Hyperspherical Multi-Prototype with Optimal Transport for Event Argument Extraction

Guangjun Zhang[1], **Hu Zhang**[1,2*], **Yujie Wang**[1], **Ru Li**[1,2], **Hongye Tan**[1,2], **Jiye Liang**[1,2]
[1]School of Computer and Information Technology, Shanxi University, Taiyuan, China
[2]Key Laboratory of Computational Intelligence and Chinese Information
Processing of Ministry of Education, Shanxi University, Taiyuan, China
zgj2866@gmail.com, zhanghu@sxu.edu.cn, init_wang@foxmail.com,
{liru,tanhongye,ljy}@sxu.edu.cn

Abstract

Event Argument Extraction (EAE) aims to extract arguments for specified events from a text. Previous research has mainly focused on addressing long-distance dependencies of arguments, modeling co-occurrence relationships between roles and events, but overlooking potential inductive biases: (i) semantic differences among arguments of the same type and (ii) large margin separation between arguments of the different types. Inspired by prototype networks, we introduce a new model named HMPEAE, which takes the two inductive biases above as targets to locate prototypes and guide the model to learn argument representations based on these prototypes. Specifically, we set multiple prototypes to represent each role to capture intra-class differences. Simultaneously, we use hypersphere as the output space for prototypes, defining large margin separation between prototypes to encourage the model to learn significant differences between different types of arguments effectively. We solve the "argument-prototype" assignment as an optimal transport problem to optimize the argument representation and minimize the absolute distance between arguments and prototypes to achieve compactness within sub-clusters. Experimental results on the RAMS and WikiEvents datasets show that HMPEAE achieves state-of-the-art performances.

1 Introduction

Event Argument Extraction (EAE) is an essential branch of event extraction that aims to extract the arguments of specified events from a text and predict their roles. EAE is critical in many downstream tasks, such as question and answer, recommender, and dialog systems.

In the EAE task, we observe two types of inductive biases. The first is the presence of intra-class variance, where arguments for the same role may fall into different sub-clusters in the embedding space due to semantic differences. For example, in Figure 1, both "America" and "Trump" play the role of "giver" in the event "gift grant provide aid". However, the former can be regarded as an "organization", while the latter denotes an "individual". Similarly, both "various organizations" and "refugees" play the role of "beneficiaries", with the former easily perceived as "entities" and the latter representing a people "group". The second is the lack of clear boundaries between different role arguments, leading to difficulties in explicitly delineating various argument representations in the embedding space and making model decisions challenging.

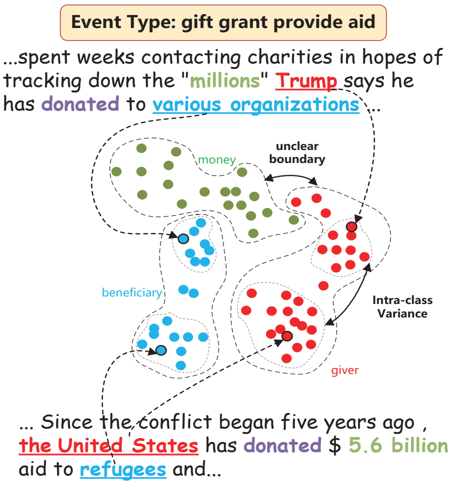

Event Type: gift grant provide aid
...spent weeks contacting charities in hopes of tracking down the "millions" Trump says he has donated to various organizations ...
money
unclear boundary
beneficiary
Intra-class Variance
giver
... Since the conflict began five years ago , the United States has donated $ 5.6 billion aid to refugees and...

Figure 1: We demonstrate two types of inductive biases present in EAE. (1) arguments with the same role may belong to different sub-clusters due to semantic differences, and (2) larger margin separations are ignored, leading to unclear boundaries between different categories.

In this paper, we propose the HMPEAE to address the two issues above. It is a prototype-based approach similar to prototype networks. In pro-

[*]Corresponding author.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9271–9284
August 11-16, 2024 ©2024 Association for Computational Linguistics

totype networks, the prototype is the mean output vector of all samples per class, guiding the model to learn the distribution of samples in the embedding space(Jetley et al., 2015; Gao et al., 2019; Li et al., 2019; Pan et al., 2019; Seth et al., 2019; Mettes et al., 2019). However, traditional prototype-based methods only assign one prototype per category, neglecting semantic variations within the same category. This oversight can lead to the model being influenced by intra-class variance. Additionally, the existing multi-prototype method(Wu et al., 2023) does not consider the large margin separation between classes. This leads to fuzzy boundaries in the embedding space for samples of different categories, causing model decision difficulties and poor robustness.

HMPEAE sets multiple prototypes for each role, allowing each prototype to represent a specific sub-cluster. In this way, we can capture the intra-class variance. Simultaneously, we use the hypersphere as the representation space(Mettes et al., 2019) for prototypes to maximize the distances between class prototypes and achieve a large inter-class separation. Further, we consider smaller distances between prototypes of roles with similar semantics and larger distances between prototypes of roles with larger semantic differences. Inspired by RankNet(Burges et al., 2005), we consider the similarity between semantics of roles as the prior information and employ a ranking-based loss function as one of the objectives for locating prototypes. We use the pre-trained prototypes to guide the EAE model in learning the arguments' representations. Specifically, we adopt TabEAE (He et al., 2023) as the backbone model, inheriting its key modules such as encoder, decoder, prompt templates, slot tables and span selectors. During training, each argument should be assigned an appropriate prototype to optimize its representation. We consider such an "argument-prototype" assignment an optimal transport problem while minimizing the absolute distances between arguments and prototypes to achieve compactness for the same sub-cluster. We summarize our contributions as follows:

- We trained a group of role prototypes with two overlooked inductive biases in mind, which set multiple prototypes for each role to capture intra-class differences and employ the hypersphere as the output space for large margin separation between classes.
- Each argument should be assigned an appropriate ground-truth prototype to optimize the its representation during EAE model training. We solve the "argument-prototype" assignment as an optimal transport problem.
- We construct experiments on RAMS and Wikievents. The results show that HMPEAE achieves sota performances, bringing about gain effects of 0.6 and 2.5 on Arg-C F1, respectively. The code is available at https://github.com/GJZhang2866/HMPEAE.

2 Methodology

In this section, we first describe the task definition of EAE in subsection 2.1, followed by a detailed description of the process of constructing hyperspherical multi-prototypes in subsection 2.2. We then briefly outline the backbone EAE model chosen for this paper in subsection 2.3, and finally explore a solution to the optimal transport problem between arguments and prototypes in subsection 2.4.

2.1 Task definition

Given a document $\mathcal{D} = \{w_i\}_{i=1}^{N_w}$ consisting of N_w words, with a predefined set of event types $\mathcal{E}$, corresponding role sets R_e for each event type $e \in \mathcal{E}$ and its trigger word $t_e \in \mathcal{D}$, the objective is to extract all (r, a) pairs from $\mathcal{D}$, where $r \in R_e$ is the role of argument a in the event e.

2.2 Hypersphere Prototypes

Due to semantic variations within a role, arguments of the same role will form multiple subclusters in the embedding space. Simultaneously, the boundaries between different roles of arguments in the embedding space become blurry, leading to challenging model decisions. To address these, we use multiple prototypes to represent each role type and employ the hypersphere as the output space of prototypes, capturing intra-class semantic variance and enabling large margin separation. As shown in Figure 2, we set up three targets to localize the hyperspherical multi-prototypes.

Hypersphere Single-Prototype. For ease of introduction, let's first consider the case of setting one prototype for each role. Specifically, for each role r, let p_r represent its prototype. Thus, $\mathbf{P} = \{p_i\}_{i=1}^{K}$ constitutes the prototypes for all K roles. The problem of how to distribute all prototypes as uniformly as possible on the hypersphere is known as

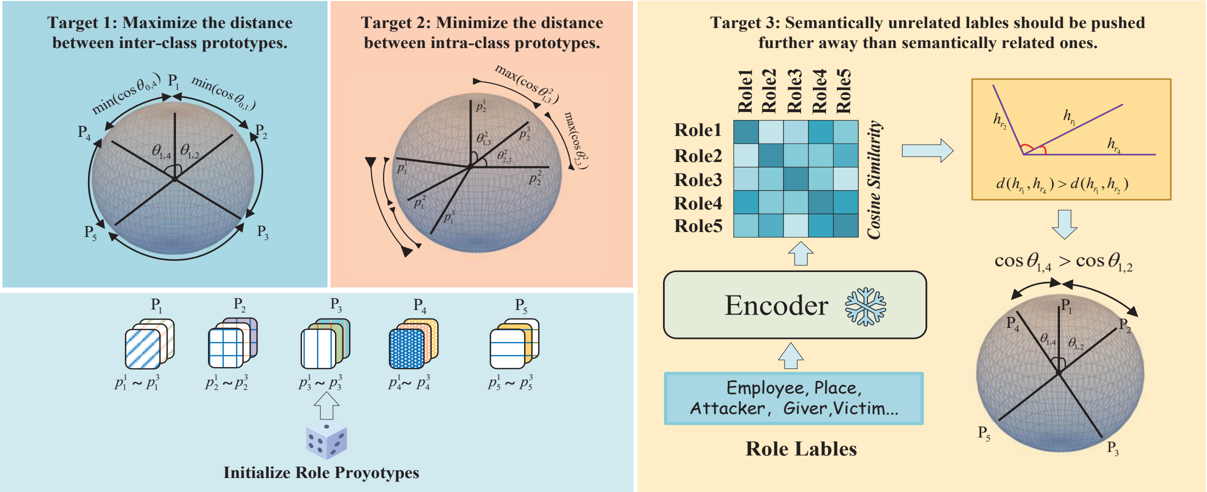

Target 1: Maximize the distance between inter-class prototypes.
$\min(\cos\theta_{1,4})$
$\min(\cos\theta_{1,2})$
P_1
P_2
P_3
P_4
P_5
$\theta_{1,4}$
$\theta_{1,2}$
Target 2: Minimize the distance between intra-class prototypes.
$\max(\cos\theta^{2}_{1,3})$
$\max(\cos\theta^{2}_{2,3})$
Target 3: Semantically unrelated lables should be pushed further away than semantically related ones.
Role1
Role2
Role3
Role4
Role5
Cosine Similarity
h_{r_1}
h_{r_2}
h_{r_4}
$d(h_{r_1},h_{r_4}) > d(h_{r_1},h_{r_2})$
$\cos\theta_{1,4} > \cos\theta_{1,2}$
Encoder
Employee, Place, Attacker, Giver,Victim...
Role Lables
$p^1_1 \sim p^3_1$
$p^1_2 \sim p^3_2$
$p^1_3 \sim p^3_3$
$p^1_4 \sim p^3_4$
$p^1_5 \sim p^3_5$
Initialize Role Proyotypes

Figure 2: Three Training Goals for Training Hypersphere Multi-prototypes. Here, we randomly initialize three prototypes for each role $P_i = \{p_i^j\}_{j=1}^3$. In target 3, we freeze the encoder parameters and obtain only the embedding representation of each role label.

the Tammes problem(Tammes, 1930). To address this issue, (Mettes et al., 2019) observes that the optimal set of prototypes $\mathbf{P}^* \in \mathbb{R}^{K\times d}$ is the one that minimizes the maximum cosine similarity between any two prototypes $p_i \in \mathbb{R}^d$ and $p_j \in \mathbb{R}^d$:

$$\mathbf{P}^* = \arg\min_{\mathbf{P}, i\neq j, i,j\in K} \cos\theta_{p_i,p_j} \tag{1}$$

where d is the hidden dimension. For this purpose, minimizing cosine similarity can be formulated as the objective function for optimization. However, computing all pairwise cosine similarities is highly inefficient. Therefore, the following optimization objective is proposed, where the goal is to optimize the prototype with the maximum similarity to the current prototype at each iteration:

$$\mathcal{L}_p = \frac{1}{K}\sum_{i=1}^{K} \max_{i\neq j, j\in K} \mathbf{M}_{ij} \tag{2}$$

$$\mathbf{M} = d(\widehat{\mathbf{P}}, \widehat{\mathbf{P}}^T) - 2\mathbf{E} \tag{3}$$

where $\widehat{\mathbf{P}} \in \mathbb{R}^{K\times d}$ is the current set of hyperspherical prototypes, $\mathbf{E}$ is the unit matrix, $\mathbf{M}$ denotes the pairwise prototype similarities and $\mathrm{d}(\cdot)$ represents the cosine distance metric. By optimizing $\mathcal{L}_p$, we can we can obtain a hyperspherical single prototypes.

Hypersphere Multi-Prototype. Next we extend the single-prototype to the multi-prototype case. Specifically, for each event role r, let $\mathcal{P}_r = \{p_1^r, ..., p_M^r\}$ be the set of M prototypes representing r. Then $\mathbf{P} = \{\mathcal{P}_i\}_{i=1}^K \in \mathbb{R}^{K\times M\times d}$ is s the collection of prototypes for all K roles. We aim to maximize the distance between prototypes of different roles while minimizing the distance between prototypes of the same role, so we set the following loss:

$$\mathcal{L}_{inter} = \frac{1}{K}\sum_{i=1}^{K}(\max_{b(i)\neq b(j), j\in K} \mathbf{M}_{ij}) \tag{4}$$

$$\mathcal{L}_{intra} = \frac{1}{KM}\sum_{k}^{K}\sum_{ij}^{M}(1 - \min_{i\neq j} \mathbf{M}_{ij}) \tag{5}$$

$$\mathcal{L}_p = \mathcal{L}_{inter} + \mathcal{L}_{intra} \tag{6}$$

where $\mathrm{b}(i)$ denotes the role represented by the i-th prototype. For Eq.4, we only consider the inter-class situations thus excluding prototypes that belong to the same role, i.e., $b(i) \neq b(j)$. For Eq.5, we only compute prototypes belonging to the same role and set the optimization objective to maximize the minimum similarity between these prototypes.

Prototype with prior information. While separating prototypes of different roles is crucial, semantically unrelated roles should be farther apart than semantically related ones. To incorporate this prior knowledge during the prototype construction process, we leverage BERT to encode the semantic information for each role, represented as $H_R = \{h_{r_1}, \ldots, h_{r_K}\}$. Inspired by RankNet(Burges et al., 2005), we use a ranking-based loss function(Mettes et al., 2019):

$$\bar{S}_{ijk} = [\![d(\mathbf{w}_i, \mathbf{w}_j) \leq d(\mathbf{w}_i, \mathbf{w}_k)]\!] \tag{7}$$

$$o_{ijk} = \mathrm{d}(p_i, p_j) - \mathrm{d}(p_i, p_k) \tag{8}$$

$$S_{ijk} = \frac{e^{o_{ijk}}}{1 + e^{o_{ijk}}} \tag{9}$$

$$\mathcal{L}_{\text{rank}} = \frac{1}{T} \sum_{(i,j,k) \in T} \Big(-\bar{S}_{ijk} \log S_{ijk} \\ -(1 - \bar{S}_{ijk}) \log(1 - S_{ijk}) \Big) \tag{10}$$

where T denotes the set of all role triples. The ground truth $\bar{S}_{ijk}$ denotes the ordering of role triples, with a indicator function $[\![\cdot]\!]$. S_{ijk} represents the likelihood of ranking. $\mathcal{L}_{\text{rank}}$ optimizes the hyperspherical prototypes, aligning its ranking order with the semantic priors. We jointly optimize the prototype by combining the ranking loss function with the formula loss function:

$$\mathcal{L}_{hp} = \mathcal{L}_p + \mathcal{L}_{rank} \tag{11}$$

After above steps, we are able to obtain hyperspherical multi-prototypes $\mathbf{P}^* = \{\mathcal{P}_i^*\}_{i=1}^K$ for K roles and each role is represented by M prototypes denoted as $\mathcal{P}_i^* = \{p_{\text{i}}^j\}_{i=1}^M$

2.3 Backbone EAE Model

We choose TabEAE(He et al., 2023) as our backbone model and inherit its encoder, decoder, prompt templates, slotted tables and span selector for extracting event arguments.

Trigger-aware context encoding. Given a word sequence $\mathcal{D} = \{w_i\}_{i=1}^{N_w}$ containing k triggers, we use a pair of tokens(<T-i>,</T-i >) to mark the i-th trigger t_i in the sequence, obtaining the input sequence X. We then feed X into a transformer-based encoder to obtain the contextual representations of $\mathcal{D}$. Furthermore, it is passed to the decoder to obtain an event-oriented contextual representation $H_X \in \mathbb{R}^{l_c}$.Here, we skip the computation of cross-attention in the decoder and only updated contextual representation H_X through the self-attention module:

$$H_X^{enc} = Encoder(\text{X}) \tag{12}$$

$$H_X = Decoder(H_X^{enc}, H_X^{enc}) \tag{13}$$

Slotted Table Construction. We follow (He et al., 2023), modeling the co-occurrence relationships of events by constructing a Slotted Table and using it as input for the decoder. Specifically, the prompt for each event is fed into the encoder in parallel,

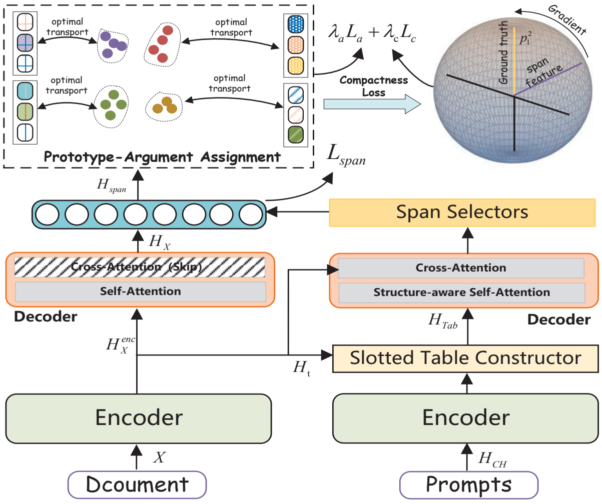


Figure 3: The overview of HMPEAE. We use TabEAE(He et al., 2023) as the backbone model. During training , we collect predicted argument features, solve the assignment problem between prototypes and arguments using the optimal transport algorithm, and introduce a compactness loss to achieve compactness within each sub-cluster.

resulting in initialized column headers representation.

$$H_P^{E_j} = Encoder\left(P_{E_j}\right) \tag{14}$$

$$H_{CH} = [H_P^{E_1} : \ldots : H_P^{E_T}] \tag{15}$$

where P_{E_j} is the j-th prompt,and T is the number of event types. H_{CH} is the concatenation of all prompts' representations. The i-th row of the table starts with the i-th trigger, followed by the argument slots S_i of the i-th prompt. The initialization representations of the table can be obtained through the following equation:

$$H_{Tab} = [H_{CH} : H_{t_1} : H_{S_1} : \ldots : H_{t_N} : H_{S_N}] \tag{16}$$

where H_{t_i} is the embedding of i-th trigger word, copied from H_X^{enc}. H_{S_i} is the average embedding of the corresponding role and trigger.

Non-autoregressive table decoder The non-autoregressive table decoder set a structure-aware self-attention with mask M_{Tab} constructed by (He et al., 2023) to make each element of the table attend to the region related to it. The cross-attention mechanism is the same as the one in Transformer(Vaswani et al., 2017).

$$\widetilde{H}_{Tab} = Decoder(H_{Tab}, H_X^{enc}) \tag{17}$$

Span Selector. We can obtain the representation of the argument slots H_S form last step output $\widetilde{H}_{Tab}$. We transform each slot representation

$h_{S_k} \in H_S$ through a linear transformation into a span selector $\{\varphi_k^s, \varphi_k^e\}$:

$$\varphi_k^s = h_{S_k} w_s \tag{18}$$

$$\varphi_k^e = h_{S_k} w_e \tag{19}$$

where w_s and w_e are learnable weights. The span selector $\{\varphi_k^s, \varphi_k^e\}$ is responsible for choosing a suitable span for the slot k in the context embeddings:

$$p_k^s = Softmax(H_X \varphi_k^s) \tag{20}$$

$$p_k^e = Softmax(H_X \varphi_k^e) \tag{21}$$

$$(\widehat{s}_k, \widehat{e}_k) = \arg max_{(i,j)\in l_c}(p_k^s(i) + p_k^e(j)) \tag{22}$$

where l_c is the context length, $\widehat{s}_k$ and $\widehat{e}_k$ are the start and end indices of the best span. Then, we obtain the argument representation $\boldsymbol{h}_a^k \in \mathbb{R}^{d_h}$.

Furthermore, we follow the approach of (Ma et al., 2022; He et al., 2023), utilizing the Hungarian algorithm to solve the assignment problem between predicted argument spans and golden argument spans. We optimize the process using a bipartite matching loss:

$$\mathcal{L}_{span} = -\sum_{i=1}^{N}\sum_{k}(\log p_k^s(s_k) + \log p_k^e(e_k)) \tag{23}$$

where (s_k, e_k) are the golden span assigned to the k-th argument slot.

2.4 Token-Prototype Assignment

As shown in Figure 3, we introduce role prototypes to guide the backbone model learning argument representations. Due to multiple prototypes representing each role, assigning each argument to the appropriate ground-truth prototype is necessary to optimize argument representations. We use $\mathcal{P}_r^* \in \mathbb{R}^{M\times d}$ to denote M prototypes for role r and $\mathcal{A}_r$ is represented as the set of arguments predicted as r. Each element $h_a^i \in \mathcal{A}_r$ is the argument representation, which can be obtain by mean pooling: $h_a^i = Mean\_pooling(H_X[\hat{s}_i{:}\hat{e}_i])$. We aim to compute the assignment matrix $\gamma^r \in \mathbb{R}^{|\mathcal{A}_r|\times M}$. We consider argument-prototype assignment as an optimal transport problem:

$$\hat{\gamma}^r = \arg\min_{\gamma^r}\sum_{i\in\mathcal{A}_r}\sum_{j=1}^{M}\gamma_{i,j}^r\mathbf{C}_{i,j}^r \tag{24}$$
$$\text{s.t.}\quad \hat{\gamma}^r\mathbb{1} = \boldsymbol{a}, \quad \hat{\gamma}^{r^{\top}}\mathbb{1} = \boldsymbol{b},$$

where $\mathbf{C}_{i,j}^r = d(\boldsymbol{h}_a^i, \boldsymbol{p}_j^r)$ serves as the cost matrix, representing the distance between $\mathcal{A}_r$ and $\mathcal{P}_r^*$. There are two constraint conditions here: (1) $\boldsymbol{a} = \mathbb{1} \in \mathbb{R}^{|\mathcal{A}_r|}$ ensures that each argument can be assigned to one prototype, and (2) $\mathbf{b} = \frac{|\mathcal{A}_r|}{M}\mathbb{1} \in \mathbb{R}^M$ prevents all arguments from being assigned to the same prototype. We follow (Wu et al., 2023), simply set it to an even distribution We then utilize the sinkhorn-knopp(Cuturi, 2013) algorithm to address the assignment problem. We show more details in Appendix A.This approach allows us to obtain a prototype assignment for all arguments. We optimize the matching through standard cross-entropy loss:

$$\mathcal{L}_a = -\sum_{i\in r} log\frac{\exp\left(d(h_a^i, p_c)\right)}{\sum_{j\in K\times M}\exp\left(d(h_a^i, p_j)\right)}, \tag{25}$$

where p_c is the ground-truth prototype obtained from the previous step. To achieve compactness of argument features within the same subset cluster, we further optimize the absolute distance between argument features and the ground-truth prototypes:

$$\mathcal{L}_c = \sum_{i\in r}(1 - d(h_a^i, p_c))^2. \tag{26}$$

The total loss can be calculated as follows:

$$\mathcal{L} = \mathcal{L}_{span} + \lambda_a\mathcal{L}_a + \lambda_c\mathcal{L}_c, \tag{27}$$

where λ_a and λ_c are regularization weights.

Prototype Updating. During the EAE model training, we update each prototype based on the argument features assigned to it. At each training step t, we update the prototype using an exponential moving average (EMA):

$$p_i^t = \alpha p_i^{t-1} + (1-\alpha)\frac{\sum_{i\in|\mathcal{A}|}\boldsymbol{h}_a^i}{|\mathcal{A}|}, \tag{28}$$

where α is the EMA update rate. In this manner, the learned prototype can be considered as a representation of some sub-cluster in the embedding space.

3 Experiment

3.1 Experimental Settings

Dataset and Metrics We conduct experiments on two widely used public datasets: RAMS(Ebner et al., 2020) and Wikievents(Li et al., 2021). Following previous works (Ma et al., 2022; He et al., 2023), we employ Argument Identification F1 (Arg-I) and Argument Classification F1 (Arg-C)

Model	PLM	RAMS		WikiEvents	
		Arg-I	Arg-C	Arg-I	Arg-C
EEQA*(2020b)	BERT	48.7	46.7	56.9	54.5
EEQA*(2020b)	RoBERTa	51.9	47.5	60.4	57.2
BART-Gen*(2021)	BART	51.2	47.1	66.8	62.4
TSAR*(2022)	RoBERTa	57.0	52.1	71.1	65.8
PAIE* (2022)	BART	57.1	52.6	70.2	65.1
SCPRG(2023b)	RoBERTa	-	52.3	-	-
(Ren et al., 2023)	T5	54.6	48.4	69.6	63.4
SPEAE (2023)	BART	$\underline{58.0}$	$\underline{53.3}$	$\underline{71.9}$	$\underline{66.1}$
TabEAE*(m2m)(2023)	RoBERTa	56.5	52.2	70.5	64.5
TabEAE*(m2s)(2023)	RoBERTa	57.3	53.1	69.8	63.9
HMPEAE(Ours)	RoBERTa	**58.6**	**53.7**	**72.1**	**66.6**

Table 1: Overall results. We highlight the best result and underline the second-best result. * indicates that we have rerun the relevant code. The symbol * indicates results from He et al. All pre-trained models (PLMs) are of large-scale. Missing values are due to unreported metrics in the original paper. Other unmarked results are sourced from the original paper.

as evaluation metrics. Arg-I refers to the correctness of predicted arguments when the boundaries of the predicted arguments match with any golden arguments; Arg-C indicates correctness only when both the boundaries and role types of the predicted arguments are correct.

Implementation Details During the training of the hyperspherical multi-prototype, we set up 2 prototypes for each role in the RAMS dataset and 4 prototypes for each role in the WikiEvents dataset. The experimental results in section 3.4 demonstrate that this is the optimal setup. We use Bert-base(Devlin et al., 2018) as the encoder for label semantic vectors and employ the SGD optimizer with a learning rate of 0.1, momentum of 0.9, and run for 10,000 epochs. For training the EAE model, we set EMA update rate $\alpha = 0.9$ and $\lambda_c = 0.1$. The other hyperparameters are consistent with TabEAE(He et al., 2023). All experiments we reran are only single experiment results. We also report the average results with 5 random seed in Appendix C.

Baselines We compare different classes of EAE models, which mainly consist of classification-based methods, e.g., EEQA(Du and Cardie, 2020b), TSAR(Xu et al., 2022), SCPRG(Liu et al., 2023b) and generation-based methods, e.g., BART-Gen(Li et al., 2021), PAIE(Ma et al., 2022), (Ren et al., 2023), SPEAE(Nguyen et al., 2023), TabEAE(He et al., 2023). We show more details in Appendix B.

3.2 Main results

We evaluate the proposed model HMPEAE and baseline methods under all benchmarks. The overall performances of our method compared to baseline models is presented in Table 1. In contrast to the baseline model PAIE, TabEAE, HMPEAE demonstrates comprehensive improvements across both datasets.Specifically, HMPEAE achieves gains of 0.6 and 1.3 in Arg-I and Arg-C metrics, respectively, on RAMS. And it shows an improvement of 1.3 in Arg-I F1 1.9 and 2.5 in Arg-C F1 on WikiEvents.

Compared to the previous state-of-the-art models, on the WikiEvents dataset, HMPEAE surpasses SPEAE by 0.2 in Arg-I and 0.5% in Arg-C metrics. On the RAMS dataset, HMPEAE outperforms SPEAE, with an increase of 0.6 in Arg-I F1 and 0.4 in Arg-C F1. These results demonstrate the effectiveness of our proposed method.

The performance improvement can be attributed to two main factors: (1) The utilization of multiple prototypes to represent intra-class variance enables better capture of intra-class semantic differences, thereby enhancing the robustness of the model. (2) Employing hyperspheres as prototype output spaces facilitates significant margin separation, enhancing the model's decision-making capabilities.

Model	RAMS		Wikievents	
	Arg-I	Arg-C	Arg-I	Arg-C
w/o Privileged info	57.1	52.3	69.3	63.1
w/o Multiple Prototype	56.9	51.4	70.2	65.1
w/o Hypersphere	56.4	52.1	70.2	63.8
w/o Compactness Loss	57.1	52.4	69.0	63.8
w/o $\mathcal{L}_a$	57.3	52.6	69.9	65.0
w/o EMA	55.7	50.9	70.5	65.5
HMPEAE	**58.6**	**53.7**	**72.1**	**66.6**

Table 2: Ablation experiments on both datasets.

3.3 Ablation Studies

To verify the validity of different components in HMPEAE, we perform a rich ablation study on two datasets, and the results are shown in Table 2. (1) **w/o Privileged info**. We deprecated role embedding as a priori information when locating more than one hyperspherical prototype. (2) **w/o Multiple Prototypes**. We dropp multiple prototypes and set only one prototype for each role. (3) **w/o Hypersphere**. We remove the hypersphere setting, i.e., instead of using the principle of large margin separation to locate prototypes, we simply randomly generate multiple prototypes for each role.(4) **w/o Compactness Loss**. In training the EAE model, we remove the compactness loss. (5) **w/o $\mathcal{L}_a$**. We remove the cross-entropy loss between spans and prototypes and kept only the compactness loss to optimize the absolute distance of arguments to the matched prototypes. (6)**w/o EMA.** During training, we freeze the prototypes and do not optimize them.

The ablation experiments show that after removing the settings of multiple prototypes and hyperspheres, the model lost the ability to capture intra-class semantic differences and handle decision difficulties, leading to a decrease in performance. Using role semantic similarity as prior information in locating prototypes can enhance the model's performance. During EAE training, compact loss and cross-entropy loss between prototypes and arguments have been proven necessary. Additionally, further fine-tuning of prototypes by EMA has positively contributed to improving performance.

3.4 Experiment on Different Number of Prototypes

We analyze the impact of the number of prototypes by increasing the number of role prototypes to find the optimal setup for each dataset. As shown in Table 3, setting two prototypes for each role achieves the best performance on RAMS. Setting three prototypes for each role achieves the best performance on WikiEvents. We did not conduct prototype experiments with more settings because additional prototypes would incur higher computational costs. And setting too large will affect the performance because there may not be enough argument features to learn representative prototypes, which leads to underfitting.

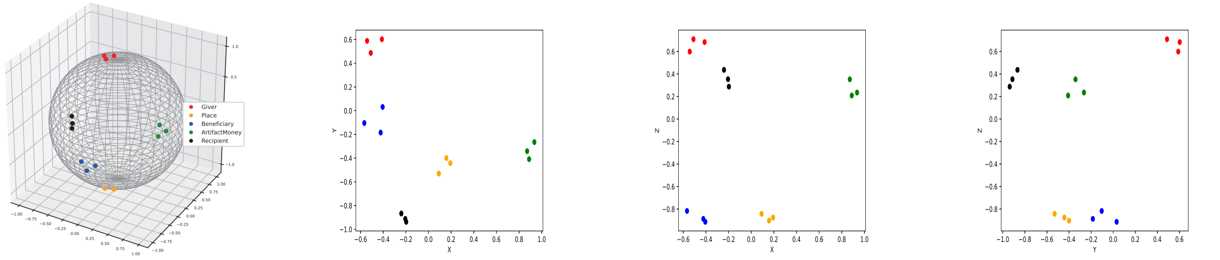
M	RAMS		Wikievents	
	Arg-I	Arg-C	Arg-I	Arg-C
1	57.1	52.3	69.3	63.1
2	**58.6**	**53.7**	70.2	65.1
3	57.1	52.4	**72.1**	**66.6**
4	57.3	52.6	69.9	65.0

Table 3: Experiment with the different number of prototypes. M denotes the number of prototypes for each role

3.5 Visual Analysis

Visualization of Prototypes In Figure 4, we visualize the prototypes of roles relevant to event type "Transaction.Donation.Unspecified" from WikiEvents in the form of a 3D hypersphere and simultaneously project them onto three 2D planes for display. The prototypes have essentially achieved our predefined targets, which involve maximizing the distance between inter-class prototypes while minimizing the distance between intra-class prototypes. Additionally, the "Giver" -"Recipient" prototype distances are shorter than the "Giver" -"Place" distances, aligning with our intuition and meeting the third objective: similar roles' prototypes have smaller distances than dissimilar ones.

Visualization of Arguments. We extract argument features of events type "contact.requestadvise.n/a" from the best checkpoint on RAMS and transform them into 2D features using t-SNE. As shown in Figure 5, firstly, the arguments playing the role of "place" form two sub-clusters in the feature space, which suggests intra-class variation. HMPEAE can capture such intra-class variance by setting multiple prototypes for each role so that arguments of the same type are more compact than TabEAE. Second, compared to TABEAE, there is a clear separation between the argument types of "place" and "recipient" in HMPEAE. Additionally, we observe that arguments of "recipient" does not partition into multiple sub-clusters within the feature space.



(a) Visualization of role prototypes on the hypersphere

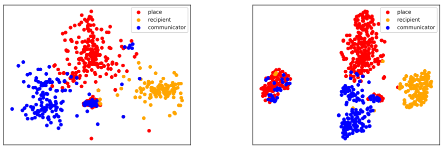
(b) Visualization of role prototypes on the X-Y plane.

(c) Visualization of role prototypes on the X-Z plane.

(d) Visualization of role prototypes on the Y-Z plane.

Figure 4: Role prototypes for event type *Transaction.Donation.Unspecified* from WikiEvents, where we set three prototypes for each role. We use different colors to represent different role prototypes.

This suggests that not all roles exhibit significant semantic differences.



(a) TabEAE

(b) HMPEAE

Figure 5: The t-SNE visualization of part of argument features.

4 Related Work

4.1 Classification-based Event Argument Extraction

Initially, (Du and Cardie, 2020a; Wang et al., 2021) adopt the paradigm of sequence labeling based on BiLSTM-CRF for extracting event arguments. (Huang and Peng, 2020) employs a deep value network to capture cross-event dependencies. (Du and Cardie, 2020b; Wei et al., 2021; Liu et al., 2021) construct questions using predefined templates, extracting corresponding arguments in a question-answering (QA) paradigm. (Xu et al., 2022) proposes an AMR-guided two-stream encoder to address long-distance dependency issues. (Liu et al., 2023b) introduces context pooling to capture context clues and role relevance. (Liu et al., 2023a) proposes a chain reasoning paradigm, utilizing first-order logical rules to capture long-distance dependency relationships between candidate spans. (Zhou and Mao, 2022) construct an entity co-reference graph to learn entity representations with co-reference awareness. (Li et al., 2023) introduces an event intra- and inter-dependency-aware graph network for establishing dependency information between event argument roles.

4.2 Generation-based Event Argument Extraction

With the rise of large language models and prompt-based learning, there has been widespread attention on extracting event arguments in generations. (Du et al., 2021) naturally model the dependencies between entities and events by constructing event templates, utilizing generative Transformers to obtain arguments. (Zeng et al., 2022) devises a context-enhanced event-aware argument extraction method to enhance argument consistency. (Ma et al., 2022) proposes a prompt-based approach to extract arguments through slot filling. Building on this, (He et al., 2023) models the co-occurrence relationships of events, extending the prompt-based EAE model into a non-autoregressive generation framework for parallel extracting arguments from multiple events. (Nguyen et al., 2023) introduces soft prompts to enhance the representation of multiple related document encodings. (Ren et al., 2023) designs a retrieval-enhanced method to non-parametrically incorporate prior external knowledge, enhancing event argument extraction by sampling pseudo-examples from the semantic region of events.

5 Conclusion

In this paper, we identify two potential inductive biases overlooked in the EAE task. Therefore, we propose the HMPEAE, which first pre-trains a set of role hyperspherical multi-prototypes targeting these two inductive biases and directs the EAE model to learn argument representations based on these prototypes. During EAE model training, we solve argument-prototype assignment as an optimal transport problem. Experimental results on two commonly used datasets show that HMPEAE achieves state-of-the-art performance, confirming the effectiveness of the approach proposed in this study.

Limitations

There are still limitations in our model that need further improvement.

- For different types, there are different numbers of sub-clusters in the embedding space. Our approach uniformly sets the same number of prototypes for each role, which may increase their inter-class variance for classes that semantically do not have intra-class differences while it does not fully capture the internal variance for categories that contain more subclusters.

- Some subclusters have relatively sparse samples, which leads to incomplete matching between prototypes and arguments during the training process, resulting in underfitting phenomena. Therefore, further research is needed to extend the hyperspherical multi-prototype to few-shot scenarios.

Ethics Statement

This paper does not involve the presentation of a new dataset and the utilization of demographic or identity characteristics information.

Acknowledgements

We thank all the anonymous reviewers for their constructive comments and suggestions. This work is supported by the National Key Research and Development Program of China (2020AAA0106100), the National Natural Science Foundation of China (62176145, 62076155), the Key Research and Development Project of Shanxi Province (202102020101008).

References

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. 2005. Learning to rank using gradient descent. In *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pages 89–96. ACM.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2292–2300.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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, ACL 2020, Online, July 5-10, 2020*, pages 8010–8020. Association for Computational Linguistics.

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 2020, Online, November 16-20, 2020*, pages 671–683. Association for Computational Linguistics.

Xinya Du, Alexander M. Rush, and Claire Cardie. 2021. GRIT: generative role-filler transformers for document-level event entity extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 634–644. Association for Computational Linguistics.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8057–8077. Association for Computational Linguistics.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6407–6414. AAAI Press.

Yuxin He, Jingyue Hu, and Buzhou Tang. 2023. Revisiting event argument extraction: Can EAE models learn better when being aware of event co-occurrences? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12542–12556, Toronto, Canada. Association for Computational Linguistics.

Kung-Hsiang Huang and Nanyun Peng. 2020. Efficient end-to-end learning of cross-event dependencies for document-level event extraction. *CoRR*, abs/2010.12787.

Saumya Jetley, Bernardino Romera-Paredes, Sadeep Jayasumana, and Philip H. S. Torr. 2015. Prototypical priors: From improving classification to zero-shot learning. In *Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015*, pages 120.1–120.12. BMVA Press.

Hao Li, Yanan Cao, Yubing Ren, Fang Fang, Lanxue Zhang, Yingjie Li, and Shi Wang. 2023. Intra-event and inter-event dependency-aware graph network for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 6362–6372. Association for Computational Linguistics.

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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 894–908. Association for Computational Linguistics.

Xiao Li, Min Fang, Dazheng Feng, Haikun Li, and Jin-Qiao Wu. 2019. Prototype adjustment for zero shot classification. *Signal Process. Image Commun.*, 74:242–252.

Jian Liu, Yufeng Chen, and Jinan Xu. 2021. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2725, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jian Liu, Chen Liang, Jinan Xu, Haoyan Liu, and Zhe Zhao. 2023a. Document-level event argument extraction with a chain reasoning paradigm. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9570–9583. Association for Computational Linguistics.

Wanlong Liu, Shaohuan Cheng, Dingyi Zeng, and Hong Qu. 2023b. Enhancing document-level event argument extraction with contextual clues and role relevance. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12908–12922. Association for Computational Linguistics.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6759–6774. Association for Computational Linguistics.

Pascal Mettes, Elise van der Pol, and Cees Snoek. 2019. Hyperspherical prototype networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 1485–1495.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Chien Van Nguyen, Hieu Man, and Thien Huu Nguyen. 2023. Contextualized soft prompts for extraction of event arguments. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4352–4361. Association for Computational Linguistics.

Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. 2019. Transferrable prototypical networks for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2239–2247. Computer Vision Foundation / IEEE.

Yubing Ren, Yanan Cao, Ping Guo, Fang Fang, Wei Ma, and Zheng Lin. 2023. Retrieve-and-sample: Document-level event argument extraction via hybrid retrieval augmentation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 293–306. Association for Computational Linguistics.

Harshita Seth, Pulkit Kumar, and Muktabh Mayank Srivastava. 2019. Prototypical metric transfer learning for continuous speech keyword spotting with limited training data. In *14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019) - Seville, Spain, May 13-15, 2019, Proceedings*, volume 950 of *Advances in Intelligent Systems and Computing*, pages 273–280. Springer.

Pieter Merkus Lambertus. Tammes. 1930. On the origin of number and arrangement of the places of exit on the surface of pollen-grains.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 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, Online. Association for Computational Linguistics.

Shuhui Wu, Yongliang Shen, Zeqi Tan, Wenqi Ren, Jietian Guo, Shiliang Pu, and Weiming Lu. 2023. Mproto: Multi-prototype network with denoised optimal transport for distantly supervised named entity recognition. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2361–2374. Association for Computational Linguistics.

Runxin Xu, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. 2022. A two-stream amr-enhanced model for document-level event argument extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 5025–5036. Association for Computational Linguistics.

Qi Zeng, Qiusi Zhan, and Heng Ji. 2022. Ea^{2}e: Improving consistency with event awareness for document-level argument extraction. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2649–2655. Association for Computational Linguistics.

Hanzhang Zhou and Kezhi Mao. 2022. Document-level event argument extraction by leveraging redundant information and closed boundary loss. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3041–3052. Association for Computational Linguistics.

A Sinkhorn-Knopp Algorithm

We apply the sinkhorn-knopp algorithm (Cuturi, 2013) to solve for optimal transportation. Also we follow (Wu et al., 2023) to add an entropy regularizer as follows:

$$\hat{\gamma} = \arg\min_{\gamma} \sum_{j} \sum_{j} \gamma_{i,j} \mathbf{C}_{i,j} + \lambda^r \, \mathrm{H}(\gamma), \qquad \text{s.t.} \quad \gamma \mathbf{1} = \mathbf{a}, \gamma^\top \mathbf{1} = \mathbf{b} \tag{29}$$

where λ^r is the weight of the regularization and $\mathrm{H}(\gamma) = \sum_{i,j} \gamma_{i,j} \log(\gamma_{i,j}$ is the entropy of the assignment matrix. Algorithm 1 is the pseudo-code of the sinkhorn-knopp algorithm. Here the $\oslash$ denotes the element-wise division, **a** and **b** are vectors that represent the weights of each sample in the source and target distributions.

Algorithm 1 Sinkhorn-Knopp Algorithm

```
Require: $\mathbf{a}, \mathbf{b}, \mathbf{C}, \lambda^r$
  $\mathbf{u}^0 = \mathbf{1}, \mathbf{K} = \exp(-\mathbf{C}/\lambda^r)$
  for $i$ in $1, \ldots, n$ do
    $\mathbf{v}^i = \mathbf{b} \oslash \mathbf{K}^\top \mathbf{u}^{i-1}$
    $\mathbf{u}^i = \mathbf{a} \oslash \mathbf{K}\mathbf{v}^i$
  end for
  return $\gamma = \mathrm{diag}(\mathbf{u}^n)\mathbf{K}\,\mathrm{diag}(\mathbf{v}^n)$
```

B Baselines

Here we present the details of the compared baselines in detail.

EEQA(Du and Cardie, 2020b), redefines the EE task as a question-answering task.

BART-Gen(Li et al., 2021) consider EE tasks as seq-to-seq condition generation.

TSAR(Xu et al., 2022) adopts abstract semantic representation for EAE and proposes local encoder and global encoder to capture of contextual information.

PAIE(Ma et al., 2022) is the first to perform EAE task based on prompt learning.

SCPRG(Liu et al., 2023b) proposes context pooling to capture context clues.

(Ren et al., 2023) introduces Retrieval-Enhanced Strategies for Extracting Event arguments.

SPEAE (Nguyen et al., 2023) introduces soft prompts to facilitate the encoding of individual example context and multiple relevant documents to boost EAE.

TabEAE (He et al., 2023) models the co-occurrence relationships of events, extending the prompt-based EAE model into a non-autoregressive generation framework for parallel extracting arguments from multiple events.

Model	PLM	RAMS		WikiEvents	
		Arg-I	Arg-C	Arg-I	Arg-C
TabEAE*(m2s)(2023)	RoBERTa	56.8	52.2	70.3	64.6
HMPEAE(Ours)	RoBERTa	**57.9**	**53.2**	**71.5**	**66.0**

Table 4: Overall results averaged over 5 different random seeds.

C Additional Experiments

To validate the stability of the model we proposed and ensure a fair comparison with the baselines, we additionally conducted experiments under four different random seeds and averaged the results. The results are shown in Table 4. The results show that our method still has advantages. Among the five results, for the RAMS, the ranges of Arg-I and Arg-C for HMPEAE are 57.3~58.6 and 52.9~53.7, respectively. The ranges of Arg-I and Arg-C for TabEAE are 56.0~57.4 and 51.7~53.1, respectively. For the WikiEvents, The ranges of Arg-I and Arg-C for HMPEAE are 70.8~72.1 and 65.5~66.6, respectively. The ranges of Arg-I and Arg-C for TabEAE are 69.3~71.3 and 63.9~65.6, respectively.

D Datasets

In this section we describe the two datasets used for the experiments and present the data statistics in Table 5.

RAMS is derived from English online news. Since the original dataset was stored on an event-by-event basis, we followed the (He et al., 2023) method to merge data from different events in the same document.

WikiEvents is collected from English articles in Wikipedia. We only use the exact argument annotations in our experiments. experiments.

Dataset	**RAMS**	**WikiEvents**
# Event types	139	50
# Args per event	2.33	1.40
# Events per text	1.25	1.78
# Roles	65	80
# Events		
Train	7329	3241
Dev	924	345
Test	871	365

Table 5: The basic information for both datasets, where Args stands for Arguments.

E Few-shot Performance

We analyze the performance of HMPEAE in a few-shot setting and conduct experiments based on the RAMS dataset. As shown in Figure 6, we compare our approach with PAIE and TabEAE at different data sampling ratios. In a multi-event training mode, (He et al., 2023) employs a depth-first search during the data processing stage to maximize the utilization of co-occurrence relationships between events to obtain various event combinations under the same document, thereby expanding the dataset. Therefore, we train TabEAE and HMPEAE to ensure fair experimentation using a single-single training-inference scheme. In the case of limited data, HMPEAE did not demonstrate significant advantages; only as data increased did its advantages gradually become apparent. The reason for this was the randomness of the sampling, where samples under the same class did not show intra-class differences, which results in insufficient features to train the matching relationship between the prototype and arguments. In contrast, HMP w/o Multiple Prototype sets one prototype for each class, retaining only the maximum separation between classes. The HMP wo/Multiple Prototype demonstrates the best performance in the case of scarce training samples, further validating the effectiveness of its separation effect.

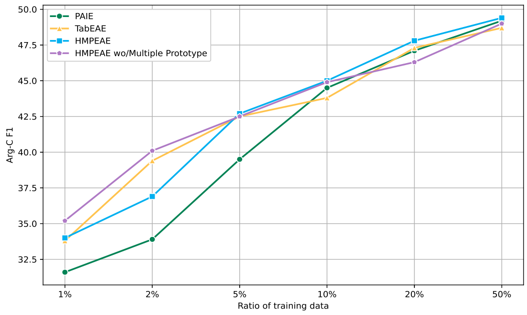
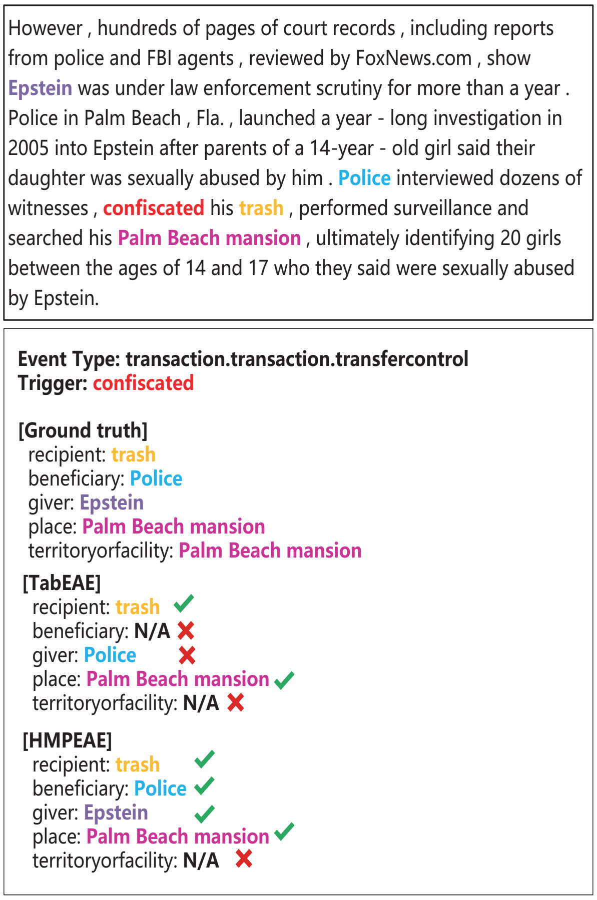


Figure 6: Arg-C F1 score on RAMS test set with different data ratio.

F Case Study

Figure 7 illustrates an individual test case from RAMS. Among them, "confiscated" triggers the "transaction.transaction.transfercontrol" event. TabEAE erroneously predicts "Police" as "giver", whereas it should fundamentally be "beneficiary", two roles prone to confusion. In contrast, HMPEAE, by establishing maximum boundaries between categories, manages to circumvent such errors.



However , hundreds of pages of court records , including reports from police and FBI agents , reviewed by FoxNews.com , show Epstein was under law enforcement scrutiny for more than a year . Police in Palm Beach , Fla. , launched a year - long investigation in 2005 into Epstein after parents of a 14-year - old girl said their daughter was sexually abused by him . Police interviewed dozens of witnesses , confiscated his trash , performed surveillance and searched his Palm Beach mansion , ultimately identifying 20 girls between the ages of 14 and 17 who they said were sexually abused by Epstein.

Event Type: transaction.transaction.transfercontrol
Trigger: confiscated

[Ground truth]
recipient: trash
beneficiary: Police
giver: Epstein
place: Palm Beach mansion
territoryorfacility: Palm Beach mansion

[TabEAE]
recipient: trash ✓
beneficiary: **N/A** ✗
giver: Police ✗
place: Palm Beach mansion ✓
territoryorfacility: **N/A** ✗

[HMPEAE]
recipient: trash ✓
beneficiary: Police ✓
giver: Epstein ✓
place: Palm Beach mansion ✓
territoryorfacility: **N/A** ✗

Figure 7: A case from RAMS.

G Error Analysis

In this section, we will manually examine the 100 error predictions in the RAMS test set and analyze their causes.

Absence of intra-class variance. Intra-class variance does not exist for all roles, e.g., "recipient" in Figure 5. Therefore, forcing multiple prototypes results in arguments being forced to optimize towards different prototypes, which in turn increases intra-class variance and negatively impacts the model's decision-making.

Insufficient training. Some roles have fewer samples during training, resulting in insufficient training of the corresponding role prototypes. This leads to the inadequate fitting of arguments and prototypes, and thus, the model affects its performance and accuracy.

H Analysis of Computing Resources

The main challenges in the implementation of HMPEAE lies in determining the number of prototypes for each role and the appropriate number of iterations for the Sinkhorn-Knopp algorithm. For the former, it necessitates in-depth analysis of the data and conducting relevant experimental validations to select the most suitable number. We discuss the performance impact of setting different numbers of prototypes and the potential risk of setting too many prototypes in section 3.4; For the latter, in general, more iterations will result in a better matching between arguments and prototypes, but will also result in additional training time.

Additionally, the number of roles K and prototypes M per role also impact the training efficiency of hyperspherical multi-prototypes. Specifically, to incorporate semantic priors into the training process, it is necessary to compare the cosine similarity between prototypes p_i and p_j, p_i and p_k for all triplets (i, j, k), where $i \neq j \neq k, b(i) \neq b(j) \neq b(k)$, i.e., $d(p_i, p_j)$ and $d(p_i, p_k)$, to obtain L_{rank} in Equ.(10). In this case, if K and M are large, more triplets will be formed, leading to increased computational overhead, additional training time, and memory usage. In the two datasets used in this paper, training role prototypes for the RAMS dataset (66 roles and two prototypes for each role) only required 3 minutes and 2.7GB of memory space, and training role prototypes for the WikiEvents dataset (81 roles and three prototypes for each role) only required 5 minutes and 3.4GB of memory space.

I Analysis of Applicability and Scalability

We find that our proposed Hyperspherical Multi-Prototype(HMP) method exhibits significant scalability. Therefore, it is necessary to analyze the scalability and applicability of the method to facilitate its extension to other tasks.

The design of HMP is based on two inductive biases: large margin separation between classes and semantic differences within classes. Inductive biases are crucial for the design of machine learning algorithms, which can be seen as prior assumptions and are validated through experiments. For instance, the design of K-Nearest Neighbor(KNN) is based on the assumption that neighboring samples in the feature space tend to belong to the same class, while Word2Vec(Mikolov et al., 2013) is designed based on the assumption that a word's meaning is given by the words that frequently appear close-by.

In the context of categorization, one significant and long-standing inductive bias is optimal class separation. A prominent example of utilizing this bias is Support Vector Machines(SVM), which work by maximizing the margin of the hyperplane between samples of two class. Thus for the first inductive bias, we can degenerate HMP into a hyperspherical single-prototype method and use it for the classification tasks. The classification loss $\mathcal{L}_{\mathrm{c}}$ to minimize is given as:

$$\mathcal{L}_{\mathrm{c}} = \sum_{i=1}^{N}(1 - \frac{|\mathbf{x}_i \cdot \mathbf{p}_{y_i}|}{||\mathbf{x}_i||||\mathbf{p}_{y_i}||})^2, \tag{30}$$

where $\mathbf{x}_i$ is the features embeddings of i-th training example, $\mathbf{p}_{y_i}$ is the corresponding golden class prototype and N is the number of training examples.

For the second inductive bias, it has already been used in designing models for Named Entity Recognition(NER)(Wu et al., 2023) and has achieved better performance. However, it is hard to determine in advance whether a dataset for a classification task has semantically differences within one class. This may require an preliminary assessment using data mining or clustering algorithms, or confirming the existence of inductive bias through designing models and validating it experimentally.