_i,r_k,l_j)}, g_{(l_i,r_k,l_j)}|S)}{L \times K \times L} \quad (3)$$

where $g_{(l_i,r_k,l_j)}$ represents the gold tag obtained from annotations. We match all triplets with the golden-label triplets to compute the topic score for each triplet. Finally, we select the high-scoring triplets as the topic relationships for the text. Augmented data in which these topic relationships have been replaced is filtered out, ensuring that the final augmented data remains both topic-relevant and structurally coherent.

4 Experiment

4.1 Experimental Setup

Baseline We compare SSDAU with six commonly used data augmentation methods, including *word substitution* (WS) (Wei and Zou, 2019), *back translation* (BT) (Xie et al., 2020), *noise introduction* (NI) (Fanghua Ye, 2022), *same-tag semantic noise* (SSN) (Yan et al., 2019), *generative models* (GM) (Hou et al., 2021), and Mixup (Hu et al., 2019).

Protocol We select five models for three different types of JERE tasks: 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 use the following metrics to measure the effectiveness, performance, and adaptability of SSDAU: precision (Prec), F1-score (F1), and Intersection over Union (IoU).

Implementation We conducted all experiments on a single server equipped with an Intel Xeon Gold 6248 2.50GHz CPU, two Tesla V100 SXM2 32GB GPUs, and Ubuntu 18.04.6 operating system.

Table 1: The number of augmented samples produced by SSDAU at various thresholds on different datasets.

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

We reused the pre-trained BERT model (base-cased English) from Huggingface [1].

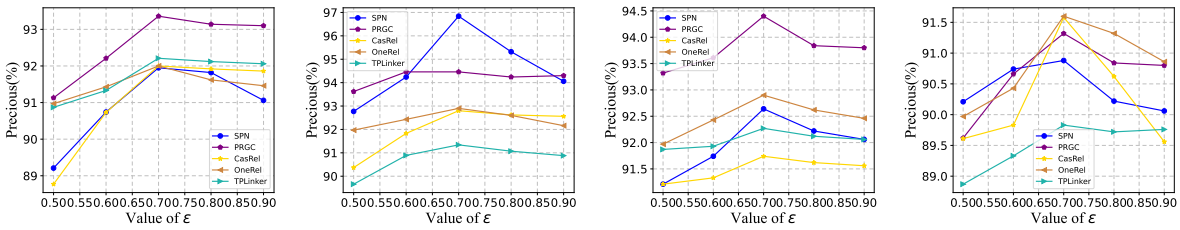
Dataset We conduct our experiments on two representative English datasets, NYT (Sandhaus, 2008) and WebNLG (Gardent et al., 2017). Both types of datasets have two variations: fully annotated type (NYT, WebNLG) and partially annotated type (NYT*, WebNLG*).

Evaluation and Selection of Thresholds Table 1 describes the number of augmented samples generated by SSDAU for different sets of semantic domains under various similarity thresholds. For the four datasets, we count the number of augmented data under different variable settings for different entities and relations. The results indicate that the number of augmented samples decreases as the threshold value increases. Figure 3 shows the precision of the four augmented datasets under different JERE models with various similarity thresholds. The results suggest that the datasets augmented by SSDAU perform the best under different JERE tasks at a threshold value of 0.7.

4.2 Results

Comparison with Baselines Table 2 presents the effectiveness (Prec), performance (F1), and adaptability (IoU) results of SSDAU and six baselines for different JERE tasks. The results demonstrate

[1] https://huggingface.co/google-bert/bert-base-cased



(a) NYT* (b) WebNLG* (c) NYT (d) WebNLG

Figure 3: Extraction precision of the JERE models with different similarity thresholds. (a), (b), (c), and (d) describe the precision of different JERE models under different datasets, respectively.

Table 2: Precision (%) , F1 score (%) and Intersection over Union (%) of our proposed SSDAU and baselines in CasRel model. All results are the the average over multiple patterns and 3 iterations.

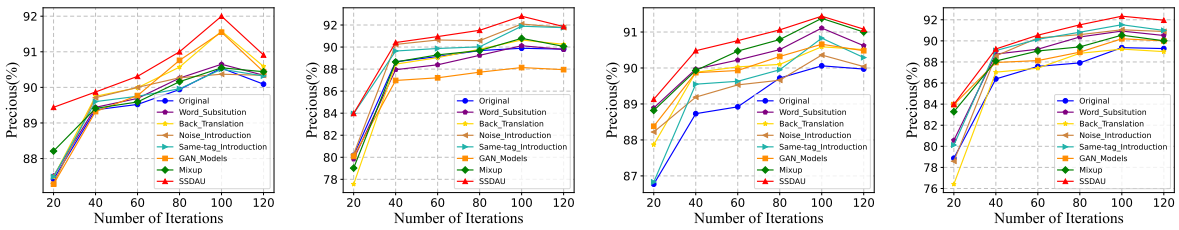
Category	Partial Match						Exact Match					
	NYT*			WebNLG*			NYT			WebNLG		
	Prec.	F1	IoU	Prec.	F1	IoU	Prec.	F1	IoU	Prec.	F1	IoU
Original	90.17	91.45	84.24	90.62	90.25	82.23	92.83	92.17	85.47	90.66	89.08	80.31
WS (Wei and Zou, 2019)	88.82	88.98	80.16	91.47	91.51	84.35	89.91	89.61	81.17	89.66	88.88	79.98
BT (Prabhumoye et al., 2018)	88.97	89.52	81.02	91.77	91.97	85.14	89.10	89.54	81.07	89.46	89.90	81.70
NI (Fanghua Ye, 2022)	89.37	89.91	81.67	92.41	92.16	85.46	88.38	89.70	81.32	88.41	87.64	78.00
SSN (Yan et al., 2019)	89.03	89.55	81.08	91.89	92.44	85.94	88.25	89.77	81.44	84.77	85.93	75.34
GM (Hou et al., 2021)	88.30	89.38	80.79	91.84	92.41	85.89	88.60	89.35	80.75	90.82	89.15	80.42
Mixup (Hu et al., 2019)	90.56	90.06	81.92	91.29	92.22	85.56	91.36	90.16	82.08	90.35	88.50	79.37
SSDAU	**92.00**	**92.05**	**85.27**	**92.80**	**92.95**	**86.83**	**91.74**	**92.90**	**86.74**	**91.58**	**89.94**	**81.77**

that SSDAU consistently outperforms the baseline in terms of the effectiveness of data augmentation for various JERE tasks. In terms of performance, SSDAU achieves the best F1 scores and generates positive outcomes, unlike the six baselines that negatively impact JERE models. Regarding adaptability, the results of IoU for augmented data indicate that our method performs better across different JERE models. These findings highlight the excellent adaptability of our approach in low-resource JERE tasks.

In comparison to Back Translation (Xie et al., 2020) and Generative Models (Hou et al., 2021), maintaining the semantic structure of the text proves to be more effective than preserving semantic continuity. Contrasted with Noise Introduction (Fanghua Ye, 2022) and Same-tag Semantic Noise (Yan et al., 2019), the method that maps discrete text by tags exhibits superior performance to adding noise directly. In contrast to Word Substitution (Wei and Zou, 2019) and Mixup (Cheng et al., 2020), labeled discrete texts demonstrate superior properties in JERE data augmentation tasks compared to unlabeled samples. Based on these results, we conclude that the method of data augmentation by preserving the structured semantics of the text is superior to existing data augmentation strategies.

Figure 4 displays the training results of SSDAU with six baselines under the CasRel model at various iterations throughout the training process. The results reveal that SSDAU consistently achieves optimal performance across different iteration numbers. These findings suggest that the augmented data produced by SSDAU is beneficial for enhancing JERE models. Additionally, SSDAU consistently delivers promising enhancements on four datasets in contrast to traditional methods.

Performance on Different JERE Tasks Table 3 displays the effectiveness of SSDAU and baselines for various JERE models. The results indicate that the SSDAU-augmented dataset exhibits improvements across different types of JERE models, such as 3.03% improvement on precision for the $WebNLG_g^*$ dataset in SPN and a 0.94% improvement for the NYT_g dataset in the TPLinker model. These outcomes demonstrate the feasibility of our approach for augmenting unstructured texts into structured semantic data for JERE tasks. Moreover, we observe that SSDAU performs better on partially annotated type datasets than on fully annotated type datasets. Notably, our method achieves about a 3% improvement with the NYT* of the



(a) NYT* (b) WebNLG* (c) NYT (d) WebNLG

Figure 4: The effects of various baselines on different datasets are examined. Specifically, (a), (b), (c), and (d) illustrate the precision of the data augmentation baseline at different iterations, with the CasRel model serving as the prediction model.

Table 3: 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**

CasRel model and the WebNLG* dataset of the SPN model.

4.3 Ablation Study

We conduct an ablation study on the NYT^* and $WebNLG^*$ benchmarks to evaluate three components: data discretization and reconstruction, structured semantic data augmentation, and scoring-based consistency filtering. Throughout this process, we maintain consistent settings across all components.

Data Discretization and Reconstruction First, we remove the pre-processing component, ***Data Discretization and Reconstruction***, and instead directly split the data based on the triad message, without semantic tags (*No Label Split*). Additionally, we apply conventional text splitting methods, including the *no-split* and complete *full-split* schemes (Gao et al., 2020).

As shown in Table 4, we evaluate the effectiveness of the pre-processing components both before and after removal using precision as a metric. Our results demonstrate that the ***Data Discretization and Reconstruction*** component outperforms the no-pre-processing approach, with an improvement of approximately 2.02%-3.20%. Furthermore, we find that incorporating semantic tagging prompts positively impacts discrete text data augmentation in low-resource JERE tasks.

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.

Dataset	NYT*	WebNLG*	Avg.
CasRel Baseline	90.17	90.62	90.39
SSDAU	92.00	92.80	92.40
Ablation for Pre-processing			
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
Ablation for Augmentation			
(h,t)	64.21	73.83	69.02
(r)	77.42	84.31	80.87
(h,r,t)	90.41	91.13	90.77
(h,r,h)	85.66	88.53	87.10
(t,r,t)	82.12	84.44	83.28
Ablation for Filtering			
No Filtering	89.92	90.84	90.38

Structured Semantic Data Augmentation We evaluate the effectiveness of the augmentation component by using the exact match method to measure the similarity between pre-processed discrete texts and generate augmented data accordingly. In this process, the labels of the composed discrete texts are substituted with the labels of the augmented data. The augmented data is then classified based on the type of triplet, used to train the model, and its utility is assessed after the removal of the augmentation component.

Table 5: Semantic consistency verification of augmented text. ν is the syntactic coherence.

Source	Text = South Africa, and the rest of Africa. Triple = [[Africa, /location/location/contains, South Africa]] Structured Semantic = location contain location
Syntax Matching	Text1 = South Africa is a part of Africa. $\nu = 0.516$ Text2 = North Africa, and the rest of Africa. $\nu = 0.923$ Triple = [[Africa, /location/location/contains, North Africa]] Structured Semantic = location contain location

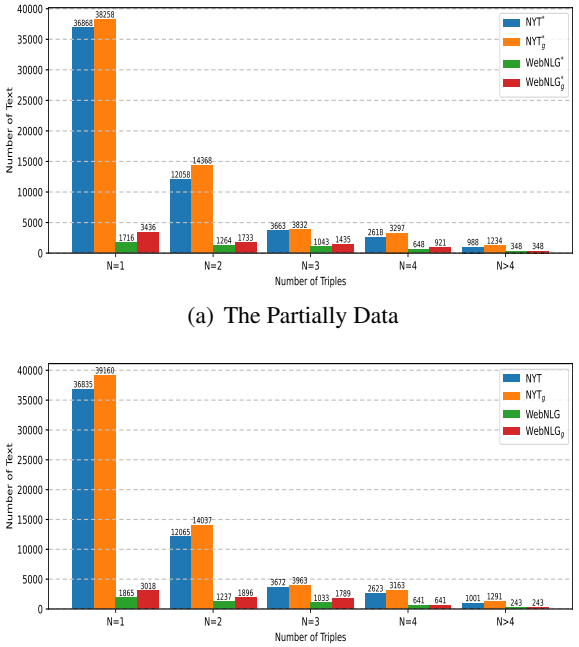
As shown in Table 4, the augmented data consists of five types of triplet tags. Among these, only the augmented texts in the third group (h, r, t) exhibit a modest positive effect (0.38%) on JERE tasks. In contrast, the other four types negatively impact the precision of JERE tasks. When the augmentation component is removed, the threshold restriction is lifted, allowing low-quality data to be included in the augmentation process. This results in a significant increase in negative data, thereby reducing the model's precision.

Furthermore, the augmentation component helps preserve the semantic structure and facilitates the mapping between augmented texts and triplet labels. Without this component, the text extraction process is disrupted, leading to performance degradation in JERE tasks. The findings highlight a substantial decrease in the precision of JERE models upon removing the augmentation component, underscoring the critical role of semantically structured data augmentation.

Scoring-based Consistency Filtering We assess the impact of the consistency filtering component in SSDAU. Table 4 shows the precision of the JERE models with and without filtered data. The results demonstrate that the filtered data positively impacts the model's precision, whereas dthe precision decreases when low-quality augmented data are not removed. This highlights the importance of consistency filtering in maintaining the model's precision.

4.4 Analysis

Semantic coherence analysis. During the semantic coherence analysis of SSDAU, we follow a two-step process to ensure semantic consistency in the augmented text. First, we augment all texts by considering similarities between annotations of the same type and entity text, while preserving the semantic annotations (e.g., "location contains location"). Next, we use Biber Tagger (A. Bergman, 2022) to match triplet texts with identical tags. The high degree of syntactic agreement between $Text1$ and $Text2$ is demonstrated in Table 5. We filter out texts with low relevance (below 0.8) and incorporate the remaining data into the training set as augmented data, ensuring the semantic consistency of the augmented text.



(a) The Partially Data

(b) The Exactly Data

Figure 5: The comparison between the number of text after augmentation with SSDAU and the initial one for different types of datasets.

In Section 3.3, we explain that during the similarity matching process, we distinguish between entities and relations in the triplets, performing separate similarity matching for entity texts and replacing all triplets containing the modified texts. This approach effectively addresses the issue of mutually exclusive relations caused by textual correlation, ensuring the semantic consistency of the augmented text with the original.

Training Cost and Convergence. Figure 5 provides details about the original and augmented texts containing varying numbers of triplets. We focus specifically on scenarios where an entity appears

Table 6: Augmented data generated by SSDAU. Black texts are the original examples. Red texts are the discrete text. Blue texts are the precondition for text segmentation and augmentation. ε_1 is the entity similarity threshold and ε_2 is the relation similarity threshold.

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: $Tag_h = people, Tag_t = place, Tag_r = place\_lived, \Theta_h \geq \varepsilon_1$. 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: $Tag_h = people, Tag_t = place, Tag_r = place\_lived, \Theta_t \geq \varepsilon_1$. 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: $Tag_h = people, Tag_t = place, \Theta_r \geq \varepsilon_2$. 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

in multiple triplet relations and categorize the texts based on the number of triplets to evaluate the effectiveness of SSDAU for such texts. By classifying the augmented data according to triplet counts and incorporating it into the training set, we assess the performance of different JERE models using the same test set. The results demonstrate the effectiveness of SSDAU for texts with different triplet counts. Our method proves valuable across texts with varying numbers of triplets, showing that as the number of triplets in the training set decreases, the availability of augmented data increases, leading to improved model precision.

4.5 Case Study

Table 6 presents three cases of SSDAU applied to JERE tasks. In the first case, we replace the head entity "$Mitch\ Mustain$" with "$Amy\ Grant$" while preserving the semantic label and other text intact. In the second case, we substitute the tail entity "$Arkansas$" with "$Nashville$" while maintaining the original semantic labels and other texts. In the third case, we modify all the text except for the entity and change the semantic label from "$people|people|place_lived$" to "$people|people|location$." Our data augmentation approach can expand texts without introducing additional noise, resulting in natural and diverse augmentations. Compared to existing methods, SSDAU's augmented data resolves diversity and quality issues more effectively.

5 Conclusion

We propose SSDAU, a data augmentation paradigm designed to perform instance augmentation for low-resource JERE tasks by labeling the semantic segmentation of entity texts and assessing similarity within neighboring semantic regions. Compared to traditional methods, SSDAU effectively addresses the challenge of data scarcity in low-resource scenarios and mitigates issues such as reduced textual relevance and overlapping relations. These findings suggest that preserving the semantic structure of texts through structured semantic tags can be a promising approach for text data augmentation.

Limitations

Although the proposed SSDAU outperforms all baseline methods, it still has some limitations. Firstly, while we alleviate the need for high-quality data in SSDAU by filtering low-quality data, incorporating more high-quality data may further improve SSDAU's performance. Secondly, we improve *Similarities* for structured semantic matching of long texts through pre-processing The efficiency of our approach can be enhanced by utilizing a more efficient semantic text-matching component. In future work, it would be interesting to validate our approach in real-time using newly acquired high-quality data and explore the development of semantic text matching components that deliver superior results for long texts.

References

Mona Diab A. Bergman. 2022. Towards responsible natural language annotation for the varieties of arabic. In *Findings of the Association for Computational Linguistics: ACL 2022*, page 364–371, Dublin, Ireland. Association for Computational Linguistics.

Ibrahim Abdelaziz, Srinivas Ravishankar, Pavan Kapanipathi, Salim Roukos, and Alexander Gray. 2021. A semantic parsing and reasoning-based approach to knowledge base question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 15985–15987.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Dylan Cashman, Shenyu Xu, Subhajit Das, Florian Heimerl, Cong Liu, Shah Rukh Humayoun, Michael Gleicher, Alex Endert, and Remco Chang. 2020. Cava: A visual analytics system for exploratory columnar data augmentation using knowledge graphs. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1731–1741.

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

Emine Yilmaz Fanghua Ye, Yue Feng. 2022. Assist: Towards label noise-robust dialogue state tracking. In *Findings of the Association for Computational Linguistics: ACL 2022*, page 2719–2731, Dublin, Ireland. 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, Florence, Italy. Association for Computational Linguistics.

Pan Gao, Qi Wan, and Linlin Shen. 2020. Split and merge: Component based segmentation network for text detection. In *International Conference on Pattern Recognition and Artificial Intelligence*, pages 14–27. Springer.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

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).

Issam Laradji Gaurav Sahu, Pau Rodriguez. 2022. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 47–57, Dublin, Ireland. Association for Computational Linguistics.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Yutai Hou, Sanyuan Chen, Wanxiang Che, Cheng Chen, and Ting Liu. 2021. C2c-genda: Cluster-to-cluster generation for data augmentation of slot filling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13027–13035.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1234–1245.

Zhiting Hu, Bowen Tan, Ruslan Salakhutdinov, Tom Mitchell, and Eric P Xing. 2019. Learning data manipulation for augmentation and weighting. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 15764–15775.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, page 4163–4174, Online. Association for Computational Lingustics.

Nishant Kambhatla, Logan Born, and Anoop Sarkar. 2022. Cipherdaug: Ciphertext based data augmentation for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 201–218, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 7999–8009, Online. Association for Computational Linguistics.

Sisi Liu, Kyungmi Lee, and Ickjai Lee. 2020. Document-level multi-topic sentiment classification of email data with bilstm and data augmentation. *Knowledge-Based Systems*, 197:105918.

Junghyun Min, R Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of*

the Association for Computational Linguistics, page 2339–2352, Online. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Seoul, South Korea.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2786–2792.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 866–876, Melbourne, Australia. Association for Computational Linguistics.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1015–1024.

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

Yu-Ming Shang, Heyan Huang, and Xianling Mao. 2022. Onerel: Joint entity and relation extraction with one module in one step. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11285–11293.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

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

William Yang Wang and Diyi Yang. 2015. 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*, pages 2557–2563.

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)*, page 1572–1582, Barcelona, Spain (Online). Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. 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)*, page 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 1476–1488, Online. Association for Computational Linguistics.

Xinyi Wu, Zhenyao Wu, Yuhang Lu, Lili Ju, and Song Wang. 2022. Style mixing and patchwise prototypical matching for one-shot unsupervised domain adaptive semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2740–2749.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2020. Unsupervised data augmentation for consistency training. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 6256–6268.

Ge Yan, Yu Li, Shu Zhang, and Zhenyu Chen. 2019. Data augmentation for deep learning of judgment documents. In *International Conference on Intelligent Science and Big Data Engineering*, pages 232–242. Springer.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Xiang Zhang, Junbo Zhao, and Yann Lecun. 2015. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 2015:649–657.

Yi Zhang, Tao Ge, and Xu Sun. 2020. Parallel data augmentation for formality style transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 3221–3228, Online. Association for Computational Linguistics.

Nikolay Arefyev Zhang Bingyu. 2022. The document vectors using cosine similarity revisited. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, page 129–133, Dublin, Ireland. Association for Computational Linguistics.

Tianyang Zhao, Zhao Yan, Yunbo Cao, and Zhoujun Li. 2021. A unified multi-task learning framework for joint extraction of entities and relations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14524–14531.

Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming, and Yefeng Zheng. 2021. 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)*, page 6225–6235, Online. Association for Computational Linguistics.

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, Vancouver, Canada. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuan-Jing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 6197–6208, Online. Association for Computational Linguistics.