Document-Level Event Argument Extraction by Leveraging Redundant Information and Closed Boundary Loss

Anonymous ACL submission

Abstract

In document-level event argument extraction, an argument is likely to appear multiple times in different expressions in the document. The redundancy of arguments underlying multiple sentences is beneficial but is often overlooked. In addition, in event argument extraction, the majority entities are regarded as class “others", i.e. universum class, which is composed of heterogeneous entities without typical common features. Classifiers trained by cross entropy loss could easily misclassify universum class because of their open decision boundary. In this paper, to make use of redundant information underlying a document, we build an entity coreference graph with graph2token module to produce comprehensive and coreference-aware representation for every entity, and then build an entity summary graph to merge the multiple extraction results. To better classify universum class, we propose a new loss function to build classifiers with closed boundaries. Experimental results show that our model outperforms the previous state-of-the-art models by 3.35% in F1-score.

1 Introduction

Event argument extraction (EAE) is a crucial subtask of event extraction (EE), aiming to identify the arguments of a given event and recognize the specific roles they play. Previous works are mostly focused on sentence level EE (Liao and Grishman, 2010; Nguyen et al., 2016; Liu et al., 2018; Yang et al., 2019b; Du and Cardie, 2020b; Wei et al., 2021; Wang et al., 2021; Lyu et al., 2021). However, events are often described in the form of documents in real world. Document-level event extraction has received considerations in recent years.

Research on document-level event extraction has been focused on tackling challenges such as arguments-scattering and multiple-events (Zheng et al., 2019; Du and Cardie, 2020a; Du et al., 2021; Lou et al., 2021; Li et al., 2021; Huang and Peng, 2021; Xu et al., 2021; Yang et al., 2021; Ahmad et al., 2021). The benefit of redundant information in a document is largely neglected. We believe that the redundant event information in a document can be used to improve event extraction, as illustrated in the example in Figure 1. The upper part of Figure 1 shows six simplified sentences selected form a document in the MUC-4 dataset. All entities marked in blue are the same entity that appears with different expressions in different sentences. For ease of description, we call it entity S. We can observe from Figure 1 that: 1) The argument information in the document is redundant since entity S appears in the article multiple times as an argument. We can successfully extract the argument by correctly recognizing any of these occurrences. This property can be potentially used to improve the robustness of the model. 2) The difficulty of extracting the entity S as an argument in its different occurrences is different. Extracting entity S

No.	Sentence	Entity label	Difficulty
s1	The killers, approximately 30 men in uniform, arrived before 0230.	1	★
s2	Soldiers with their faces painted black arrived in Cayara last Saturday. They broke down doors, looted stores, and burned several houses.	1	★★★
s3	The murder was carried out by soldiers.	1	★
s4	The house was surrounded by soldiers.	0	-
s5	The house was searched by the soldiers 2 days before the crime.	0	-
s6	How can men in uniform be in a militarized area?	0	-
...	......	...	...

⬇

Argument role	Entity	Entity label	Summative label
Perpetrator individual	men in uniform	1	1
	soldiers with their faces painted black	1	0
	soldiers	1	0
Perpetrator organization	amed forces	1	1
	military	1	0
Physical target	houses	1	1
	vehicle	1	1

Figure 1: An example of redundant information in the document-level event argument extraction.

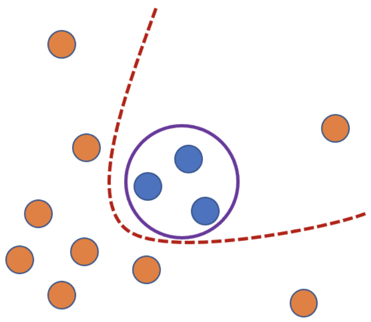

Figure 2: A simplified illustration of closed boundary loss. Blue dots represent target samples, orange dots represent universum samples. The red dotted line represents cross entropy loss, the purple solid line represents proposed closed boundary loss.

in sentence 1 and sentence 3 is much easier than extracting it from sentence 2. Hence, by utilizing the redundant information of the document, we can extract arguments from relatively simple positions and reduce the difficulty of the task. 3) An entity may appear multiple times in the document, directly averaging them as the entity's feature representation (Xu et al., 2021) may introduce noise, such as the feature of entity S in s4, s5, s6. On the other hand, the redundant argument information results in redundant extraction results, as shown in the bottom table in Figure 1. The three entities extracted as perpetrator individual need to be merged into one. However, the extracted physical target "houses" and "vehicles" are different entities and should not be merged. Therefore, the use of redundant information underlying a document is not straightforward, a sophisticated algorithm for merging multiple extraction results is needed.

Extraction of arguments can be solved as an entity classification problem by treating entities as argument candidates (Zheng et al., 2019; Xu et al., 2021; Yang et al., 2021). In document-level event argument extraction, only a subset of the entities in a document are arguments, while the majority entities are regarded as class "others"or "neither"(neither of the target classes). This kind of data was first studied by Vapnik (2006) under the name Universum. The universum data are usually very diverse and do not have typical common features. In addition, universum data is much more than the target class data, exhibiting severe class imbalance problem. Figure 2 demonstrates a simplified distribution of data samples in document-level event argument extraction. The blue dots represent argument entities, the orange dots represent a large number of universum entities. Since the samples in universum class do not have typical common features, they tend to scatter in the feature space. This characteristic of the universum data is largely overlooked in the information extraction community. Universum data is simply considered as another class "others", without any special consideration in the classifier design. Cross entropy loss is usually employed in classifier training (Zheng et al., 2019; Huang and Peng, 2021; Xu et al., 2021; Yang et al., 2021). However, classifiers trained by cross entropy loss have open decision boundary, and hence some universum samples, such as the orange dot on the upper right of the figure, could be easily misclassified. We think a classifier with closed decision boundary could better deal with the universum class in document-level event argument extraction, as illustrated by the purple line in Figure 2.

The contribution of this work is three-fold. Firstly, it is the pioneering work to leverage redundant information in documents for event extraction. We propose the entity coreference graph with graph2token module and entity summary graph to leverage the redundant information. Experimental results show that redundant information helps improve recall significantly. Secondly, we analyse the issue of universum data in document-level event argument extraction and the problem of classifiers trained by cross entropy loss, and propose a closed boundary loss to address the problems. Finally, our model consistently outperforms latest baseline models in F1-score and achieves the state-of-the-art performance. Compared to three baseline models, our proposed model improve the absolute F1-score by 3.35%, 5.27%, and 6.45%, respectively.

2 Related Work

2.1 Event Argument Extraction

Most previous event argument extraction models make predictions at sentence-level (Nguyen et al., 2016; Liu et al., 2018; Yang et al., 2019b; Du and Cardie, 2020b; Wei et al., 2021; Wang et al., 2021; Dutta et al., 2021). Considering that real world events are often distributed across sentences, document-level event extraction has attracted more attentions recently. Zheng et al. (2019) propose the Doc2EDAG model to overcome the argument scattering problem. Du and Cardie (2020a) first argue the importance of document-level extraction and adopt sequence model on document-level event extraction. Lou et al. (2021) investigate a novel bidirectional decoder to overcome the long-range forgetting problem. Li et al. (2021) formulate document-level event extraction model as conditional generation based on templates. Huang and Peng (2021) attach importance to event coreference and entity coreference in document-level event extraction tasks. Xu et al. (2021) build a heterogeneous graph with the Tracker module to deal with problems of event scattering and multiple events. Yang et al. (2021) adopt parallel prediction networks to extract events parallelly from document-level representations. However, none of these works pay attention to the characteristic of information redundancy in the document, which we believe is an unique and beneficial property for document-level event argument extraction. In addition, to our knowledge, closed boundary classification has never been adopted in event extraction. Classification-based event augment extraction works (Huang and Peng, 2021; Xu et al., 2021; Yang et al., 2021) all employ cross entropy loss for classifier training, without considering the characteristics of universum class: scattered distribution in the feature space due to heterogeneity and diversity of the samples in this class.

2.2 Closed Boundary Loss

We found that a classifier trained by cross entropy could easily misclassify entities in the class "others", i.e. universum class. We found the root cause of the problem is the open decision boundary of the classifier. To address this problem, we propose a novel closed boundary loss for classifier training.

Research works in universum usually employ additional unlabeled universum data to provide prior knowledge for the task, such as universum support vector machine (SVM) (Weston et al., 2006; Qi et al., 2012; Richhariya and Tanveer, 2020), and semi-supervised learning (Liu et al., 2015; Xiao et al., 2021). However, the SVM-based methods above are developed for structured data and are hardly to integrate with deep neural network-based representation learning to form an end-to-end training procedure for natural language processing tasks. One possible solution is to use a deep neural network to learn representations first, and then feed the representations learned to the universum SVM classifiers. But the disadvantage of this two-step procedure is that the classification result cannot be back-propagated to representation learning. It is desired that the closed boundary classifier could be integrated with deep neural network-based representation learning to form end-to-end training for optimal performance.

Closed boundary classification methods are also developed in anomaly detection and open set recognition, such as deep one-class learning (Ruff et al., 2018; Defard et al., 2021), auto-encoder based anomaly detection (Ionescu et al., 2019), OpenMax layer for open set recognition (Bendale and Boult, 2016). However, these methods cannot use the information in outlier samples due to task setting.

Closed boundary classifier works best in feature space with compact class distribution. In the literature, some loss functions have been proposed to generate such feature space such as Deep SVDD (Ruff et al., 2018), contrastive loss (Hadsell et al., 2006) and ii-loss (Hassen and Chan, 2020). However, Deep SVDD only minimizes the intra-class distance and cannot maximize the inter-class distance. Contrastive loss and ii-loss need to be combined with cross entropy loss to classify samples. But cross entropy loss generates open decision boundaries for the classifier.

In this paper, we propose a new loss function which could train a classifier with closed decision boundary. In addition, it can be directly integrated with representation learning layers in a neural network to form an end-to-end training procedure to produce a feature space with minimum intra-class difference and maximum inter-class difference, which in turn leads to improved performance.

3 Method

As shown in the Figure 3, our model consists of four main components: context encoding module (Sec 3.1), entity coreference graph (Sec 3.2), closed boundary loss (Sec. 3.3), and entity summary graph (Sec. 3.4), which are illustrated in this section.

3.1 Context Encoding

Given the input document, we apply a Bi-LSTM to obtain token representations of the document: $\mathrm{D} = \{\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1}\} \in \mathbb{R}^{n\times l}$ where n is the document length, and l is the the hidden state dimension. We construct entity representation and sentence representation from token representations:

$$\mathbf{e}_i = \left(\mathbf{e}^{(i)}_{\text{memory}}; \mathbf{e}^{(i)}_{\text{rule}}\right) \quad (1)$$

$$\mathbf{s}_i = \left(\mathbf{s}^{(i)}_{\text{memory}}; \mathbf{s}^{(i)}_{\text{rule}}\right) \quad (2)$$

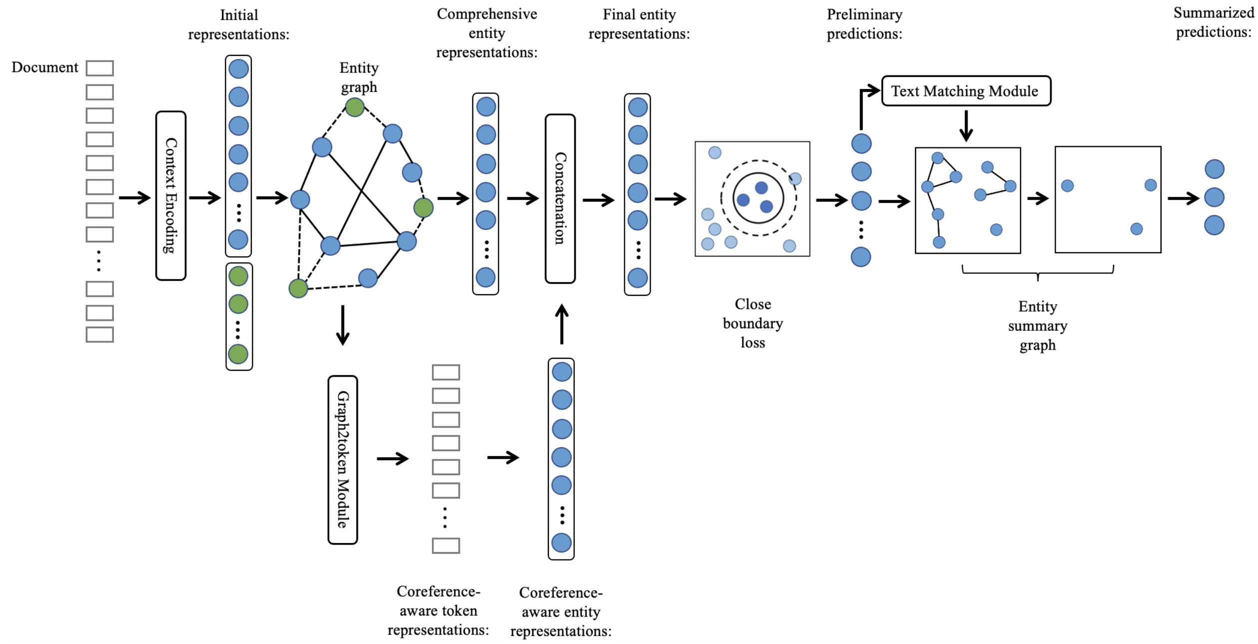

Figure 3: The overall model structure. Blue dots represent entity nodes, green dots represent sentence nodes.

$$\mathbf{e}^{(i)}_{\text{memory}} = \left(\mathrm{D}\left[ent^{(i)}_{\text{start}}[l:]\right]; \mathrm{D}\left[ent^{(i)}_{\text{end}}[:l]\right]\right)$$

$$\mathbf{e}^{(i)}_{\text{rule}} = \left(\mathrm{D}\left[ent^{(i)}_{\text{start}}[:l]\right]; \mathrm{D}\left[ent^{(i)}_{\text{end}}[l:]\right]\right)$$

$$\mathbf{s}^{(i)}_{\text{memory}} = \left(\mathrm{D}\left[sent^{(i)}_{\text{start}}[l:]\right]; \mathrm{D}\left[sent^{(i)}_{\text{end}}[:l]\right]\right)$$

$$\mathbf{s}^{(i)}_{\text{rule}} = \left(\mathrm{D}\left[sent^{(i)}_{\text{start}}[:l]\right]; \mathrm{D}\left[sent^{(i)}_{\text{end}}[l:]\right]\right)$$

where D is the output of the Bi-LSTM encoding layer, $ent^{(i)}_{\text{start}}$, $ent^{(i)}_{\text{end}}$, $sent^{(i)}_{\text{start}}$ and $sent^{(i)}_{\text{end}}$ are the start and end position of i-th entity and i-th sentence, respectively, and [;] denotes the concatenation operation. $\mathbf{e}^{(i)}_{\text{memory}}$ and $\mathbf{s}^{(i)}_{\text{memory}}$ mainly contain the information inside the entity and sentence. $\mathbf{e}^{(i)}_{\text{rule}}$ and $\mathbf{s}^{(i)}_{\text{rule}}$ mainly contain the context information outside the entity and sentence. We separate the memory representation and rule representation because they correspond to memory-based and rule-based learning process of human.

3.2 Entity Coreference Graph

Leveraging redundant information of document is not straightforward to classify every entity in the document. On the one hand, better entity representation is needed. Therefore, we construct entity coreference graph with graph2token module to produce comprehensive and coreference-aware representation for every entity.

The entity coreference graph is inspired by the observation of coreference's role in document un-

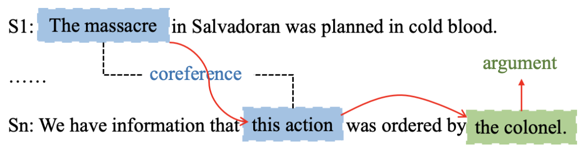

Figure 4: An example of coreference in document and its impact on entity understanding and document-level event argument extraction

derstanding. Firstly, for the repeatedly referred entity (coreference entity), the understanding to this entity is constantly enriched or enhanced by each reference. For the example illustrated in Figure 4, "the massacre" and "this action" are two different mentions of the same entity. The understanding of this entity is enriched by combining the location of the massacre mentioned in the first sentence and the commander of the massacre mentioned in the second sentence. Secondly, for other entities locating in the context of the coreference entity, their meanings are clearer by recognizing the connotation of coreference entity. For example, "the colonel" cannot be recognized as an argument unless the model understands that "this action" refers to "the massacre". Research works in event extraction (Xu et al., 2021; Luan et al., 2019; Qian et al., 2019) take the first observation into consideration but neglect the second one. Specifically, previous works in event extraction use graph structure to merge information in different mentions of the same entity. However, the adoption of merely such a graph structure cannot feed back the fused information to the context of coreference entities. In this sense, for the representation of "the colonel", its context information still excludes "the massacre". Therefore, we adopt a graph2token module to feed back the comprehensive entity information obtained through graph structure to tokens, and then rebuild entity representations that are both comprehensive and coreference-aware.

Graph Construction. There are two types of nodes in the entity graph: entity nodes and sentence nodes. Entities are recognized from document following Fisher and Vlachos (2019). Entity nodes and sentence nodes are denoted as $\mathrm{E} = \{\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_p\}$, and $\mathrm{S} = \{\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_q\}$, respectively.

There are two types of edges in the entity graph: 1) entity-entity edge is created according to the coreference relationship. We use SpanBERT (Joshi et al., 2020) to implement coreference resolution on documents during preprocessing. 2) entity-sentence edge is the connection between the entity node and the sentence node where it is located.

Graph Propagation. After the graph is constructed, Graph Attention Network (Veličković et al., 2017) is applied to propagate information between connected nodes. Assuming that graph nodes are denoted by $\mathrm{H} = \{\mathrm{E}, \mathrm{S}\} = \{\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{p+q}\} \in \mathbb{R}^{(p+q)\times 2l}$, the information that a node receives from its neighbors is formulated as:

$$\mathbf{h}_i' = \mathrm{RELU}\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\mathbf{W}\mathbf{h}_j\right) \tag{3}$$

$$\alpha_{ij} = \frac{\exp\left(L\left(\mathbf{W}_{e_{ij}}[\mathbf{W}\mathbf{h}_i;\mathbf{W}\mathbf{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(L\left(\mathbf{W}_{e_i}[\mathbf{W}\mathbf{h}_i;\mathbf{W}\mathbf{h}_k]\right)\right)} \tag{4}$$

where $\mathbf{h}_i'$ is the neighbor information of i-th node, $\mathbf{h}_j$ is the representation of j-th node, $\mathbf{W}$, $\mathbf{W}_{e_i}$ are weight matrixes, $\mathcal{N}_i$ denotes the set of neighbors of node i, and $L(\cdot)$ is the LeakyReLU function.

The representation of i-th node $\mathbf{h}_i$ and its neighbor information $\mathbf{h}_i'$ is fused by the gated mechanism:

$$\beta_i = \sigma\left(f\left(\mathbf{h}_i;\mathbf{h}_i'\right)\right) \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function, $f(\cdot)$ denotes the linear transformation. The fused representation of i-th node $\mathbf{h}_i''$ is obtained as:

$$\mathbf{h}_i'' = \beta_i \odot \mathbf{h}_i + (1-\beta_i)\odot \mathbf{h}_i' \tag{6}$$

where $\odot$ stands for element-wise multiplication. Through propagating and fusing information of coreference entities and sentence, a comprehensive representation of entity is obtained.

Graph2token. To address the second insight we put forward in this section, we adopt graph2token module to feed back the information behind coreference entities to tokens.

We concatenate the token representation $\mathbf{d}_i$ with the entity representation $\mathbf{h}_j''$ in which it is located, and feed it to a LSTM layer to generate coreference-aware token representation $\mathbf{d}_i'$:

$$\mathbf{d}_i' = \mathrm{LSTM}(\mathbf{d}_i;\mathbf{h}_j'') \tag{7}$$

Then, we build coreference-aware entity representations from updated token representations.

$$\mathbf{e}_{\text{rule}}^{(i)\,\prime} = \left(D'\left[ent_{\text{start}}^{(i)}[:l]\right]; D'\left[ent_{\text{end}}^{(i)}[l:]\right]\right)$$

where $\mathrm{D}' = \left\{\mathbf{d}_0', \mathbf{d}_1', \ldots, \mathbf{d}_{n-1}'\right\}$. Finally, a comprehensive and coreference-aware entity representation $\mathrm{E}\prime = \{\mathbf{e}_0\prime, \mathbf{e}_1\prime, \ldots, \mathbf{e}_p\prime\}$ is obtained by concatenation:

$$\mathbf{e}_i\prime = \left(\mathbf{h}_i''; \mathbf{e}_{\text{rule}}^{(i)\,\prime}\right) \tag{8}$$

3.3 Closed Boundary Loss

We have analyzed that classifiers trained by cross entropy loss have open decision boundaries and could easily misclassify the universum class. To address this problem, we propose a novel loss function that could be used to train classifiers with closed decision boundaries.

Comprehensive and coreference-aware entity representations $\mathrm{E}\prime = \{\mathbf{e}_0\prime, \mathbf{e}_1\prime, \ldots, \mathbf{e}_p\prime\}$ are obtained in the last section. We treat entities as argument candidates and classify entities by classifiers trained by our proposed closed boundary loss:

$$\begin{aligned} \mathrm{L_{CB}} = {} & R^2 + \frac{1}{n}\sum_{i=1}^{n}\max\left(0, \|\mathbf{e}_i\prime - \mathbf{c}\|^2 - R^2\right) \\ & + \frac{1}{m}\sum_{i=1}^{m}\max\left(0, (1+\mu)R^2 - \|\mathbf{e}_i\prime - \mathbf{c}\|^2\right) \end{aligned} \tag{9}$$

The intention of the closed boundary loss is to include samples of each target class using a hypersphere characterized by center $\mathbf{c}$ and radius R in the feature space, and locate universum samples outside the hypersphere. Due to the heterogeneous nature of universum samples, we allow them to scatter outside the hypersphere and do not require them to be aggregated like cross entropy loss.

The goal of the first term R^2 is to minimize the volume of the hypersphere while the second term aims to enclose target class samples by the hypersphere, where the center $\mathbf{c}$ is initialized as the mean of target samples in the feature space. If the Euclidean distance between the sample $\mathbf{h}_i''$ and the center $\mathbf{c}$ exceeds the radius, it will lead to a penalty in the loss function. The third term aims to make universum samples to be located outside the hypersphere. Parameter μ is introduced to adjust the gap between the closed boundary hypersphere and universum samples.

Unlike contrastive loss and ii-loss that cannot be directly used for classifying samples in the test set and need to be combined with cross entropy loss, our proposed closed boundary loss can be easily adopted for classification by following calculation:

$$g(\mathbf{e}_i\prime) = \begin{cases} 1 & \|\mathbf{e}_i\prime - \mathbf{c}\|^2 - R^2 < 0 \\ 0 & \|\mathbf{e}_i\prime - \mathbf{c}\|^2 - R^2 > 0 \end{cases}$$

3.4 Entity Summary Graph

To make full use of the redundant argument information, we classify every entity in the document. For the same argument, we may obtain multiple preliminary extraction results. The advantage is the robustness because the correct argument is more likely to be extracted from from relatively simple positions. The challenge is how to merge the multiple extraction results. To address the challenge, we propose an entity summary graph.

Text Matching Module. We notice that most redundant expressions of the same entity are either character-level spelling similar or word-level semantics similar. In some cases, special domain knowledge is needed to determine if two expressions are actually the same. For example, “Army of National Liberation” and "ELN" are refereed to the same entity. Therefore, we adopt text matching model with both character embedding and word embedding to evaluate the spelling similarity and semantics similarity of extracted arguments. We also construct text matching dataset from ground truth labels of training set of our event extraction dataset to make the model learn necessary domain knowledge.

We build the text matching module (TMM) by adopting the structure of RE2 (Yang et al., 2019a) and adding character embedding to the RE2 model to enhance model’s capability of recognizing spelling similarity. We denote the initially predicted arguments as $\mathrm{A} = \{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{k-1}\}$. Then, we feed these entities into text matching module to produce the matching score for each pair of arguments.

$$\mathbf{M}_{ij} = \mathrm{TMM}\left(\mathbf{a}_i, \mathbf{a}_j\right) \tag{10}$$

where $\mathbf{M}$ is the matching score matrix, which contains text matching score of every pair of entities from A. $\mathbf{M} = [\mathbf{M}_{ij}], i, j = 1, 2, \ldots, k$.

Entity Summary Graph. The graph node is composed of preliminary predicted entities A. The i-th node and j-th node is connected if $\mathbf{M}_{ij} > s$, where s is a boundary score. The weight of each edge is the text matching score $\mathbf{M}_{ij}$ of two entity nodes at the ends of the edge.

The constructed entity summary graph is mostly disconnected because there usually exists multiple argument clusters in a document. The argument cluster refers to a set of different expressions of the same argument, such as ["the armed forces", "military"]. Thus, an entity summary graph consists of several connected subgraphs as shown in figure 3. Each subgraph corresponds to an argument cluster. We denote the entity summary graph G and its subgraphs as $\mathrm{G} = \left\{\mathrm{G}_{sub}^{(1)}, \mathrm{G}_{sub}^{(2)}, \ldots, \mathrm{G}_{sub}^{(u)}\right\}$. The final predicted arguments are summarized by selecting an entity node with the largest sum of weights (LSW) from each subgraph.

$$\mathrm{A}' = \left\{\mathbf{a}_0', \mathbf{a}_1', \ldots, \mathbf{a}_{v-1}'\right\}, \quad \mathbf{a}_i' = \mathrm{LSW}\left(\mathrm{G}_{sub}^{(i)}\right)$$

4 Experiments

4.1 Dataset

Our model is evaluated on the MUC-4 dataset (McLean, 1992). The dataset is composed of 1,700 documents, each containing an average of 400 tokens and 7 paragraphs. We use 1300 documents for training, 200 documents for testing, and 200 documents as development set following (Du and Cardie, 2020a). Five argument roles are extracted in the dataset: perpetrator individual, perpetrator organization, target, victim, and weapon.

4.2 Baselines and Evaluation Metric

In this work, we propose a document-level EAE model leveraging **R**edundant **I**nformation and **C**losed **B**oundary Loss (**RICB**). We compare our model with the following baseline models: **DYGIE++** (Wadden et al., 2019) incorporates local

	PerpInd	PerpOrg	Target	Victim	Weapon
GTT (Du et al., 2021)	65.48/39.86/49.55	66.04/42.68/51.85	55.05/44.12/48.98	76.32/61.05/**67.84**	61.82/56.67/59.13
NST (Du and Cardie, 2020a)	48.39/32.61/38.96	60.00/43.90/50.70	54.96/52.94/53.93	62.50/63.16/62.83	61.67/61.67/61.67
DYGIE++ (Wadden et al., 2019)	59.49/34.06/43.32	56.00/34.15/42.42	53.49/50.74/52.08	60.00/66.32/63.00	57.14/53.33/55.17
RICB	50.76/49.62/**50.18**	50.00/63.75/**56.04**	65.63/63.64/**64.62**	64.86/50.52/56.80	63.49/65.57/**64.51**

Table 1: Performance comparison with baseline models for each argument role on MUC-4 dataset. Results for each column are displayed in the order of precision, recall, and F1 score.

Models	P	R	F1
GTT (Du et al., 2021)	**64.19**	47.36	54.50
NST (Du and Cardie, 2020a)	56.82	48.92	52.58
DYGIE++ (Wadden et al., 2019)	57.04	46.77	51.40
RICB	57.68	**58.03**	**57.85**

Table 2: Averaged EAE result on MUC-4 dataset. Precision, recall and F1-score are used for evaluation.

and global context to build a multi-task framework for named entity recognition, relation extraction, and event extraction. **NST** (Du and Cardie, 2020a) aggregates sentence representation and paragraph representation via gate mechanism and treats document-level EAE as a sequence tagging problem. **GTT** (Du et al., 2021) proposes a generative transformer based framework for document-level EAE.

We evaluate the performance of our model by **CEAF-TF** metric following (Du et al., 2021). The metric find the best alignment of predicted arguments and gold arguments. It penalize the system that do not merge multiple extraction results by setting a constraint that a gold argument can be aligned with at most one predicted argument. Precision (P), recall (R) and F1-score (F1) are used to evaluate the model's performance.

4.3 Overall Results

The per-role EAE results on MUC-4 dataset of our RICB model and baseline models are summarized in Table 1, and the micro-averaged performance is shown in Table 2. Table 2 shows that our model consistently outperforms latest baselines in F1-score and achieves the state-of-the-art (SOTA) performance. Specifically, the proposed model improve the absolute F1-score by 3.35%, 5.27%, and 6.45% compared to baseline models. Noticeably, our model achieves an over 9% improvement in recall. In terms of the per-role extraction performance of our model, it achieves the highest F1-score in four of five argument roles: perpetrator individual, perpetrator organization, target, and weapon. Specifically, the absolute F1-score is improved by 0.63%, 4.19%, 10.69%, and 2.84% in these argument roles.

4.4 Effect of Graph2token Module

Graph structure is used in EAE to produce comprehensive representation for coreference entities (Luan et al., 2019; Qian et al., 2019; Xu et al., 2021). In this work, we further adopt a graph2token module to feed back the comprehensive representation of coreference entities to their context tokens. The updated token representations can generate additional coreference-aware representations for entities near the coreference entity. For ablation study, we conduct experiment on without applying graph2token module, and compare per-role extraction results between with and without graph2token module in Table 3. We find that the experiment without graph2token module results in performance drop on every argument role. In addition, the recall is decreased by 0.38%, 4.92%, 6.06%, and 0.99% in four argument roles. This indicates that the model can recognize more arguments by providing argument candidates with additional coreference-aware representations.

4.5 Effect of Closed Boundary Loss

We find that classifier trained by cross entropy loss could easily misclassify entities in the universum class. We propose a closed boundary loss to address this issue. For ablation study, we conduct experiments of applying cross entropy loss for argument classification, and compare the performance with our model. The comparison of two loss functions is summarized in Table 3, which shows that in all argument roles, closed boundary loss consistently outperforms cross entropy in F1 score. We further notice that the precision of the model is improved in all argument roles at 0.76%, 1.43%,

	PerpInd	PerpOrg	Target	Victim	Weapon
Without graph2token	50.39/49.24/49.80	50.02/58.83/54.07	63.87/57.58/60.56	62.54/49.53/55.28	58.72/69.47/63.64
Cross entropy loss	50.00/50.34/50.17	48.57/63.75/55.14	62.04/64.39/63.19	49.55/58.95/53.85	55.13/70.49/61.87
String matching	48.80/45.86/47.28	45.30/66.25/53.81	65.71/63.44/64.56	59.49/49.47/54.02	58.57/67.21/62.60
RICB	50.76/49.62/**50.18**	50.00/63.75/**56.04**	65.63/63.64/**64.62**	64.86/50.52/**56.80**	63.49/65.57/**64.51**

Table 3: Ablation studies on graph2token module, closed boundary loss, and entity summary graph, respectively. The results in each column are displayed in the order of precision, recall, and F1 score.

3.59%, 15.31%, and 8.36% by using closed boundary loss. The improvement in precision indicates that the use of closed boundary result in a smaller number of universum samples that are misclassified as target samples.

4.6 Effect of Entity Summary Graph

To fully leverage the redundant argument information, we classify every entity in the document. For the same argument, we may obtain multiple preliminary extraction results. We propose the entity summary graph to merge the results. For ablation study, we conduct experiments on merging multiple extraction results based on string matching following Zheng et al. (2019); Xu et al. (2021). We compare the string matching performance with our proposed entity summary graph in Table 3. It shows that entity summary graph outperforms string matching method significantly in F1-score. Furthermore, the precision of model is improved in four of five argument roles by 1.96%, 4.70%, 5.37%, and 4.92% by using entity summary graph, and this verifies the effect of our proposed entity summary graph, i.e. merging multiple extraction results and reducing false positives accordingly.

4.7 Further Analysis

Firstly, it is effective to leverage redundant information of document for document-level EAE, which is not only reflected in the F1 score, but also in the significant improvement in recall. The micro-averaged recalls of baseline models are distributed between 46% to 49%, but our model reaches 58%. As we analyzed in the introduction, leveraging redundant argument information of document allows the model to extract the argument from any of its occurrences and relatively simple positions. Therefore, the difficulty of recognizing event arguments is reduced and the recall is improved accordingly.

Secondly, leveraging redundant information of document is not simply classifying every entity in the document. On the one hand, better entity representations need to be produced, on the other hand, multiple extraction results need to be merged. Therefore, we add graph2token module to entity coreference graph to generate comprehensive and coreference-aware entity representation, which improves the recall significantly. We also propose entity summary graph to merge multiple extraction results, which successfully improve the precision.

Finally, we propose a novel closed boundary loss to better deal with the universum class in our task. Its effectiveness is verified in ablation studies. We highlight two other potential benefits of closed boundary loss here. Firstly, since it generates closed decision boundary for classifiers, it may also be valid for dealing with unseen samples in the test set. However, this property is not evaluated in this work. In addition, our dataset is highly imbalanced because only a small number of entities are arguments. Weighted cross entropy loss is cumbersome to adjust the appropriate weights, however, the closed boundary loss does not need to adjust weights and works well with imbalanced dataset.

5 Conclusion and Future Works

In this work, we emphasize that the redundant information of document is beneficial but is often overlooked in document-level EAE. In addition, we find that classifiers trained by cross entropy loss are problematic in classifying the universum class. Specifically, we generate comprehensive and coreference-aware representation for every entity through entity coreference graph with graph2token module. In addition we propose an entity summary graph to merge the multiple extraction results of a same argument. Furthermore, we propose a novel closed boundary loss to deal with the universum class in classification. As a limitation, our proposed closed boundary loss is used for binary classification because we extract arguments in a role-by-role manner to make full use of the property of each argument role. In the future, we will extend it for multiclass classification and apply it to other tasks in natural language processing that face the problem of classifying universum class. Experimental results show that our RICB model achieves the SOTA performance and outperforms prior approaches on MUC-4 dataset.

References

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2021. Gate: Graph attention transformer encoder for cross-lingual relation and event extraction. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*.

Abhijit Bendale and Terrance E Boult. 2016. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572.

Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. 2021. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer.

X. Du, Alexander M. Rush, and Claire Cardie. 2021. Document-level event-based extraction using generative template-filling transformers. In *EACL*.

Xinya Du and Claire Cardie. 2020a. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020.

Xinya Du and Claire Cardie. 2020b. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683.

Sanghamitra Dutta, Liang Ma, Tanay Kumar Saha, Di Liu, Joel Tetreault, and Alejandro Jaimes. 2021. Gtn-ed: Event detection using graph transformer networks. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 132–137.

Joseph Fisher and Andreas Vlachos. 2019. Merge and label: A novel neural network architecture for nested ner. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5840–5850.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Mehadi Hassen and Philip K Chan. 2020. Learning a neural-network-based representation for open set recognition. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 154–162. SIAM.

Kung-Hsiang Huang and Nanyun Peng. 2021. Document-level event extraction with efficient end-to-end learning of cross-event dependencies. *NAACL HLT 2021*, page 36.

Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. 2019. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7842–7851.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797.

Chien-Liang Liu, Wen-Hoar Hsaio, Chia-Hoang Lee, Tao-Hsing Chang, and Tsung-Hsun Kuo. 2015. Semi-supervised text classification with universum learning. *IEEE transactions on cybernetics*, 46(2):462–473.

Xiao Liu, Zhunchen Luo, and He-Yan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256.

Dongfang Lou, Zhilin Liao, Shumin Deng, Ningyu Zhang, and Huajun Chen. 2021. MLBiNet: A cross-sentence collective event detection network. 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)*, pages 4829–4839.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332.

Virginia McLean. 1992. Fourth message understanding conference (muc-4).

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309.

Zhiquan Qi, Yingjie Tian, and Yong Shi. 2012. Twin support vector machine with universum data. *Neural Networks*, 36:112–119.

Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. 2019. Graphie: A graph-based framework for information extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 751–761.

Bharat Richhariya and Muhammad Tanveer. 2020. A reduced universum twin support vector machine for class imbalance learning. *Pattern Recognition*, 102:107150.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR.

Vladimir Vapnik. 2006. *Estimation of dependences based on empirical data*. Springer Science & Business Media.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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)*, pages 5784–5789.

Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021. CLEVE: Contrastive Pre-training for Event 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)*, pages 6283–6297. Association for Computational Linguistics.

Kaiwen Wei, Xian Sun, Zequn Zhang, Jingyuan Zhang, Guo Zhi, and Li Jin. 2021. Trigger is not sufficient: Exploiting frame-aware knowledge for implicit event argument 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)*, pages 4672–4682.

Jason Weston, Ronan Collobert, Fabian Sinz, Léon Bottou, and Vladimir Vapnik. 2006. Inference with the universum. In *Proceedings of the 23rd international conference on Machine learning*, pages 1009–1016.

Yanshan Xiao, Junyao Feng, and Bo Liu. 2021. A new transductive learning method with universum data. *Applied Intelligence*, pages 1–13.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, page 3533–3546.

Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021. Document-level event extraction via parallel prediction networks. 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)*, pages 6298–6308.

Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019a. Simple and effective text matching with richer alignment features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4699–4709.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019b. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2edag: An end-to-end document-level framework for chinese financial event extraction. 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)*, pages 337–346.

A Appendix

A.1 Dataset Information

Some supplementary information about the dataset is illustrated in this section. We use the MUC-4 dataset to evaluate the performance of our model. The dataset is intended for research purpose, which is consistent with our purpose of use. Beside the statistic information we provided in the main part, we illustrate the documentation of the dataset in this section. MUC-4 dataset is made of English news articles on the subject of terrorist attacks. Specifically, five arguments are extracted for the dataset: perpetrator individual, perpetrator organization, target, victim, and weapon.

A.2 Implementation Details

Spacy 3.0.3 is used in data preprocessing. Experiments are conducted on NVIDIA GTX 1080Ti, and the training time is about four hours. Experimental results of our RICB model are from the average of two experiments of different random seeds, and experimental results in ablation studies are from a single run. The hyper-parameters are given in the table below.

Hyper-parameter	**Value**
Embedding size	300
Hidden size	150
Bidirectional	True
Layers of encoder	2
Layers of graph2token module	1
Layers of graph	1
Heads of graph	2
Optimizer	Adam
Learning rate	$2e^{-4}$
Batch size	4
Dropout	0.3
Training epoch	130