PeerDA: Data Augmentation via Modeling Peer Relation for Span Identification Tasks

Anonymous ACL submission

Abstract

Span identification aims at identifying specific text spans from a text input and classifying them into pre-defined categories. Different from previous works that merely leverage the Subordinate (SUB) relation (i.e., if a span is an instance of a certain category) to train models, this paper for the first time explores the Peer (PR) relation, which indicates that two spans are instances of the same category and share similar features. Specifically, a novel Peer Data Augmentation (PeerDA) approach is proposed which employs span pairs with the PR relation as the augmentation data for training. PeerDA has two unique advantages: (1) There are a large number of PR span pairs for augmenting the training data. (2) The augmented data can prevent the trained model from over-fitting the superficial span-category mapping by pushing the model to leverage the span semantics. Experimental results on ten datasets over four diverse tasks across seven domains demonstrate the effectiveness of PeerDA. Notably, PeerDA achieves state-of-the-art results on six of them.¹

1 Introduction

Span Identification (SpanID) is a family of Natural Language Processing (NLP) tasks with the goal of detecting specific text spans and further classifying them into pre-defined categories (Papay et al., 2020). It serves as the initial step for complex text analysis by narrowing down the search scopes of important spans, which holds a pivotal position in the field of NLP (Ding et al., 2021). Recently, different domain-specific SpanID tasks, such as social media Named Entity Recognition (NER) (Derczynski et al., 2017), Aspect Based Sentiment Analysis (ABSA) (Liu, 2012), Contract Clause Extraction (CCE) (Chalkidis et al., 2017) and Span Based Propaganda Detection (SBPD) (Da San Martino et al., 2019), have emerged for various NLP applications.

¹Our code and data are available at github.com/XXX.

PeerDA: Data Augmentation via Modeling Peer Relation for Span Identification Tasks

Anonymous ACL submission

Abstract

Span identification aims at identifying specific text spans from a text input and classifying them into pre-defined categories. Different from previous works that merely leverage the Subordinate (SUB) relation (i.e., if a span is an instance of a certain category) to train models, this paper for the first time explores the Peer (PR) relation, which indicates that two spans are instances of the same category and share similar features. Specifically, a novel Peer Data Augmentation (PeerDA) approach is proposed which employs span pairs with the PR relation as the augmentation data for training. PeerDA has two unique advantages: (1) There are a large number of PR span pairs for augmenting the training data. (2) The augmented data can prevent the trained model from over-fitting the superficial span-category mapping by pushing the model to leverage the span semantics. Experimental results on ten datasets over four diverse tasks across seven domains demonstrate the effectiveness of PeerDA. Notably, PeerDA achieves state-of-the-art results on six of them.¹

1 Introduction

Span Identification (SpanID) is a family of Natural Language Processing (NLP) tasks with the goal of detecting specific text spans and further classifying them into pre-defined categories (Papay et al., 2020). It serves as the initial step for complex text analysis by narrowing down the search scopes of important spans, which holds a pivotal position in the field of NLP (Ding et al., 2021). Recently, different domain-specific SpanID tasks, such as social media Named Entity Recognition (NER) (Derczynski et al., 2017), Aspect Based Sentiment Analysis (ABSA) (Liu, 2012), Contract Clause Extraction (CCE) (Chalkidis et al., 2017) and Span Based Propaganda Detection (SBPD) (Da San Martino et al., 2019), have emerged for various NLP applications.

¹Our code and data are available at github.com/XXX.
the semantics of the given span but make predictions based on the memorized span-category patterns, which hurts the generalization capability of the models. 2) **Data Scarcity:** For low-resource scenarios or long-tailed categories, the number of span-category pairs with **R** relation (**R** pairs) could be very limited and insufficient to learn a reliable SpanID model.

In this paper, we explore the span-span **Peer** (**P**) relation to alleviate the above limitations. Specifically, the **P** relation indicates that two spans are two different instances of the same category. The major difference between **P** relation and **R** relation is that the former one intends to correlate two spans without giving the categories they belong to. For example, in Figure 1 (a), "Hawaii" and "London" are connected with the **P** relation because they are instances of the same category. By jointly recognizing **R** relation and **P** relation in the input text, the model is enforced to favor the usage of span semantics instead of span-category patterns for prediction, reducing the risk of over-fitting. In addition, the number of span-span pairs with the **P** relation (**P** pairs) grows quadratically over the number of **R** pairs. Therefore, we can still construct a reasonable number of training data with **P** pairs for categories having insufficient examples.

In this paper, with the aim of leveraging the **P** relation to enhance SpanID models, we propose a Peer Data Augmentation (**PeerDA**) approach that treats **P** pairs as a kind of augmented training data. To achieve this, as depicted in Figure 1 (b), we extend the usage of the original training data into two views. The first view is the **R**-based training data. It is used to directly solve the SpanID tasks by extracting the **R** relation, which is the typical formulation of MRC-based approaches. The second view is the **P**-based training data. It is our augmentation to enrich the semantics of spans by extracting the **P** relation in the original training data, where one span is used to identify its peer from the input context. Note that our **P**-based training data can be easily formulated into the MRC paradigm. Therefore, the knowledge learned from such augmentation data can be directly transferred to enhance the model’s capability to capture **R** relation (i.e., the SpanID tasks).

To better accommodate the MRC-style **R** and **P** data, we develop a stronger and more memory-efficient MRC model. Compared to the designs in Li et al. (2020b), our model introduces a bilinear component to calculate the span scores and consistently achieves better performance with a 4 times smaller memory consumption. Besides, we propose a margin-based contrastive learning strategy to additionally model the negative spans to the query (e.g., when querying the context in Figure 1 for "ORG" entities, "London" becomes a negative span) so that the spans from different categories are separated more apart in the semantic space.

We evaluate the effectiveness of **PeerDA** on ten datasets across seven domains, from four different SpanID tasks, namely, NER, ABSA, CCE, and SBPD. Experimental results show that extracting **P** relation benefits the learning of semantics and encourages models to identify more possible spans. As a result, **PeerDA** is a new state-of-the-art (**SOTA**) method on six SpanID datasets. Our analyses further demonstrate the capability of **PeerDA** to alleviate scarcity and over-fitting issues.

Our contributions are summarized as follows:

1) We propose a novel **PeerDA** approach to tackle SpanID tasks via augmenting training data with **P** relation.
2) We conduct extensive experiments on ten datasets, including four different SpanID tasks across seven domains, and achieve SOTA performance on six SpanID datasets.
3) **PeerDA** is more effective in low-resource scenarios or long-tailed categories and thus, it alleviates the scarcity issue. Meanwhile, **PeerDA** pushes models to weigh more on the span semantics to prevent over-fitting.

## 2 Related Work

**DA for SpanID:** DA, which increases the diversity of training data at a low cost, is a widely-adopted solution to address data scarcity (Feng et al., 2021). In the scope of SpanID, existing DA approaches aim to introduce more span-category patterns, including: (1) **Word Replacement** that keeps the labels unchanged but replaces or paraphrases some context tokens either using simple rules (Wei and Zou, 2019; Dai and Adel, 2020) or strong language models (Kobayashi, 2018; Wu et al., 2019; Li et al., 2020a). (2) **Self-training** is to continually train the model on its predicted data (Xie et al., 2019, 2020), which shows promising results on NER (Wang et al., 2020), and propaganda detection (Hou et al., 2021). (3) **Distantly Supervised Training** focuses on leveraging external knowledge to roughly label spans in the target tasks. For example, Huang et al. (2021) leverage Wikipedia to create distant
labels for NER. Chen et al. (2021) transfer data from high-resource to low-resource domains. Jain et al. (2019); Li et al. (2020c) tackle cross-lingual NER by projecting labels from high-resource to low-resource languages. Differently, the motivation of PeerDA is to leverage the augmented data to enhance models’ capability on semantic understanding by minimizing(maximizing) the distances between semantically similar(distant) spans.

MRC: MRC is to extract an answer span from a relevant context conditioned on a given query. It is initially designed to solve question answering tasks (Hermann et al., 2015), while recent trends have shown great advantages of formulating NLP tasks as MRC problems. In the context of SpanID, Li et al. (2020b) address the nested NER issues by decomposing nested entities under multiple queries. Mao et al. (2021) tackle ABSA by combining aspect term extraction and sentiment polarity classification in a dual MRC framework. Hendrycks et al. (2021) tackle CCE with MRC to deal with the extraction of long clauses. Moreover, other tasks such as relation extraction (Li et al., 2019a), event detection (Liu et al., 2020, 2021), and summarization (McCann et al., 2018) are also reported to benefit from the MRC paradigm.

3 PeerDA

Overview of SpanID: Given the input text $X = \{x_1, ..., x_n\}$, SpanID is to detect all appropriate spans $\{x_k\}_{k=1}^K$ and classify them with proper labels $\{y_k\}_{k=1}^K$, where each span $x_k = \{x_{s_k}, x_{s_k}+1, ..., x_{s_k-1}, x_{c_k}\}$ is a subsequence of $X$ satisfying $s_k \leq c_k$ and the label comes from a predefined category set $Y$ (e.g. "Person" in NER).

3.1 Training Data Construction

The training data $D$ consists of two parts: (1) The SUB-based training data $D^{\text{SUB}}$, where the query is about a category and the MRC context is the input text. (2) The PR-based training data $D^{\text{PR}}$ is constructed with PR pairs, where one span is used to create the query and the input text containing the second span serves as the MRC context.

3.1.1 SUB-based Training Data

First, we need to transform the original training examples into (query, context, answers) triples following the paradigm of MRC (Li et al., 2020b). To extract the SUB relation between categories and relevant spans, a natural language query $Q^{\text{SUB}}_y$ is constructed to reflect the semantics of each category $y$. Following Hendrycks et al. (2021), we include both category mention $\text{M}[y]$ and its definition $\text{D}[y]$ from the annotation guideline (or Wikipedia if the guideline is not accessible) in the query to introduce more comprehensive semantics:

$$Q^{\text{SUB}}_y = \text{Highlight the parts (if any) related to } \text{M}[y], \text{Details: } \text{D}[y]$$

Given the input text $X$ as the context, the answers to $Q^{\text{SUB}}_y$ are the spans belonging to category $y$. Then we can obtain one MRC example denoted as $(Q^{\text{SUB}}_y, X, \{x_k | x_k \in X, y_k = y\}_{k=1}^K)$. To guarantee the identification of all possible spans, we create $|Y|$ training examples by querying the input text with each pre-defined category.

3.1.2 PR-based training data

To construct augmented data that derived from the PR relation, we first create a category-wise span set $S_y$ that includes all training spans with category $y$:

$$S_y = \{x_k | (x_k, y_k) \in D^{\text{SUB}}, y_k = y\}$$

Obviously, any two different spans in $S_y$ have the same category and shall hold the PR relation. Therefore, we pair every two different spans in $S_y$ to create a peer set $P_y$:

$$P_y = \{(x^a, x^a') | x^a, x^a' \in S_y, x^a \neq x^a'\}$$

For each PR pair $(x^a, x^a')$ in $P_y$, we can construct one training example by constructing the query with the first span $x^a$:

$$Q^{\text{PR}}_y = \text{Highlight the parts (if any) similar to } x^a$$

Then we treat the text $X^a$ containing the second span $x^a$ as the MRC context to be queried and $x^a$ as the answer to $Q^{\text{PR}}_y$. Note that there may exist more than one span in $X^a$ satisfying PR relation with $x^a$, we set all of them as the valid answers to $Q^{\text{PR}}_y$, yielding one training example $(Q^{\text{PR}}_y, X^a, \{x_k^a | x_k^a \in X^a, y_k^a = y\}_{k=1}^K)$ of our PeerDA.

Theoretically, given the span set $S_y$, there are only $|S_y|$ SUB pairs in the training data but we can obtain $|S_y| \times (|S_y| - 1)$ PR pairs to construct $D^{\text{PR}}$. Such a large number of augmented data shall hold great potential to enrich spans’ semantics. However, putting all PR-based examples into training would exacerbate the skewed data distribution issue since the long-tailed categories get fewer PR pairs.
for augmentation and also increase the training cost. Therefore, as the first step for DA with the PR relation, we propose three augmentation strategies to control the size and distribution of augmented data.

**PeerDA-Size**: This is to increase the size of augmented data while keeping the data distribution unchanged. Specifically, for each category $y$, we randomly sample $\lambda|S_y|$ PR pairs from $P_y$. Then we collect all sampled PR pairs to construct $D_{Pr}$, where $\lambda$ is the DA rate to control the size of $D_{Pr}$.

**PeerDA-Categ**: Categories are not evenly distributed in the training data, and in general SpanID models perform poorly on long-tailed categories. To tackle this, we propose PeerDA-Categ to augment more training data for long-tailed categories. Specifically, let $y^*$ denote the category having the largest span set of size $|S_{y^*}|$. We sample up to $|S_{y^*}| - |S_y|$ PR pairs from $P_y$ for each category $y$ and construct a category-balanced training set $D_{Pr}$ using all sampled pairs. Except for the extreme cases where $|S_y|$ is smaller than $\sqrt{|S_{y^*}|}$, we would get the same size of the training data for each category after the augmentation, which significantly increases the exposure for spans from the long-tailed categories.

**PeerDA-Both** (The final version of PeerDA): To take advantage of the above two strategies, we further propose PeerDA-Both to maintain the data distribution while effectively increasing the size of training data. In PeerDA-Both, we randomly sample $\max(\lambda|S_{y^*}| + (|S_{y^*}| - |S_y|), 0)$ PR pairs from $P_y$ for each category $y$ to construct $D_{Pr}$, where $\lambda|S_{y^*}|$ determines the size of the augmented data, and $|S_{y^*}| - |S_y|$ controls the data distribution.

### 3.1.3 Data Balance

We combine the $D_{Sub}$ and the $D_{Pr}$ created above as the final training data. Since an input text usually mentions spans from a few categories, when converting the text into the MRC paradigm, many of the $|Y|$ examples are unanswerable. If a SpanID model is trained on this unbalanced data, then the model may favor the majority of the training examples and output an empty span. To balance answerable and unanswerable examples, we follow Hendrycks et al. (2021) to randomly remove some unanswerable examples from the training data.

### 3.2 Model Architecture

As shown in Figure 2, to achieve the detection of multiple spans for the given query, we follow Li et al. (2020b) to build the MRC model. Compared to the original designs, we further optimize the computation of span scores following a general way of Luong et al. (2015).

Specifically, the base model consists of three components: an encoder, a span predictor, and a start-end selector. First, given the concatenation of the query $Q$ and the context $X$ as the MRC input $\overline{X} = \{[CLS], Q, [SEP], X, [SEP]\}$, where $[CLS], [SEP]$ are special tokens, the encoder would encode the input text into hidden states $H$:

$$H = Encoder(\overline{X})$$ (5)

Second, the span predictor consists of two binary classifiers, one to predict whether each context token is the start index of the answer, and the other to predict whether the token is the end index:

$$P_{start} = HW^s \quad P_{end} = HW^e$$ (6)

where $W^s, W^e \in \mathbb{R}^{d \times 2}$ are the weights of two classifiers and $d$ is the dimension of hidden states. The span predictor would output multiple start and end indexes for the given query and context.

Third, the start-end selector matches each start index to each end index and selects the most possible spans from all combinations as the outputs. Different from the concat way that would create a large $\mathbb{R}^{|X| \times |X| \times 2d}$ shape tensor (Li et al., 2020b), we leverage a general way following Luong et al. (2015) to compute the span score, consuming fewer resources for better training efficiency:

$$P_{s,e} = FFN(H_s)^T H_e$$ (7)

where $FFN$ is the feed-forward network (Vaswani et al., 2017). $P_{s,e}$ denotes the likelihood of $X_{s,e}$ to form a possible answer.

### 3.3 Training Objective

The standard objective is to minimize the cross-entropy loss (CE) between above three predictions and their corresponding ground-truth labels, i.e., $Y_{start}, Y_{end}, Y_{s,e}$ (Li et al., 2020b):

$$\mathcal{L}_{mrc} = CE(\sigma(P_{start}), Y_{start}) + CE(\sigma(P_{end}), Y_{end}) + CE(\sigma(P_{s,e}), Y_{s,e})$$ (8)

where $\sigma$ is the sigmoid function.

However, these objectives only capture the semantic similarity between the query and positive
spans (i.e., the span instances of the query category). In this paper, we propose to explicitly separate the query and its negative spans (i.e., the span instances of other categories) apart with a margin-based contrastive learning strategy, for better distinguishing the spans from different categories.

Specifically, given the MRC input $X$ with query of category $y$, there may be multiple positive spans $\overline{X}^+ = \{s_k \in X, y_k = y\}$ and negative spans $\overline{X}^- = \{s_k' \in X, y_k' \neq y\}$. We leverage the following margin-based contrastive loss to penalize negative spans (Chechik et al., 2010):

$$L_{ct} = \max_{\forall s \in \overline{X}^+} \max(0, M - (\sigma(P_{s,e}) - \sigma(P_{s,e}^c)))$$

where $M$ is the margin term, $\max(\cdot, \cdot)$ is to select the larger one from two candidates, and the span score $P_{s,e}$ can be regarded as the semantic similarity between the query and the target span $s$. Note that our contrastive loss maximizes the similarity difference between the query and the most confusing positive and negative span pairs (Max-Min), which we demonstrate to be effective in Sec. 5.3.

Finally, the overall training objective is:

$$\mathcal{L} = \mathcal{L}_{mrc} + \alpha \mathcal{L}_{ct}$$

where $\alpha$ is the balance rate.

### 4 Experimental Setup

#### 4.1 Tasks

We conduct experiments on four SpanID tasks from diverse domains, including NER, ABSA, Contract Clause Extraction (CCE), and Span Based Propaganda Detection (SBPD). The dataset statistics are summarized in Table 1. The detailed task description can be found in Appendix A.1.

<table>
<thead>
<tr>
<th>Task</th>
<th>NER</th>
<th>ABSA</th>
<th>SBPD</th>
<th>CCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>OntoNotes5</td>
<td>WNUT17</td>
<td>Movie</td>
<td>Restaurant</td>
</tr>
<tr>
<td># Train</td>
<td>60.0k</td>
<td>3.4k</td>
<td>7.8k</td>
<td>7.7k</td>
</tr>
<tr>
<td># Test</td>
<td>8.3k</td>
<td>1.3k</td>
<td>2.0k</td>
<td>1.5k</td>
</tr>
<tr>
<td># Category</td>
<td>11</td>
<td>6</td>
<td>12</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 1: Statistics on the ten SpanID datasets. Note that 1 / 3 denotes that there is 1 category in ATE and 3 categories in UABSA. dev denotes that we evaluate News20 on the dev set.

WNUT17 (Derczynski et al., 2017), Movie (Liu et al., 2013b), and Restaurant (Liu et al., 2013a) and a Chinese dataset Weibo (Peng and Dredze, 2015). We use micro-averaged Precision, Recall, and F1 as evaluation metrics.

ABSA: We explore two ABSA sub-tasks: Aspect Term Extraction (ATE) to only extract aspect terms, and Unified Aspect Based Sentiment Analysis (UABSA) to jointly identify aspect terms and their sentiment polarities. We evaluate the two sub-tasks on two datasets, including the laptop domain Lap14 and restaurant domain Rest14. We use micro-averaged F1 as the evaluation metric.

SBPD: It aims to detect both the text fragment where a persuasion technique is used and its technique type. We use News20 and Social21 from SemEval shared tasks (Da San Martino et al., 2020; Dimitrov et al., 2021). For News20, we report the results on its dev set since the test set is not publicly available. We use micro-averaged Precision, Recall, and F1 as evaluation metrics.

CCE: It is a legal task to detect and classify contract clauses into relevant clause types, such as "Governing Law". We conduct CCE experiments using CUAD (Hendrycks et al., 2021). We follow Hendrycks et al. (2021) to use Area Under the Precision-Recall Curve (AUPR) and Precision at 80% Recall (P@0.8R) as the evaluation metrics.

#### 4.2 Implementations

Since legal SpanID tasks have a lower tolerance for missing important spans, we do not include start-end selector (i.e. CE($P_{s,e}$, $Y_{s,e}$) and $\alpha \mathcal{L}_{ct}$ in Eq. (10)) in the CCE models but follow Hendrycks et al. (2021) to output top 20 spans from span predictor for each input example in order to extract spans as much as possible. While for NER, ABSA, and SBPD, we use our optimized architecture and objective. For fair comparison with existing works, our models utilize BERT (Devlin et al., 2019) as the text encoder for ABSA and RoBERTa (Liu et al., 2019) for NER, CCE, and SBPD. Detailed
4.3 Baselines

Note that our main contribution is to provide a new perspective to treat the Pr relation as a kind of training data for augmentation. Therefore, we compare with models built on the same encoder-only PLMs (Devlin et al., 2019; Liu et al., 2019). We are not focusing on pushing the SOTA results to new heights though some of the baselines already achieved SOTA performance.

**NER:** We compare with Tagging (Liu et al., 2019) and MRC (Li et al., 2020b) baselines. We also report the previous best approaches for each dataset, including RB-CRF+RM (Lin et al., 2021), CL-KL (Wang et al., 2021), T-NER (Ushio and Camacho-Collados, 2021) KaNa (Nie et al., 2021), and RoBERTa+BS (Zhu and Li, 2022).

**ABSA:** In addition to MRC baseline, we also compare with previous approaches based on top of BERT. These are SPAN-BERT (Hu et al., 2019), IMN-BERT (He et al., 2019), RACL (Chen and Qian, 2020) and Dual-MRC (Mao et al., 2021).

**SBPD:** For News20 we only compare with MRC baseline due to the lack of related work. For Social21, we compare with top three approaches on its leaderboard, namely, Volta (Gupta et al., 2021), HOMADOS (Kaczyński and Przybyła, 2021), and TeamFPAI (Hou et al., 2021).

**CCE:** We compare with (1) MRC baseline, (2) stronger text encoders, including ALBERT (Lan et al., 2019) and DeBERTa (He et al., 2020), and (3) the model continually pretrained on contracts: RoBERTa + CP (Hendrycks et al., 2021).

Table 2: Performance on NER datasets. The best models are bolded.

<table>
<thead>
<tr>
<th>Methods</th>
<th>OntoNotes5</th>
<th>WNUT17</th>
<th>Movie</th>
<th>Restaurant</th>
<th>Weibo</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F1</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>SOTA</td>
<td>RB-CRF+RM</td>
<td>-</td>
<td>60.5</td>
<td>T-NER</td>
<td>KaNa</td>
</tr>
<tr>
<td></td>
<td>92.8</td>
<td>92.4</td>
<td>92.6</td>
<td>73.0</td>
<td>71.2</td>
</tr>
<tr>
<td>Tagging</td>
<td>91.0</td>
<td>91.8</td>
<td>91.4</td>
<td>62.1</td>
<td>48.2</td>
</tr>
<tr>
<td>MRC</td>
<td>92.4</td>
<td>91.8</td>
<td>92.1</td>
<td>66.4</td>
<td>40.7</td>
</tr>
<tr>
<td>PeerDA</td>
<td>91.9</td>
<td>92.6</td>
<td>92.4</td>
<td>71.1</td>
<td>46.9</td>
</tr>
</tbody>
</table>

**Large**

<table>
<thead>
<tr>
<th>Methods</th>
<th>Lap14</th>
<th>Rest14</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UABSA</td>
<td>ATE</td>
</tr>
<tr>
<td>SPAN-BERT</td>
<td>61.3</td>
<td>82.3</td>
</tr>
<tr>
<td>IMN-BERT</td>
<td>61.7</td>
<td>77.6</td>
</tr>
<tr>
<td>RACL</td>
<td>63.4</td>
<td>81.8</td>
</tr>
<tr>
<td>Dual-MRC</td>
<td>65.9</td>
<td>82.5</td>
</tr>
<tr>
<td>MRC (Large)</td>
<td>63.2</td>
<td>83.9</td>
</tr>
<tr>
<td>PeerDA</td>
<td>65.9</td>
<td>84.6</td>
</tr>
</tbody>
</table>

Table 3: Performance on two ABSA subtasks on two datasets. Results are averages F1 over 5 runs.

5 Results

5.1 Comparison Results

**NER:** Table 2 shows the performance on five NER datasets. Our PeerDA significantly outperforms the Tagging and MRC baselines. Precisely, compared to RoBERTa base MRC, PeerDA obtains 0.3, 1.6, 3.2, 1.5, and 2.9 F1 gains on five datasets respectively. When implemented on RoBERTa large, our PeerDA can further boost the performance and establishes new SOTA on three datasets, namely, OntoNotes5, Movie, and Restaurant. Note that the major improvement of PeerDA over MRC comes from higher Recall. It implies that PeerDA encourages models to give more span predictions.

**ABSA:** Table 3 depicts the results on ABSA. Compared to previous approaches, PeerDA mostly achieves better results on two subtasks, where it outperforms vanilla MRC by 2.7 and 1.0 F1 on UABSA for two domains respectively.

**SBPD:** The results of two SBPD tasks are presented in Table 4. PeerDA outperforms MRC by 8.2 and 9.2 F1, and achieves SOTA performance on News20 and Social21 respectively.

**CCE:** The results of CCE are shown in Table 5. PeerDA surpasses MRC by 8.7 AUPR and 13.3...
To explore how the size and category distribution by more than 4 times with no performance drop, the same experimental setup (RoBERTa for CCE) are averaged of all datasets in each task.

Table 7: Ablation study on model designs. The F₁ scores are averaged of all datasets in each task. The |GPU| column denotes the GPU memory footprint of each variant under the same experimental setup.

<table>
<thead>
<tr>
<th>Ablation Type</th>
<th></th>
<th>GPU</th>
<th></th>
<th>NER</th>
<th>UABSA</th>
<th>SBPD</th>
<th>CCE</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculation of ( P_{s,e} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>concat ( \text{general (final)} )</td>
<td>1x</td>
<td>74.5</td>
<td>69.2</td>
<td>40.3</td>
<td>61.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max-Min ( \text{final} )</td>
<td>0.23x</td>
<td>75.5</td>
<td>69.9</td>
<td>42.0</td>
<td>52.3</td>
<td>59.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contrastive Loss</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>0.23x</td>
<td>75.1</td>
<td>69.6</td>
<td>37.6</td>
<td>54.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max-Min ( \text{final} )</td>
<td>0.23x</td>
<td>75.0</td>
<td>69.4</td>
<td>40.8</td>
<td>51.3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Ablation study on data augmentation strategies. The results (F₁ for NER, UABSA, and SBPD, AUPR for CCE) are averaged of all datasets in each task.
Figure 3: Performance on low-resource scenarios. We select one dataset for each SpanID task and report the test results (AUPR for CCE and $F_1$ for others) from the models trained on different proportions of the training data.

<table>
<thead>
<tr>
<th>SRC → TGT</th>
<th>MRC $\rightarrow$ W NUT172</th>
<th>MRC $\rightarrow$ Rest.4</th>
<th>MRC $\rightarrow$ Movie2</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Onto. $\rightarrow$ WNUT17</td>
<td>43.1</td>
<td>46.8</td>
<td>44.2</td>
<td>46.9</td>
</tr>
<tr>
<td>Onto. $\rightarrow$ Rest.</td>
<td>1.6</td>
<td>5.0</td>
<td>2.7</td>
<td>11.0</td>
</tr>
<tr>
<td>Onto. $\rightarrow$ Movie</td>
<td>25.0</td>
<td>26.7</td>
<td>26.7</td>
<td>27.8</td>
</tr>
<tr>
<td>Average</td>
<td>23.3</td>
<td>26.2</td>
<td>24.5</td>
<td>28.6</td>
</tr>
</tbody>
</table>

Table 8: $F_1$ scores on NER cross-domain transfer, where models trained on source-domain training data (SRC) are evaluated on target-domain test sets (TGT).

Figure 4: The distribution of similarity score between categories and their corresponding positive/negative spans on Ontonote5 test set.

the MRC model exploits span semantics for prediction. The results are presented in Table 8. PeerDA can significantly exceed MRC on all three transfer pairs. On average, PeerDA achieves 2.9 and 4.1 $F_1$ gains over base-size MRC and large-size MRC respectively. These results verify our postulation that modeling the PR relation allows models to weigh more on the semantics for making predictions, and thus mitigates the over-fitting issue.

Semantic Distance: To gain a deeper understanding of the way in which PeerDA enhances model performance, we consider the span score (Eq. 7) as a measure of semantic similarity between a query and a span. In this context, we can create queries for all categories and visualize the similarity distribution between the categories and their corresponding positive and negative spans on Ontonote5 test set. As shown in Figure 4, we can observe that the use of PeerDA leads to an increased semantic similarity between spans and their corresponding categories, resulting in higher confidence in the prediction of correct spans. Furthermore, PeerDA has been shown to also create a larger similarity gap between positive and negative spans, facilitating their distinction.

Low-resource Evaluation: We simulate low-resource scenarios by randomly selecting 10%, 30%, 50%, and 100% of the training data for training SpanID models and show the comparison results between PeerDA and MRC on four SpanID tasks in Figure 3. As can be seen, our PeerDA further enhances the MRC model in all sizes of training data and the overall trends are consistent across the above four tasks. When training PeerDA with 50% of the training data, it can reach or even exceed the performance of MRC trained on the full training set. These results demonstrate the effectiveness of our PeerDA in low-resource scenarios.

7 Conclusions

In this paper, we propose a novel PeerDA approach for SpanID tasks to augment training data from the perspective of capturing the PR relation. PeerDA has two unique advantages: (1) It is capable to leverage abundant but previously unused PR relation as additional training data. (2) It alleviates the over-fitting issue of MRC models by pushing the models to weigh more on semantics. We conduct extensive experiments to verify the effectiveness of PeerDA. Further in-depth analyses demonstrate that the improvement of PeerDA comes from a better semantic understanding capability.
Limitations

In this section, we discuss the limitations of this work as follows:

- PeerDA leverages labeled spans in the existing training set to conduct data augmentation. This means that PeerDA improves the semantics learning of existing labeled spans, but is ineffective to classify other spans outside the training set. Therefore, it would be beneficial to engage outer source knowledge (e.g. Wikipedia), where a variety of important entities and text spans can also be better learned with our PeerDA approach.

- Since PeerDA is designed in the MRC formulation on top of the encoder-only Pre-trained Language Models (PLMs) (Devlin et al., 2019; Liu et al., 2019), it is not comparable with other methods built on encoder-decoder PLMs (Yan et al., 2021b; Chen et al., 2022; Zhang et al., 2021; Yan et al., 2021a). It would be of great value to try PeerDA on encoder-decoder PLMs such as BART (Lewis et al., 2020) and T5 (Raffel et al., 2020), to see whether PeerDA is a general approach regardless of model architecture.

- As shown in Table 12, although PeerDA can significantly alleviate the Missing Predictions, the most prevailing error in the MRC model, PeerDA also introduces some new errors, i.e. Multiple labels and Incorrect Label. It should be noted that those problematic spans are usually observed in different span sets, where they would learn different category semantics from their peers. Therefore, we speculate that those spans tend to leverage the learned category semantics more than their context information to determine their categories. We hope such finding can shed light on future research to further improve PeerDA.

References


A Appendix

A.1 Task Overview

We conduct experiments on four SpanID tasks with diverse domains, including Named Entity Recognition (NER), Aspect Based Sentiment Analysis (ABSA), Contract Clause Extraction (CCE) and Span Based Propaganda Detection (SBPD), to show the overall effectiveness of our PeerDA. The dataset statistics are summarized in Table 1.

NER is a traditional SpanID task, where spans denote the named entities in the input text and category labels denote their associated entity types. We evaluate five datasets from four domains:

- OntoNotes5 (Pradhan et al., 2013) is a large-scale mixed domain NER dataset covering News, Blog and Dialogue. To make a fair comparison in the robustness experiments in Sec. A.4, we use the datasets from Lin et al. (2021), which only add adversarial attack to the 11 entity types, while leaving out 7 numerical types.

- WNUT17 (Derczynski et al., 2017) is a benchmark NER dataset in social media domain. For fair comparison, we follow the data preprocessing protocols in Nie et al. (2020).

- Movie (Liu et al., 2013b) is a movie domain dataset containing movie queries, where long spans are annotated such as a movie’s origin or plot. We use the defaulted data split strategy into train, dev and test sets.

- Restaurant (Liu et al., 2013a) contains queries in restaurant domain. Similar to Movie, we use the defaulted data split strategy.

- Weibo (Peng and Dredze, 2015) is a Chinese benchmark NER dataset in social media domain. We exactly follow the official data split strategy into train, dev and test sets.

ABSA (Li et al., 2019b; Chen and Qian, 2020) is a fine-grained sentiment analysis task centering at aspect terms. We explore two ABSA sub-tasks:

- Aspect Term Extraction (ATE) is to extract aspect terms, where there is only one query asking if there are any aspect terms in the input text.

- Unified Aspect Based Sentiment Analysis (UABSA) is to jointly extract aspect terms and predict their sentiment polarities. We formulate it as a SpanID task by treating the sentiment polarities, namely, positive, negative, and neutral, as three category labels, and aspect terms as spans.

We evaluate the two sub-tasks on two datasets, including the laptop domain dataset Lap14 and restaurant domain dataset Rest14 from SemEval Shared tasks (Pontiki et al., 2014). We use the processed data from Zhang et al. (2021).

CCE is a legal NLP task to detect and classify contract clauses into relevant clause types, such as "Governing Law" and "Uncapped Liability". The goal of CCE is to reduce the labor of legal professionals in reviewing contracts of dozens or hundreds of pages long. CCE is also a kind of SpanID task where spans are those contract clauses that warrant review or analysis and labels are predefined clause types. We conduct experiments on CCE using CUAD (Hendrycks et al., 2021), where they annotate contracts from Electronic Data Gathering, Analysis and Retrieval (EDGAR) with 41 clause types. We follow Hendrycks et al. (2021) to split the contracts into segments within the length limitation of pretrained language models and treat each individual segment as one example. We also follow their data split strategy.

SBPD (Da San Martino et al., 2019) is a typical SpanID task that aims to detect both the text fragment (i.e. spans) where a persuasion technique is being used as well as its technique type (i.e. category labels). We use the News20 and Social21 from two SemEval shared tasks (Da San Martino et al., 2020; Dimitrov et al., 2021) and follow the official data split strategy. Note that News20 does not provide the golden label for the test set. Therefore, we evaluate News20 on the dev set.

A.2 Implementations

We use Huggingface’s implementations of BERT and RoBERTa (Wolf et al., 2020) 3. The hyper-parameters can be found in Table 9. We use Tesla V100 GPU cards for conducting all the experiments. We follow the default learning rate schedule and dropout settings used in BERT. We use AdamW (Loshchilov and Hutter, 2019) as our optimizer. The margin term \( M \) is set to 0 for NER and ABSA, and 1 for SBPD. The balance rate \( \alpha \) is set to 0.1.

3Chinese RoBERTa is from https://github.com/ymcui/Chinese-BERT-wwm.
Dataset & OntoNote5 & WNUT17 & Movie & Restaurant & Weibo & Lap14 & Rest14 & CUAD & News20 & Social21  
Query Length & 32 & 32 & 64 & 64 & 64 & 24 & 24 & 256 & 80 & 80  
Batch Size & 32 & 32 & 32 & 32 & 8 & 16 & 16 & 16 & 16 & 16  
$\lambda$ & 1 & 1 & 1 & 1 & 1 & -0.5 & 0.5 & 1 & 

Table 9: Hyper-parameters settings.

Figure 5: Performance in terms of different DA rate $\lambda$. We vary $\lambda$ to get different volumes of PR-based training data.

### A.3 Effect of DA Rate

We vary the DA rate $\lambda$ to investigate how the volume of PR-based training data affect the SpanID models performance.

Figure 5 shows the effect of different $\lambda$ in four SpanID tasks. PeerDA mostly improves the MRC in all different trials of $\lambda$ and we suggest that some parameter tuning for $\lambda$ is beneficial to obtain optimal results.

Another observation is that too large $\lambda$ would do harm to the performance. Especially on CCE, due to the skewed distribution and a large number of categories, PeerDA can produce a huge size of PR-based training data. We speculate that too much PR-based training data would affect the learning of BIL-based training data and thus affect the model’s ability to solve a SpanID task, causing the optimal $\lambda$ to be a negative value. In addition, too much PR-based training data would also increase the training cost. As a result, we should maintain an appropriate ratio of BIL-based and PR-based training data to keep a reasonable performance on SpanID tasks.

### A.4 Robustness:

To verify the advantage of PeerDA against the adversarial attack, we conduct robustness experiments using the adversarial dev set of OntoNotes5 (Lin et al., 2021) on NER and adversarial test set of Lap14 (Xing et al., 2020) on UABSA. Table 10 shows the performance on the original and the adversarial sets. On OntoNotes5 full adversarial set, PeerDA improves the robustness of the model compared to MRC but slightly degrades compared to Tagging. To investigate why this happens, we evaluate each type of adversarial attack independently, including entity attack that replaces entities to other entities not presented in the training set and context attack that replaces the context of entities. It shows that PeerDA does not work well on entity attack because we only use entities in the training set to conduct data augmentation, which is intrinsically ineffective to this adversarial attack. This motivates us to engage outer source knowledge (e.g. Wikipedia) into our PeerDA approach in future work. On Lap14, PeerDA significantly improves Tagging and MRC by 5.6 and 3.2 F1 on the adversarial set respectively.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tagging</td>
<td>89.8</td>
<td>56.6</td>
<td>61.9</td>
<td>83.6</td>
<td>62.3</td>
</tr>
<tr>
<td>MRC</td>
<td>90.0</td>
<td>55.3</td>
<td>61.3</td>
<td>83.3</td>
<td>63.2</td>
</tr>
<tr>
<td>PeerDA</td>
<td>90.1</td>
<td>55.9</td>
<td>61.0</td>
<td>84.1</td>
<td>65.9</td>
</tr>
</tbody>
</table>

Table 10: Robustness experiments against adversarial attacks. The results are reported on both original (Ori.) sets and the adversarial (Adv.) sets.
Table 11: Performance on peer-driven DA approaches.

<table>
<thead>
<tr>
<th>Methods</th>
<th>OntoNotes5</th>
<th>Lap14</th>
<th>CUAD</th>
<th>Social21</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRC+MenReplace</td>
<td>91.1</td>
<td>63.7</td>
<td>45.2</td>
<td>50.8</td>
</tr>
<tr>
<td>PeerDA</td>
<td>92.4</td>
<td>65.9</td>
<td>52.3</td>
<td>58.1</td>
</tr>
</tbody>
</table>

A.5 Peer-driven DA:
We compare PeerDA with Mention Replacement (MenReplace) (Dai and Adel, 2020), another Peer-driven DA approach randomly replaces a span mention in the context with another mention of the same category in the training set. The results of four SpanID tasks are presented in Table 11. PeerDA exhibits better performance than MenReplace on all four tasks. In addition, MenReplace would easily break the text coherence as a result of putting span mentions into the incompatible context, while PeerDA can do a more natural augmentation without harming the context.

A.6 Error Analysis:
In order to know the typical failure of PeerDA, we randomly sample 100 error cases from Ontonotes5 test set for analysis. As shown in Table 12, there are four major groups:

- **Multiple Labels**: PeerDA would assign multiple labels to the same detected span. And in most cases (35/41), this error occurs among similar categories, such as LOC, GPE, and ORG.
- **Incorrect Label**: Although spans are correctly detected, PeerDA assigns them the wrong categories. Note that MRC even cannot detect many of those spans (23/37). As a result, PeerDA significantly improves the model’s capability to detect spans, but still faces challenges in category classification.
- **Missing Prediction**: Compared to MRC, PeerDA tends to predict more spans. Therefore it alleviates the missing prediction issue that MRC mostly suffers.
- **Other Errors**: There are several other errors, such as the incorrect span boundary caused by articles or nested entities.
Multiple Labels

- **I'm in Atlanta.**
  - **Gold:** ("Atlanta", GPE)
  - **PeerDA:** ("Atlanta", GPE); ("Atlanta", LOC) (41%)
  - **MRC:** ("Atlanta", GPE) ("Atlanta", LOC) (3%)

Incorrect Label

- **Why did it take us to get Sixty Minutes to do basic reporting to verify facts?**
  - **Gold:** ("Sixty Minutes", ORG)
  - **PeerDA:** ("Sixty Minutes", WORK_OF_ART) (37%)
  - **MRC:** ("Sixty Minutes", WORK_OF_ART) (20%)

Missing Prediction

- **Coming to a retailer near you, PlayStation pandemonium.**
  - **Gold:** ("PlayStation", PRODUCT)
  - **PeerDA:** ∅ (19%)
  - **MRC:** ∅ (74%)

Other Errors

- **I was guarded uh by the British Royal Marines actually because unfortunately they’ve had now um roadside bombs down there not suicide bombs.**
  - **Gold:** ("the British Royal Marines", ORG)
  - **PeerDA:** ("Royal Marines", ORG) (3%)
  - **MRC:** ("Royal Marines", ORG) (3%)

Table 12: Error analysis of base-sized PeerDA and MRC models on Ontonotes5 test set. We randomly select 100 examples from the test set and compare the predictions and error percentage of the two models.