BiSPN: Generating Entity Set and Relation Set Coherently in One Pass

Yuxin He\textsuperscript{1} and Buzhou Tang\textsuperscript{1,2,}\textsuperscript{*}
\textsuperscript{1}Department of Computer Science, Harbin Institute of Technology, Shenzhen, China
\textsuperscript{2}Peng Cheng Laboratory, Shenzhen, China
215651847@stu.hit.edu.cn
tangbuzhou@gmail.com

Abstract

By modeling the interaction among instances and avoiding error propagation, Set Prediction Networks (SPNs) achieve state-of-the-art performance on the tasks of named entity recognition and relation triple extraction respectively. However, how to jointly extract entities and relation triples via SPNs remains an unexplored problem, where the main challenge is the maintenance of coherence between the predicted entity/relation sets during one-pass generation. In this work, we present Bipartite Set Prediction Network (BiSPN), a novel joint entity-relation extraction model that can efficiently generate entity set and relation set in parallel. To overcome the challenge of coherence, BiSPN is equipped with a novel bipartite consistency loss as well as an entity-relation linking loss during training. Experiments on three biomedical/clinical datasets and a general-domain dataset show that BiSPN achieves new state of the art in knowledge-intensive scene and performs competitively in general-domain, while being more efficient than two-stage joint extraction methods.

1 Introduction

Extracting entities and relation triples from text is a fundamental task of Information Extraction. There have been many efforts that decompose the problem into separate tasks, i.e. named entity recognition (NER) and relation triple extraction (RE), and solve them respectively. Among these efforts, Set Prediction Networks (SPNs) have demonstrated state-of-the-art performance on NER (Tan et al., 2021; Shen et al., 2022) and RE (Sui et al., 2020; Tan et al., 2022).

Typically, SPNs leverage a set of learnable queries to model the interaction among instances (entities or relation triples) via attention mechanism and generate the set of instances naturally. The success of SPNs on NER and RE inspires us to explore the possibility of jointly solving the extraction of entities and relation triples with SPNs, which is a promising but unexplored direction.

In this paper, we propose Bipartite Set Prediction Network (BiSPN), a variant of SPNs, to generate target entity set and relation set in one pass. It can not only avoid the negative effect of cascade error but also enjoy the benefit of high inference speed. However, it is challenged by the difficulty to maintain the coherence between the generated entity set and relation set, due to its parallel design.

As illustrated in Figure 1, the head/tail entities of generated relation triples should be included in the generated entity set. The significance of this coherence is two-fold: 1) by requiring the generated entity set to contain the head/tail entities, the recall of the generated entities is more guaranteed; 2) by restricting the head/tail entities within the generated entity set, the precision and recall of the generated triples are more guaranteed.

Despite that, it is difficult to maintain such consistency when generating the two sets in parallel, since all instance queries are assumed to be of equal status and their hidden representations are updated.

Interestingly, it has been described that GSK-3 can bind E-cadherin to phosphorilate it, which would result in a tighter interaction with • catenin.
using bidirectional attention without any further restriction.

To overcome this challenge, we come up with two novel solutions. The first one is a Bipartite Consistency Loss function. It works by looking for a reference entity from the generated entity set for each relational subject/object and forcing the subject/object to simulate the reference entity. Symmetrically, it also finds a reference subject/object for each entity classified as involved in relation and force the entity to simulate the reference subject/object. Our second solution is an Entity-Relation Linking Loss function, which works in the hidden semantic space. By computing the linking scores between the projected representations of entity queries and relation queries, it encourages the model to learn the interaction between entity instances and relation triple instances.

To sum up, our main contributions include:

- We present BiSPN, the first SPN-based joint entity and relation extraction model, which is able to generate target entity set and relation set in one pass.
- To maintain the coherence between the generated entity set and relation set, two novel loss functions, i.e., Bipartite Consistency Loss and Entity-Relation Linking Loss are introduced.
- BiSPN outperforms SOTA methods on three biomedical/clinical datasets, which is knowledge-intensive, and achieves competitive results on a general-domain benchmark. Besides, it infers much faster than two-stage joint extraction methods.

2 Related Work

2.1 Joint Entity and Relation Extraction

The target of joint entity and relation extraction is to predict the set of entities \( \{ e_i \}_{i=1}^{N_e} \) and the set of relation triples \( \{ r_j \}_{j=1}^{N_r} \) mentioned in the sentence. Here, \( L \) is the length of the input sentence, \( N_e \) and \( N_r \) are the numbers of target entities and target relation triples respectively. The \( i \)-th entity \( e_i \) is denoted as \((\text{start}_i, \text{end}_i, t)^{e}_{h} \), where \( \text{start}_i \) and \( \text{end}_i \) are the start token index and end token index of the entity, \( t^{e}_{h} \in \mathcal{T}_e \) is the entity type label. The \( j \)-th relation triple \( r_j \) is denoted as \((e^{h}_{j}, r^{e}_{j}, e^{t}_{j})\), where \( e^{h}_{j} \) and \( e^{t}_{j} \) are the head entity \((\text{start}^{h}_{j}, \text{end}^{h}_{j}, t^{e}_{h})\) and tail entity \((\text{start}^{t}_{j}, \text{end}^{t}_{j}, t^{e}_{t})\) of the relation triple, \( t^{e}_{j} \in \mathcal{T}_r \) is the relation type label. We additionally define a null label \( \emptyset \) for the set of entity types \( \mathcal{T}_e \) and the
set of relation types $\mathcal{T}_r$ respectively, to indicate that no entity or no relation is recognized.

3.2 BiSPN

As illustrated in Figure 2, the proposed model mainly consists of a shared encoder, a shared decoder, an entity decoding module (in blue) and a relation decoding module (in red).

3.2.1 Shared Encoder

The encoder of BiSPN is essentially a bidirectional pretrained language model (Devlin et al., 2019) with modified input style and attention design.

We first transform the input sentence $x$ into input embeddings $X \in \mathbb{R}^{L \times d}$, and then concatenate $X$ with a series of learnable entity queries $Q_e$ and relation queries $Q_r$ to form the model input $\tilde{X}$:

$$\tilde{X} = [X; Q_e; Q_r] \in \mathbb{R}^{(L + M_e + M_r) \times d} \quad (1)$$

where $d$ is the model dimension, $M_e$ and $M_r$ are hyperparameters controlling the number of entity queries and the number of relation queries ($M_e \gg N_e$, $M_r \gg N_r$).

To prevent the randomly initialized queries from negatively affecting the contextual token encodings, we follow the work (Shen et al., 2022) to modify the bidirectional self-attention into one-way self-attention. Concretely, the upper right $L \times (M_e + M_r)$ sub-matrix of the attention mask is filled with negative infinity value so that the entity/relation queries become invisible to the token encodings, while the entity/relation queries can still attend to each other and the token encodings.

After multiple one-way self-attention layers and feed-forward layers, the encoder outputs the contextual token encodings as well as the contextual entity/relation queries.

3.2.2 Shared Decoder

The shared decoder consists of $N$ decoding blocks. Each decoding block includes an one-way self-attention layer (as described above), a bidirectional self-attention layer and feed-forward layers. The one-way self-attention layer here functions as the cross-attention layer of Transformer decoder (Vaswani et al., 2017), which aggregates textual context for decoding. (The main difference between one-way self-attention and cross-attention is that the contextual token encodings also get updated by one-way self-attention.) The bidirectional self-attention layer updates the entity/relation...
queries via modeling the interaction among them.

After shared decoding, the decoder outputs the updated token representations $H^x$, entity queries $H^e$ and relation queries $H^r$.

### 3.2.3 Entity Decoding Module

The entity decoding module consists of an entity-view projection layer, an entity decoder and an entity predictor.

The entity-view projection layer first linearly transforms the token encodings $H^x$ into entity-view:

$$H^e = \text{Linear}(H^x) \quad (2)$$

The entity decoder, which includes multiple layers of cross-attention and bidirectional self-attention, receives the transformed token encodings $H^e$ as decoding context and the entity queries $H_e$ as decoder input, and output the final representation of entity queries $H^e$:

$$\tilde{H}^e = \text{EntityDecoder}(H^r|H^e) \quad (3)$$

The entity predictor is responsible for predicting the boundary and entity type of each entity query.

For each entity query, it first fuse the query representation with the transformed token encodings, and then calculate the probability of each token in the sentence being the start/end token of the corresponding entity:

$$S^e_i = \text{Linear}(\text{Relu}(\tilde{H}^e_i) + \text{Linear}(H^e_i)) \quad (4)$$

$$P^e_i = \text{Softmax}(S^e_i), \quad \delta \in \{\text{start}, \text{end}\} \quad (5)$$

where $S^e_i \in \mathbb{R}^L$ is a vector of logits of each token being the start/end token of the entity associated with $i$-th entity query, $P^e_i$ is the corresponding probability distribution.

An MLP-based classifier is leveraged to predict the type of the entity associated with the $i$-th entity query:

$$P^{e_{\text{ent}}} = \text{Softmax}(\text{MLP}(\tilde{H}^e_i)) \quad (6)$$

During inference, the predicted boundary and entity type corresponding to the $k$-th entity query are calculated as:

$$\text{score}_k(i, j) = P^{\text{start}}_k[i] + P^{\text{end}}_k[j] \quad (7)$$

$$\text{(start}_k, \text{end}_k) = \arg \max_{(i, j): 0 < j - i < L} \text{score}_k(i, j) \quad (8)$$

$$\hat{t}^e_k = \arg \max_{i, j} P^{e_{\text{ent}}}_k \quad (9)$$

Note that, the entity predictor will filter out the entity whose predicted type label is $\emptyset$.

### 3.2.4 Relation Decoding Module

The relation decoding module consists of a relation-view projection layer, a relation decoder, a head-tail predictor and a relation type predictor.

The relation-view projection layer and relation decoder work in the same manner as the entity-view projection layer and entity decoder, except that the relation decoder splits relation queries into head/tail queries before decoding:

$$[H^h, H^t] = \text{Linear}(H^r) \quad (10)$$

$$H^r = \text{Linear}(H^x) \quad (11)$$

$$\tilde{H}^h, \tilde{H}^t, \tilde{H}^r = \text{RelationDecoder}(H^h, H^t, H^r|H^x) \quad (12)$$

The head-tail predictor then predicts the boundary and entity type of the head/tail entity associated with each relation query. This process is similar to the entity prediction process (Equation 4-10). The only difference is that the entities queries becomes the head/tail queries $\tilde{H}^{h/t}$ and the token encodings is now in relation-view $H^r$.

The relation type predictor classifies the category of $i$-th relation query according to $\tilde{H}^r_i$:

$$P^{r_{\text{rel}}} = \text{Softmax}(\text{MLP}(\tilde{H}^r_i)) \quad (13)$$

### 3.3 Prediction Loss

To train the model, we should find the optimal assignment between the gold entity set and the generated entity set, as well as the optimal assignment between the gold relation set and the generated relation set, which are calculated in the same way as in (Tan et al., 2021; Shen et al., 2022) using the Hungarian algorithm (Kuhn, 1955).

After the optimal assignments are obtained, we calculate the following prediction loss $L_{\text{pred}}$ for each sample:

$$L_{\text{ent}} = - \sum_{i=1}^{M_e} (\log P^{\text{start}}_{i,[\text{start}_{i}(1)]} + \log P^{\text{end}}_{i,[\text{end}_{i}(1)]}$$

$$+ \log P^{e_{\text{ent}}}_{i,[\text{t}^e_{i}(1)]}) \quad (14)$$

$$L^{\text{h/t}}_{\text{ent}} = - \sum_{j=1}^{M_r} (\log P^{\text{start}}_{j,[\text{start}_{j}(1)]} + \log P^{\text{end}}_{j,[\text{end}_{j}(1)]}$$

$$+ \log P^{r_{\text{rel}}}_{j,[\text{t}^e_{j}(1)]}) \quad (15)$$

$$L_{\text{rel}} = L^{\text{h/t}}_{\text{ent}} - \sum_{j=1}^{M_r} \log P^{r_{\text{rel}}}_{j,[\text{t}^e_{j}(1)]} \quad (16)$$

$$L_{\text{pred}} = L_{\text{ent}} + L_{\text{rel}} \quad (17)$$

where $\varphi(i)$ is the index of the gold entity assigned to the $i$-th generated entity, $\sigma(j)$ is the index of the
gold relation triple assigned to the \( j \)-th generated relation triple, \( L^\text{ent}_{\text{ent}} \) represents the loss of head/tail entity prediction.

### 3.4 Bipartite Consistency Loss

To calculate the bipartite consistency loss, we first find a reference entity from the generated entity set for each head/tail entity. A reference entity is defined as the entity most similar to the referring head/tail entity. Concretely, the similarity between \( e_a \), the \( \alpha \)-th generated entity, and \( e^\text{h/t}_b \), the head/tail entity of the \( b \)-th generated relation triple is measured in KL divergence between the start/end/type probability distributions of \( e_a \) and \( e^\text{h/t}_b \):

\[
\text{sim}(e_a, e^\text{h/t}_b) = \sum_{\delta \in \text{\{start,end,type\}}} -D_{KL}(P^\delta_a || P^\delta^\text{h/t}_b)
\]

where \( D_{KL}(P||Q) \) is the KL divergence between target distribution \( P \) and approximate distribution \( Q \): \( P^\delta^\text{h/t}_b \) means \( P^\delta_b \) when \( e^\text{h/t}_b \) is a head entity, otherwise \( P^\delta^\text{h/t}_b \) means \( P^\delta^\text{t} \).

We want every head/tail entity to simulate its reference entity, which is equivalent to maximizing the similarity. Hence, the consistency loss in the relation \( \rightarrow \) entity direction is computed as:

\[
L_{\text{rel} \rightarrow \text{ent}} = -\sum_{i=1}^{M_e} \left[ \max_{j \in (1,M_e)} \text{sim}(e_j, e^\text{h/t}_i) \right] + \max_{j \in (1,M_e)} \left[ \text{sim}(e_j, e^\text{t}_i) \right]
\]

(18)

Symmetrically, we also find a reference head/tail entity for each generated entity that is classified as having relation. The classification is conducted by a binary classifier, which is trained with a binary cross-entropy loss function:

\[
P^\text{has-rel}_i = \text{sigmoid}(\text{MLP}(\vec{H}^c_i))
\]

(19)

\[
L_{\text{has-rel}} = -\frac{1}{M_e} \sum_{i=1}^{M_e} \left[ y_{i,\text{has-rel}} \log P^\text{has-rel}_i + (1 - y_{i,\text{has-rel}}) \log (1 - P^\text{has-rel}_i) \right]
\]

(20)

where \( y_{i,\text{has-rel}} = 1 \) only if the gold entity assigned to the \( i \)-th entity query is involved in relation.

The consistency loss in the entity \( \rightarrow \) relation direction is then calculated as follows:

\[
\text{sim}(e^\text{h/t}_b, e_a) = \sum_{\delta \in \{\text{start,end,type}\}} -D_{KL}(P^\delta_b || P^\delta_a)
\]

\[
L_{\text{ent} \rightarrow \text{rel}} = -\sum_{i=1}^{M_e} \max_{j \in (1,M_e)} \text{sim}(e^\text{h/t}_j, e_i)
\]

(21)

\[
\Omega = \{ i \mid P^\text{has-rel}_i \geq 0.5, i \in (1,M_e) \}
\]

(22)

where \( \Omega \) is the set of indices of the entities classified as involved in relation.

We sum up \( L_{\text{ent} \rightarrow \text{rel}} \), \( L_{\text{rel} \rightarrow \text{ent}} \) and \( L_{\text{has-rel}} \) to get the overall bipartite consistency loss \( L_{\text{ent} \rightarrow \text{rel}} \).

### 3.5 Entity-Relation Linking Loss

While the bipartite consistency loss softly aligns the predicted distributions between the generated entity set and relation set during training, the entity-relation linking loss encourages BiSPN to model the interaction between entity queries and relation queries.

To this end, we first project the intermediate representations of entity queries and relation queries and then compute the linking scores between them via a Biaffine layer:

\[
\vec{H}^e = \text{Linear}(H^e)
\]

(23)

\[
\vec{H}^r = \text{Linear}(H^r)
\]

(24)

\[
S^\text{link} = \text{Biaffine}(\vec{H}^e, \vec{H}^r) \in \mathbb{R}^{M_e \times M_r}
\]

(25)

With the linking scores, we calculate the following binary cross-entropy loss:

\[
P^\text{link}_i = \text{sigmoid}(S^\text{link}_{i,j})
\]

(26)

\[
L_{\text{link}} = -\frac{1}{M_e M_r} \sum_{i=1}^{M_e} \sum_{j=1}^{M_r} \left[ y^\text{link}_{i,j} \log P^\text{link}_{i,j} + (1 - y^\text{link}_{i,j}) \log (1 - P^\text{link}_{i,j}) \right]
\]

(27)

where \( y^\text{link}_{i,j} = 1 \) only if the gold entity assigned to the \( i \)-th entity query appears in the gold relation triple assigned to the \( j \)-th relation query.

### 4 Experiments

#### 4.1 Experiment Setups

<table>
<thead>
<tr>
<th>Statistics</th>
<th>ACE05</th>
<th>BioRelEx</th>
<th>ADE</th>
<th>Text2DT</th>
</tr>
</thead>
<tbody>
<tr>
<td># Sentences</td>
<td>14525</td>
<td>2010</td>
<td>4272</td>
<td>500</td>
</tr>
<tr>
<td>Avg sent. length</td>
<td>14.6</td>
<td>29.0</td>
<td>20.3</td>
<td>66.5</td>
</tr>
<tr>
<td># Entity types</td>
<td>7</td>
<td>33</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td># Relation types</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td># Entities</td>
<td>38287</td>
<td>9871</td>
<td>17808</td>
<td>3280</td>
</tr>
<tr>
<td># Relations</td>
<td>7691</td>
<td>3235</td>
<td>6821</td>
<td>3196</td>
</tr>
<tr>
<td># Ent. per sent.</td>
<td>2.63</td>
<td>4.91</td>
<td>4.17</td>
<td>6.56</td>
</tr>
<tr>
<td># Rel. per sent.</td>
<td>0.53</td>
<td>1.61</td>
<td>1.60</td>
<td>6.39</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the ACE05, BioRelEx, ADE and Text2DT datasets.
Datasets. We experiment on one general-domain dataset (ACE05) and three knowledge-intensive datasets (BioRelEx, ADE, Text2DT). **ACE05** (Walker et al., 2006) includes a corpus of newswire, broadcast news, telephone conversations. **BioRelEx** (Khachatrian et al., 2019) contains biomedical literature about binding interactions between proteins and/or biomolecules. **ADE** (Gurulingappa et al., 2012) consists of medical reports describing drug-related adverse effects. **Text2DT** is originally an benchmark for medical Decision Trees extraction task in China Health Information Processing Conference 2022. And we only use its entity/relation annotation for experiments. See Table 1 for detailed statistics of the datasets.

We additionally experiment on the **SciERC** dataset, where we follow the same setting as in (Wang et al., 2021; Ye et al., 2022). See Appendix B for the results.

Evaluation Metrics. Strict evaluation metrics are applied, where an entity is confirmed correct only if its boundary and entity type are correctly predicted; a relation triple is confirmed correct only if its relation type and head/tail entity are correctly predicted. For ACE05 and BioRelEx, we report the averaged Micro F1 scores over 3 random seeds. For ADE, we follows (Ji et al., 2020; Lai et al., 2021) to conduct 10-fold cross-validation and report the averaged Macro F1 scores. For Text2DT, we follows the top-1 system on the evaluation task to ensemble 5 models trained with different random seeds and report the Micro F1 scores.

### Table 3: Inference speed comparison on the ACE05, BioRelEx datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>ACE05</th>
<th>BioRelEx</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rel-F1</td>
<td>sent/s</td>
</tr>
<tr>
<td>KEIC (2021)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>UniRE (2021)</td>
<td>64.3</td>
<td><strong>134.6</strong></td>
</tr>
<tr>
<td>PL-Marker (2022)</td>
<td><strong>66.5</strong></td>
<td>32.0</td>
</tr>
<tr>
<td>BiSPN (Ours)</td>
<td>65.5</td>
<td>59.7</td>
</tr>
</tbody>
</table>
MADR (Haq et al., 2023): A pipeline of independent NER and RE models.
PL-Marker (Ye et al., 2022): A span-based method that models the interrelation between spans by packing levitated markers in the encoder.
PFN (Yan et al., 2021): A partition filter network that models two-way interaction between NER and RE subtasks.
UniRE (Wang et al., 2021): A method based on table filling, featured with a unified label space for one-stage joint entity and relation extraction.

TOP1: The top-1 system on the Text2DT evaluation task, which combines PFN (Yan et al., 2021) with Efficient GlobalPointer (Su, 2022).

Note that, we do not compare with SPN (Sui et al., 2020), UniRel (Tang et al., 2022) and QIDN (Tan et al., 2022), since these methods can only extract relation triples and cannot recognize those entities uninvolved in relation.

4.4 Main Results

Table 2 summarizes the overall performance of BiSPN and compared baselines on ACE05, BioRelEx, ADE and Text2DT. In terms of entity extraction, BiSPN performs competitively with or slightly better than SOTA methods on ACE05, BioRelEx and ADE, while outperforming the SOTA method PL-Marker by 0.8 F1 on Text2DT.

In terms of relation extraction, BiSPN boosts SOTA performance by 0.7, 0.5 and 0.5 F1 on BioRelEx, ADE and Text2DT respectively, verifying the effectiveness of our method on knowledge-intensive scene. However, although BiSPN outperforms SOTA one-stage methods by 0.6 F1 on ACE05, it is no match for the two-stage method PL-Marker on this general-domain dataset. We will look into the reason behind this in Section 4.6.2.

4.5 Inference Efficiency

We compare the inference efficiency of KEIC, UniRE, PL-Marker, BiSPN on the BioRelEx and Text2DT datasets. For a fair comparison, the experiments are all conducted on a server with Intel(R) Xeon(R) E5-2698 CPUs and NVIDIA Tesla V100 GPUs. And we fix the batch size as 8 during evaluation. As shown in Table 3, BiSPN can process around 40~60 sentences per second, which is 2 times faster than the SOTA two-stage method PL-Marker. Although UniRE, a SOTA one-stage method, is about 2.5 times faster than BiSPN, its performance of relation extraction is uncompetitive against BiSPN.

![Visualization of attention between entity queries and relation queries of a sample from BioRelEx.](image)

The input text in this sample is “Moreover, the in vitro binding of NF-B or Sp1 to its target DNA was not affected by the presence of K-12”.

Figure 3: Visualization of attention between entity queries and relation queries of a sample from BioRelEx.

![Table 4: Relation F1 scores of BiSPN on ACE05 with different settings of knowledge density.](image)

<table>
<thead>
<tr>
<th>Method</th>
<th># Rel $\geq$ 0</th>
<th># Rel $\geq$ 1</th>
<th># Rel $\geq$ 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PL-Marker</td>
<td>66.5</td>
<td>63.8</td>
<td>57.9</td>
</tr>
<tr>
<td>BiSPN</td>
<td>65.5 (-1.0)</td>
<td>63.4 (-0.4)</td>
<td>58.2 (+0.3)</td>
</tr>
</tbody>
</table>

Table 4: Relation F1 scores of BiSPN on ACE05 with different settings of knowledge density. The numbers in each bracket are the numbers of training, testing samples in the dataset under specific setting.

4.6 Analysis

4.6.1 Effects of $\mathcal{L}_{\text{ent-rel}}$ and $\mathcal{L}_{\text{link}}$

We conduct ablation study and qualitative visualization to analyze how the bipartite consistency loss $\mathcal{L}_{\text{ent-rel}}$ and entity-relation linking loss $\mathcal{L}_{\text{link}}$ work.

The results of ablation study are shown in Table 2. Without the bipartite consistency loss, the entity F1 scores drop by 0.5, 1.4, 0.7, 0.3 and the relation F1 scores drop by 1.5, 0.8, 0.2, 0.5 on ACE05, BioRelEx, ADE and Text2DT respectively. Without the entity-relation linking loss, the entity F1 scores decrease by 0.2, 0.1, 0.3, 0 and the relation F1 scores decrease by 0.4, 0.3, 0, 0.3 on the four datasets respectively. After removing the bipartite consistency loss and entity-relation linking loss together, the entity F1 scores decrease by 0.8, 1.8, 1.2, 0.9 and the relation F1 scores decrease by 2.1, 1.0, 0.7, 0.8 on the four datasets respectively.

Three conclusions can be derived from these results: 1) both the bipartite consistency loss and
entity-relation linking loss contribute to the overall performance of BiSPN; 2) the bipartite consistency loss is much more effective than the entity-relation linking loss; 3) the effects of the two losses are not orthogonal but still complementary in some degree.

To visualize the attention between entity queries and relation queries, we record the attention weight in the last bidirectional self-attention layer of shared decoder for a sample from BioRelEx. Since the original weight is bidirectional, we average the weight over two directions and conduct normalization to obtain the weight for visualization. As shown in Figure 3, without $L_{\text{ent} \leftrightarrow \text{rel}}$ and $L_{\text{link}}$, the interaction between entity queries and relation queries is chaotic and pointless. After applying $L_{\text{ent} \leftrightarrow \text{rel}}$ or $L_{\text{link}}$ separately, the relation queries associated with gold relation triples tend to focus on the entity queries associated with gold entities. When applying the two loss functions together, the relation queries associated with gold relation triples further concentrate on the entity queries whose target entities are the same as their head/tail entities. This phenomenon coincides with the purpose of the two loss functions.

### 4.6.2 Influence of Knowledge Density

The performance of BiSPN on ACE05 is not so good as its performance on BioRelEx, ADE and Text2DT. We hypothesize it is the sparsity of relation triples that hinders the learning of BiSPN. As listed in Table 1, the average number of relations per sentence is 0.53 on ACE05. In contrast, the average numbers of relations per sentence are 1.61, 1.60 and 6.39 on the other three datasets.

To verify our hypothesis, we filter samples according to the number of relations they contain and experiment on different versions of the filtered dataset. As shown in Table 4, when the samples without relation are discarded, the performance gap between PL-Marker and BiSPN narrows from 1.0 to 0.4. When further discarding the samples with less than 2 relation triples, BiSPN even performs slightly better than PL-Marker. This reveals that the strength of BiSPN emerges in knowledge-intensive scene, which is reasonable, since BiSPN works by modeling interaction among knowledge instances.

### 4.7 Case Study

Figure 4 illustrates two test cases from ACE05 and BioRelEx respectively. In the first case, BiSPN without $L_{\text{ent} \leftrightarrow \text{rel}}$ and $L_{\text{link}}$ fails to recognizes the WEA entity “more” and the PART-WHOLE relation between “it” and “more”, while BiSPN successfully extracts them by considering the context in entity-view and relation-view concurrently. Likewise, in the second case, BiSPN successfully recognizes the entity “promoter” after $L_{\text{ent} \leftrightarrow \text{rel}}$ and $L_{\text{link}}$ is applied. However, it still fails to recognize “RNA polymerase II preinitiation complex”, which is complicated and may require domain knowledge for recognition.

### 5 Conclusion

In this work, we present BiSPN, a novel joint entity relation extraction framework based on bipartite set prediction. It generates entity set and relation set in a distributed manner, so as to avoid error propagation. To maintain the coherency between the generated entity set and relation set, we come up with two novel loss designs, namely bipartite consistency loss and entity-relation linking loss. The first one pulls closer the predicted boundary/type distributions of entities and head/tail entities, while the second one enforces the interaction between entity queries and relation queries. Extensive experiments demonstrate the advantage of BiSPN in
knowledge-intensive scene, as well as the effectiveness of the proposed bipartite consistency loss and entity-relation linking loss.

Limitations

As mentioned in Section 4.6.2, the performance of our BiSPN framework can be hindered when the distribution of relation triples are overly sparse in a dataset. This suggests that BiSPN is a better choice for biomedical and clinical domains but not the general-domain, where knowledge sparsity is common.

Another limitation of BiSPN is its reliance on a fixed number of entity/relation queries. Although it is possible to set the number of queries larger in order to let BiSPN generalize to longer text input, the cost is additional memory and time consumption that grows quadratically. To effectively address this, future work can draw lessons from the field of dynamic neural network and explore dynamic selection of instance queries.

Acknowledgments

We thank the reviewers for their valuable suggestions. This study is partially supported by National Key R&D Program of China (2021ZD0113402), National Natural Science Foundation of China (62276082), Major Key Project of PCL (PCL2021A06), Shenzhen Soft Science Research Program Project (No.KX20220705152815035) and the Fundamental Research Fund for the Central Universities (HIT.DZJJ.2023117).

References


A Hyperparameter Configuration

We tune the hyperparameters for each dataset by manually trying different values of each hyperparameter within a specific interval and choosing the
value that results in the highest relation F1 on the development set (For ADE, the validation set of its first fold of data is employed as the development set). After trial, we find it optimal to set the numbers of shared decoder layers, entity decoder layers and relation decoder layers as 4, 1, 1 for all datasets. The trial intervals and final configuration of other hyperparameters are shown in Table 5.

B Experiment Results on the SciERC dataset

Here, we append the results of additional experiment on the SciERC dataset. As shown in Table 6, in terms of entity recognition, BiSPN underperforms SOTA two-stage method PL-Marker by Ent-F1, but still outperforms SOTA one-stage methods (PFN, UniRE) substantially. In terms of relation triple extraction, BiSPN establishes new SOTA (Rel-F1) on the SciERC dataset. In terms of inference speed, BiSPN is about faster than PL-Marker, but slower than other one-stage methods. The results after ablating the consistency loss and the linking loss verify the effectiveness of them on the dataset.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Trial Interval</th>
<th>ACE05</th>
<th>BioRelEx</th>
<th>ADE</th>
<th>Text2DT</th>
<th>SciERC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epochs</td>
<td>[50, 100]</td>
<td>70</td>
<td>75</td>
<td>70</td>
<td>80</td>
<td>70</td>
</tr>
<tr>
<td>Warmup Rate</td>
<td>[0.01, 0.2]</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Decoder $lr$</td>
<td>[1e-5, 5e-5]</td>
<td>4e-5</td>
<td>4e-5</td>
<td>4e-5</td>
<td>4e-5</td>
<td>4e-5</td>
</tr>
<tr>
<td>Batch Size</td>
<td>[2, 64]</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>$M_e$</td>
<td>[20, 40]</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>35</td>
</tr>
<tr>
<td>$M_r$</td>
<td>[20, 40]</td>
<td>35</td>
<td>25</td>
<td>25</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>[1e-5, 1e-2]</td>
<td>1e-4</td>
<td>1e-4</td>
<td>1e-3</td>
<td>1e-4</td>
<td></td>
</tr>
<tr>
<td>$\beta$</td>
<td>[1e-5, 1e-2]</td>
<td>1e-3</td>
<td>1e-4</td>
<td>1e-4</td>
<td>1e-3</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Configuration of Hyperparameters. $lr$ represents the initial learning rate. $M_e$, $M_r$ are the number of entity queries and the number of relation queries respectively. $\alpha$, $\beta$ are the weights of the bipartite consistency loss and entity-relation linking loss respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>Ent-F1</th>
<th>Rel-F1</th>
<th>sent/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFN (2021)</td>
<td>66.8</td>
<td>38.4</td>
<td>55.1</td>
</tr>
<tr>
<td>UniRE (2021)</td>
<td>68.4</td>
<td>36.9</td>
<td>79.6</td>
</tr>
<tr>
<td>PL-Marker (2022)</td>
<td>69.9</td>
<td>41.6</td>
<td>18.5</td>
</tr>
<tr>
<td>BiSPN (Ours)</td>
<td>68.9</td>
<td>42.0</td>
<td>37.2</td>
</tr>
<tr>
<td>w/o $L_{ent}$</td>
<td>67.7</td>
<td>41.4</td>
<td>-</td>
</tr>
<tr>
<td>w/o $L_{link}$</td>
<td>68.0</td>
<td>41.3</td>
<td>-</td>
</tr>
<tr>
<td>w/o $L_{ent+rel}$</td>
<td>67.1</td>
<td>40.8</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6: Results on the SciERC dataset.