HiURE: Hierarchical Exemplar Contrastive Learning for Unsupervised Relation Extraction

Anonymous ACL submission

Abstract

Unsupervised relation extraction aims to extract the relationship between entities from natural language sentences without prior information on relational scope or distribution. Existing works either utilize self-supervised schemes to refine relational feature signals by iteratively leveraging adaptive clustering and classification that provoke gradual drift problems, or adopt instance-wise contrastive learning which unreasonably pushes apart those sentence pairs that are semantically similar. To overcome these defects, we propose a novel contrastive learning framework named `HiURE`, which has the capability to derive hierarchical signals from relational feature space using cross hierarchy attention and effectively optimize relation representation of sentences under exemplar-wise contrastive learning. Experimental results on two public datasets demonstrate the advanced effectiveness and robustness of `HiURE` on unsupervised relation extraction when compared with state-of-the-art models.

1 Introduction

Relation Extraction (RE) aims to discover the semantic (binary) relation that holds between two entities from plain text. For instance, "Kissel$_{head}$ was born in Adrian$_{tail}$ ...", we can extract a relation `/people/person/place_of_birth` between the two head-tail entities. The extracted relations could be used in various downstream applications such as information retrieval (Corcoglioniti et al., 2016), question answering (Bordes et al., 2014), and dialog systems (Madotto et al., 2018).

Existing RE methods can achieve decent results with high-quality manually annotated data or human-curated knowledge bases (KBs). While in practice, human annotation can be labor-intensive to obtain and hard to scale up to newly created relations. Lots of efforts are devoted to alleviating the impact of human annotations in relation extraction. Unsupervised Relation Extraction (URE) is especially promising since it does not require any prior information on relation scope and distribution.

The main challenge in URE is how to cluster semantic information of sentences in the relational feature space. Simon et al. (2019) adopted skewness and dispersion losses to enforce relation classifier to be confident in the relational feature prediction and ensure all relation types can be predicted averagely in a minibatch. But it still requires the exact number of relation types in advance, and the relation classifier could not be improved by obtained clustering results. Hu et al. (2020) encoded relational feature space in a self-supervised method that bootstraps relational feature signals by leveraging adaptive clustering and classification iteratively. Nonetheless, like other self-training methods, the noisy clustering results will iteratively result in the model deviating from the global minima, which is also known as gradual drift problem (Curran et al., 2007; Zhang et al., 2016).

Peng et al. (2020) leveraged contrastive learning to obtain a flat metric for sentence similarity in a relational feature space. However, it only considers the relational semantics in the feature space from an instance perspective, which will treat each sentence as an independent data point. As scaling up to a larger corpus with potentially more relations in a contrastive learning framework, it becomes more frequent that sentence pairs sharing similar semantics are undesirably pushed apart in a flat relational feature space. Meanwhile, we observe that many relation types can be organized in a hierarchical structure. For example, the relations `/people/person/place_of_birth` and `/people/family/country` share the same parent semantic on `/people`, which means that they belong to the same semantic cluster from a hierarchical perspective. Unfortunately, these two relations will be pushed away from each other in an instance-wise contrastive learning framework.

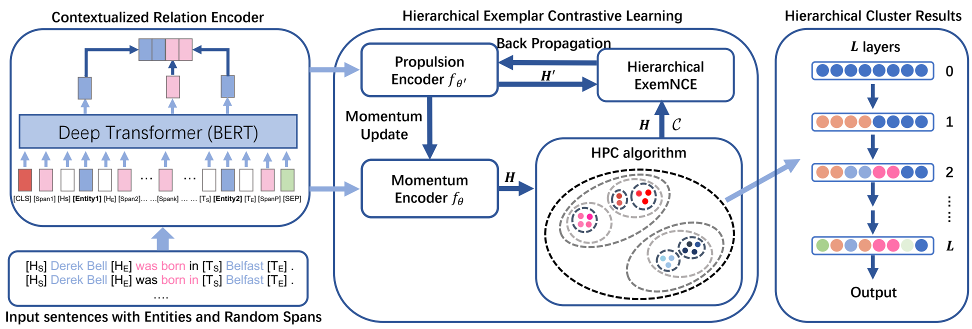


Figure 1: Framework of `HiURE`. Sentence representations will be augmented through Random Spans with fixed Entities, then transmitted into Propulsion and Momentum Encoder respectively. The HPC algorithm takes Momentum feature H as input and generates L layers of clustering results together with L exemplar sets $\mathcal{C}$. HiNCE takes H and H' for instance-wise while H and $\mathcal{C}$ for exemplar-wise contrastive learning.

Therefore, our intuitive approach is to alleviate the dilemma of similar sentences being pushed apart in contrastive learning by leveraging the hierarchical cluster semantic structure of sentences. Nevertheless, traditional hierarchical clustering methods all suffer from the gradual drift problem. Thereby, we try to exploit a new approach of hierarchical clustering by combining propagation clustering and attention mechanism. We first define **exemplar** as a representative instance for a group of semantically similar sentences in certain clustering results. Exemplars can be in different granularities and organized in a hierarchical structure. In order to enforce relational features to be more similar to their corresponding exemplars in all parent granularities than others, we propose `HiURE`, a novel contrastive learning framework for URE which combines both the instance-wise and exemplar-wise learning strategies, to gather more reasonable relation representations and better classification results.

The proposed `HiURE` model is composed of two modules: Contextualized Relation Encoder and Hierarchical Exemplar Contrastive Learning. As shown in Figure 1, the encoder module leverages pre-trained BERT model to obtain two augmented entity-level relational features of each sentence for instance-wise contrastive learning, while the learning module retrieves hierarchical exemplars in a top-down fashion for exemplar-wise contrastive learning and updates the features of sentences iteratively according to the hierarchy. These updated features could be utilized to optimize the parameters of encoders by a combined loss function noted as Hierarchical ExemNCE (HiNCE) in this work. To summarize, the main contributions of this paper are as follows:

- We develop a novel hierarchical exemplar contrastive learning framework `HiURE` that incorporates top-down hierarchical propagation clustering for URE.
- We demonstrate how to leverage the semantic structure of sentences to extract hierarchical relational exemplars which could be used to refine contextualized entity-level relational features via HiNCE.
- We conduct extensive experiments on two datasets and `HiURE` achieves better performance than the existing state-of-the-art methods. This clearly shows the superior capability of our model for URE by leveraging different types of contrastive learning. Our ablation analysis also shows the impacts of different modules in our framework.

2 Proposed Model

The proposed model `HiURE` consists of two modules: Contextualized Relation Encoder and Hierarchical Exemplar Contrastive Learning. As illustrated in Figure 1, the encoder module takes natural language sentences as input, where named entities are recognized and marked in advance, then employs the pre-trained BERT (Devlin et al., 2019) model to output two contextualized entity-level feature sets H and H' for each sentence based on Random Spans. The learning module takes these relational features as input, and aims to retrieve exemplars that represent a group of semantically similar sentences in different granularities, denoted as $\mathcal{C}$. We leverage these exemplars to iteratively update relational features of sentences in a hierarchy and construct an exemplar-wise contrastive learning loss called Hierarchical ExemNCE which enforces the relational feature of a sentence to be more similar to its corresponding exemplars than others.

2.1 Contextualized Relation Encoder

The Contextualized Relation Encoder aims to obtain two relational features from each sentence based on the context information of two given entity pairs for instance-wise contrastive learning. In this work, we assume named entities in the sentence have been recognized in advance.

For a sentence $x = [w_1, .., w_T]$ with T words where each w_i represents a word and two entities Head and Tail are mentioned, we follow the labeling schema adopted in Soares et al. (2019) and argument x with four reserved tokens to mark the beginning and the end of each entity. We introduce $[\mathrm{H_S}]$, $[\mathrm{H_E}]$, $[\mathrm{T_S}]$, $[\mathrm{T_E}]$ to represent the start or end position of head or tail entities respectively and inject them to x:

$$\begin{aligned} x' = [&w_1, ..., [\mathrm{H_S}], ..., w_i, ..., [\mathrm{H_E}], ..., w_{Span1}, ..., \\ &w_{SpanP}, ..., [\mathrm{T_S}], ..., w_j, ..., [\mathrm{T_E}], ..., w_T] \end{aligned} \tag{1}$$

where x' will be the input token sequence for the encoder and $Span$ subscript indicates the Random Span words. Considering the relational features between entity pairs are normally embraced in the context, we use pre-trained BERT (Devlin et al., 2019) model to effectively encode every tokens in the sentence along with their contextual information, and get the token embedding $\mathbf{b}_i = f_{\mathrm{BERT}}(w_i)$, where $i \in [1, T]$ including the special tokens in x' and $\mathbf{b}_i \in \mathbb{R}^{\cdot b_R}$.

We utilize the outputs $\mathbf{b}_i$ corresponding to $[\mathrm{H_S}]$ and $[\mathrm{T_S}]$ as the contextualized entity-level features instead of using sentence-level marker $[CLS]$ to get embedding for target entity pair. For contrastive learning purposes, we randomly select P words as **Random Span** from the context words except for those entity words between special tokens to augment the entity-level features as $\mathbf{b}_{Span}$, where multiple different Random Span selections lead to different semantically invariant embedding of the same sentence. We concatenate them to derive a fixed-length relational feature $\mathbf{h} \in \mathbb{R}^{(2+P)\cdot b_R}$:

$$\mathbf{h} = [\mathbf{b}_{[\mathrm{H_S}]}, \mathbf{b}_{[\mathrm{T_S}]}, \mathbf{b}_{Span1}, ..., \mathbf{b}_{SpanP}] \tag{2}$$

where $\mathbf{h}$ is the output of the Contextualized Relation Encoder which can be denoted as $f_\theta(x, \mathrm{Head}, \mathrm{Tail}, Span)$.

2.2 Hierarchical Exemplar Contrastive Learning

In order to adaptively generate more positive samples other than sentences themselves to introduce more similarity information in contrastive learning, we design hierarchical propagation clustering to obtain multi-level cluster exemplars as positive samples of corresponding instances. We assume the relation hierarchies are tree-structured and define hierarchical exemplars as representative relational features for a group of semantically similar sentences with different granularities. The exemplar-wise contrastive learning encourages relational features to be more similar to their corresponding exemplars than other exemplars.

Algorithm 1 Hierarchical Propagation Clustering

Input: Encoder outputs $H = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n\}$,
Hierarchical cluster layers L
Output: Hierarchical clusters results $\mathcal{C}$

$H^1 \leftarrow H,\ \mathcal{C} \leftarrow [\,]$
Initialize $\{s_{ij} | i, j \in [1, n]\}$ by Eq. 3
$\forall i \neq j:\ p_\top = \min(s_{ij}),\ p_\perp = \mathrm{median}(s_{ij})$
$ps = \left\{ p_l \mid p_l = p_\top + \frac{p_\perp - p_\top}{L-1} \cdot (l-1), l \in [1, L] \right\}$
for l in $[1, L]$ **do**
  Update $\{s_{ij}\}$ according to H^l by Eq. 3
  Set diagonal to preference $s_{ii} = p_l$
  for all iterations **do**
    Update $\{r_{ij}\}$ and $\{a_{ij}\}$ by Eq. 4 and 5
    $\hat{\mathbf{c}} = (\hat{c}_1, \dots, \hat{c}_n),\ \hat{c}_i = \mathrm{argmax}_j (a_{ij} + r_{ij})$
    Get exemplar set $E^l = \{\mathbf{e}^l_{\hat{c}_i} | \mathbf{e}^l_{\hat{c}_i} = \mathbf{h}^l_{\hat{c}_i}, \hat{c}_i \in \hat{\mathbf{c}}\}$
    if Changes of E^l have converged **then**
      break
    end if
  end for
  $\mathcal{C}$.add(E^l)
  $H^{l+1} \leftarrow (H^l, E^l)$ by Eq. 8
end for
return $\mathcal{C}$

The process is completed through *Hierarchical Propagation Clustering* (HPC) to generate cluster results of different granularities and *Hierarchical Exemplar Contrastive Loss* (HiNCE) to optimize the encoder. The main procedure of HPC consists of Propagation Clustering and *Cross Hierarchy Attention* (CHA), as is elaborated in Algorithm 1, which will be explained in detail below.

Propagation Clustering

We use propagation clustering to obtain hierarchical exemplars in an iterative, top-down fashion. Traditional clustering methods such as k-means cluster data points into specific cluster numbers, however, these methods could not utilize hierarchical information in the dataset and require the specific cluster number in advance. Propagation clustering possesses the following advantages: 1) It considers all feature points as potential exemplars and uses their mutual similarity to extract potential tree-structured clusters. 2) It neither requires the actual number of target relations in advance nor the distribution of relations. 3) It will not be affected by the quality of the initial point selection.

In practice, propagation clustering exchanges real-valued messages between points until a high-quality set of exemplars and corresponding clusters are generated. Inspired by Frey and Dueck (2007), we adopt similarity s_{ij} to measure the distance between points i and j, responsibility r_{ij} to indicate the appropriateness for j to serve as the exemplar for i and availability a_{ij} to represent the suitability for i to choose j as its exemplar:

$$s_{ij} = -\left\|\mathbf{h}_i - \mathbf{h}_j\right\|^2 \tag{3}$$

$$r_{ij} = s_{ij} - \max_{j' \neq j}\left(s_{ij'} + a_{ij'}\right) \tag{4}$$

$$a_{ij} = \begin{cases} \sum_{i' \neq i} \max\left(0, r_{i'j}\right), j = i \\ \min\left[0, r_{jj} + \sum\limits_{i' \notin \{i,j\}} \max\left(0, r_{i'j}\right)\right], j \neq i \end{cases} \tag{5}$$

where r_{ij} and a_{ij} will be updated through the propagation iterations until convergence (Lines 8-15) and a set of cluster centers , which is called exemplar, will be chosen as E (Line 11). Then we wish to find a set of L consecutive layers of clustering, where the points to be clustered in layer l are closer to the corresponding exemplar of layer $l-1$. We perform propagation clustering L times (Lines 5-18) with different preferences (Lines 2-4) to generate L different layers of clustering result, where a larger preference leads to more numbers of clusters (Moiane and Machado, 2018). The Hyperparameter Analysis part provides a detailed explanation about how to select L and the reason for building the preference sequence ps according to the formula in Line 4.

Cross Hierarchy Attention

The traditional hierarchical clustering method either merge fine-grained clusters into coarse-grained one or split coarse cluster into fine-grained ones, which will both cause the problem of error accumulation. Preference sequence leads to cluster results with the hierarchical number in propagation clustering but lost the interaction information between adjacent levels. Based on this intuition, we introduce CHA mechanism to leverage signals from coarse-grained exemplar to fine-grained clusters.

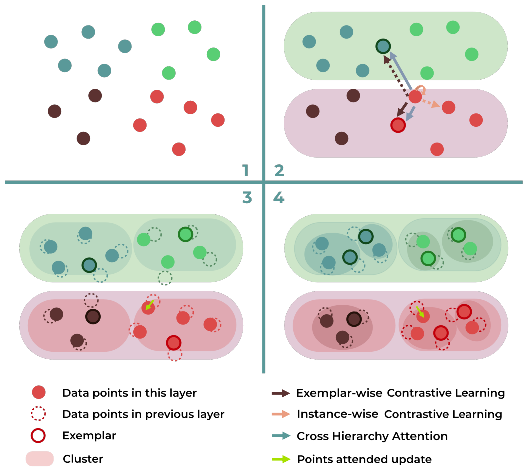


Figure 2: Overview of cross hierarchy attention. The first part shows original data. The second part divides data points into two clusters and utilizes attention to update every points which contribute to the next level of clustering. The dotted line indicates negative sample pair while solid line with positive in contrastive learning.

Formally, we derive a CHA matrix A^l at layer l where the element at (j,k) is obtained by a scaled softmax:

$$\alpha_{jk}^l = \frac{exp(\lambda \mathbf{e}_j^l \cdot \mathbf{e}_k^l)}{\sum_{k'} exp(\lambda \mathbf{e}_j^l \cdot \mathbf{e}_{k'}^l)} \tag{6}$$

where λ is a trainable scalar variable, not a hyper-parameter (Luong et al., 2015). The attention weight α_{jk}^l reflects the proximity between exemplar j and exemplar k in layer l and measures the influence and interactions to corresponding data points between these exemplars. Typically, exemplars that are visually close to each other would have higher attention weights. Then we derive attended point representation at layer $l+1$ by taking the attention weighted sum of its corresponding exemplar from other exemplars:

$$\hat{\mathbf{h}}_i^{l+1} = \sum_k \alpha_{jk}^l \mathbf{e}_k^l \tag{7}$$

where $\mathbf{e}_j^l$ is the exemplar of $\mathbf{h}_i^l$. The attended representation aggregates signals from other exemplars weighted by how close they are to exemplar $\mathbf{e}_j^l$ and transfer the signals from layer l to $l+1$. They reflect how likely a neighboring cluster is relevant or the point will get close to it. Then we combine the attended representation with the original one to obtain the CHA based embedding $\mathbf{h}_i^{l+1}$, defined as:

$$\mathbf{h}_i^{l+1} = \mathbf{h}_i^l + \lambda_{att} \hat{\mathbf{h}}_i^{l+1} \tag{8}$$

where λ_{att} is not a hyper-parameter, but a weighting variable to be automatically trained. As illustrated in Figure 2, the CHA mechanism helps data points to get closer with corresponding exemplars in previous layer and thus perform better in the current layer.

Hierarchical Exemplar Contrastive Loss Given a training set $X = \{x_1, x_2, ..., x_n\}$ of n sentences, Contextualized Relation Encoder can obtain two augmented relational features for each input sentences by randomly sampling spans twice for the same entity pair. We do this for all sentences and obtain $H = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n\}$ and $H' = \{\mathbf{h}'_1, \mathbf{h}'_2, ..., \mathbf{h}'_n\}$. Traditional instance-wise contrastive learning treats two features as a negative pair as long as they are from different instances regardless of their semantic similarity. It updates encoder by optimizing InfoNCE (Oord et al., 2018; Peng et al., 2020):

$$\mathcal{L}_{\text{InfoNCE}} = \sum_{i=1}^{n} -\log \frac{\exp(\mathbf{h}_i \cdot \mathbf{h}'_i/\tau)}{\sum_{j=1}^{J} \exp(\mathbf{h}_i \cdot \mathbf{h}'_j/\tau)} \tag{9}$$

where $\mathbf{h}_i$ and $\mathbf{h}'_i$ are positive samples for instance i, while $\mathbf{h}'_j$ includes one positive sample and $J-1$ negative samples for other sentences, and τ is a temperature hyper-parameter (Wu et al., 2018).

Compared with the traditional instance-wise contrastive learning which unreasonably pushes apart many negative pairs that possess similar semantics, we employ the inherent hierarchical structure in relations. As illustrated in Figure **??**, we perform the HPC algorithm iteratively at each epoch to utilize hierarchical relational features. Note that the relational feature $\mathbf{h}_i$ will be updated in each batch while training, but the exemplars will not be retrieved until the epoch is finished. To maintain the invariance of exemplars and avoid representation shift problems with the relational features in an epoch, we need to smoothly update the parameters of the encoder to ensure a fairly stable relational feature space. In practice, we construct two encoders: Momentum Encoder f_θ and Propulsion Encoder $f_{\theta'}$, both of which is a instance of the Contextualized Relation Encoder. θ' is updated by contrastive learning loss and θ is a moving average of the updated θ' to ensure a smoothly update of relational features (He et al., 2020). We leverage HPC on the momentum features $\mathbf{h}_i = f_\theta(x_i)$ to obtain $\mathcal{C}$ (Line 19), which contains L layers of cluster results with c_l exemplars respectively, where c_l is the number of clusters at layer l. In order to enforce the relational features more similar to their corresponding exemplars compared to other exemplars (Caron et al., 2020; Li et al., 2020), we define exemplar-wise contrastive learning as ExemNCE:

$$\mathcal{L}_{\text{ExemNCE}} = -\sum_{i=1}^{n} \frac{1}{L} \sum_{l=1}^{L} \log \frac{\exp(\mathbf{h}_i \cdot \mathbf{e}^l_j/\tau)}{\sum_{q=1}^{c_l} \exp(\mathbf{h}_i \cdot \mathbf{e}^l_q/\tau)} \tag{10}$$

where $j \in [1, c_l]$ and $\mathbf{e}^l_j$ is the corresponding exemplar of instance i at layer l. As we have explicitly constrained $\mathbf{h}_i$ and $\mathbf{e}^l_j$ into approximate feature space, so the temperature parameter τ can be shared here. The difference between InfoNCE and ExemNCE is described in the second part of Figure 2, where the solid line represents positive while the dashed line represents negative.

Furthermore, we add InfoNCE loss to retain the local smoothness which could help propagation clustering. Overall, our objective named Hierarchical ExemNCE is defined as:

$$\mathcal{L}_{\text{HiNCE}} = \mathcal{L}_{\text{InfoNCE}} + \mathcal{L}_{\text{ExemNCE}} \tag{11}$$

After we update Propulsion Encoder $f_{\theta'}$ with HiNCE, the Momentum Encoder f_θ can be propulsed by:

$$\theta \leftarrow m \cdot \theta + (1-m) \cdot \theta' \tag{12}$$

where $m \in [0, 1)$ is a momentum coefficient. The momentum update in Eq. 12 makes θ evolve more smoothly than θ' especially when m is closer to 1.

3 Experiments

We conduct extensive experiments on real-world datasets to prove the effectiveness of our model for Unsupervised Relation Extraction tasks and give a detailed analysis of each module to show the advantages of `HiURE`. Implementation details and evaluation metrics are illustrated in Appendix A and B respectively.

3.1 Datasets

Following previous work (Simon et al., 2019; Hu et al., 2020; Tran et al., 2020), we employ NYT+FB to train and evaluate our model. The NYT+FB dataset is generated via distant supervision, aligning sentences from the New York Times corpus (Sandhaus, 2008) with Freebase (Bollacker et al., 2008) triplets. We follow the setting in Hu et al. (2020); Tran et al. (2020) and filter out sentences with non-binary relations. We get 41,685 labeled sentences containing 262 target relations (including *no_relation*) from 1.86 million sentences.

There are two more further concerns when we use the NYT+FB dataset, which is also raised by

Dataset	Model	B^3			V-measure			ARI
		F1	Prec.	Rec.	F1	Hom.	Comp.	
NYT+FB	rel-LDA(Yao et al., 2011)	29.1±2.5	24.8±3.2	35.2±2.1	30.0±2.3	26.1±3.3	35.1±3.5	13.3±2.7
	March(Marcheggiani and Titov, 2016)	35.2±3.5	23.8±3.2	67.1±4.1	27.0±3.0	18.6 ±1.8	49.6±3.1	18.7±2.6
	UIE-PCNN(Simon et al., 2019)	37.5±2.9	31.1±3.0	47.4±2.8	38.7±3.2	32.6±3.3	47.8±2.9	27.6±2.5
	UIE-BERT(Simon et al., 2019)	38.7±2.8	32.2±2.4	48.5±2.9	37.8±2.1	32.3±2.9	45.7±3.1	29.4±2.3
	SelfORE(Hu et al., 2020)	41.4±1.9	38.5±2.2	44.7±1.8	40.4±1.7	37.8±2.4	43.3±1.9	35.0±2.0
	EType(Tran et al., 2020)	41.9±2.0	31.3±2.1	63.7±2.0	40.6±2.2	31.8±2.5	56.2±1.8	32.7±1.9
	MORE(Wang et al., 2021)	42.0±2.2	43.8±1.9	40.3±2.0	41.9±2.1	40.8±2.2	43.1±2.4	35.6±2.1
	OHRE(Zhang et al., 2021)	42.5±1.9	32.7±1.8	60.7±2.3	42.3±1.8	34.8±2.1	53.9 ±2.5	33.6±1.8
	EIURE(Liu et al., 2021)	43.1±1.8	48.4±1.9	38.8±1.8	42.7±1.6	37.7±1.5	49.2±1.6	34.5±1.4
	`HiURE` w/o ExemNCE	40.2±1.4	37.4±1.6	43.5±1.5	39.5±1.6	34.2±1.7	46.7±1.6	32.9±1.1
	`HiURE` w/o HPC	41.4±1.2	38.7±1.0	44.3±0.9	41.5±1.3	37.2±1.1	47.0±0.8	34.3±0.9
	`HiURE` w. 10 clusters	44.3±0.5	39.9±0.6	49.8±0.5	44.9±0.4	40.0±0.5	51.2±0.4	38.3±0.6
	`HiURE`	**45.3±0.6**	40.2±0.7	51.8±0.6	**45.9±0.5**	40.0±0.6	53.8±0.5	**38.6±0.7**
TACRED	rel-LDA(Yao et al., 2011)	35.6±2.6	32.9±2.5	38.8±3.1	38.0±3.5	33.7±2.6	43.6±3.7	21.9±2.6
	March(Marcheggiani and Titov, 2016)	38.8±2.9	35.5±2.8	42.7±3.2	40.6±3.1	36.1±2.7	46.5±3.2	25.3±2.7
	UIE-PCNN(Simon et al., 2019)	41.4±2.4	44.0±2.7	39.1±2.1	41.3±2.3	40.6±2.2	42.1±2.6	30.6±2.5
	UIE-BERT(Simon et al., 2019)	43.1±2.0	43.1±1.9	43.2±2.3	49.4±2.1	48.8±2.1	50.1±2.5	32.5±2.4
	SelfORE(Hu et al., 2020)	47.6±1.7	51.6±2.0	44.2±1.9	52.1±2.2	51.3±2.0	52.9±2.3	36.1±2.0
	EType(Tran et al., 2020)	49.3±1.9	51.9±2.1	47.0±1.8	53.6±2.2	52.5±2.1	54.8±1.9	35.7±2.1
	MORE(Wang et al., 2021)	50.2±1.8	56.9±2.2	44.9±1.8	57.4±2.1	56.7±1.8	58.1±2.3	37.3±1.9
	OHRE(Zhang et al., 2021)	51.8±1.6	55.2±2.1	48.7±1.7	56.4±1.8	55.5±1.9	57.3±2.1	38.0±1.7
	EIURE(Liu et al., 2021)	52.2±1.4	57.4±1.3	47.8±1.5	58.7±1.2	57.7±1.4	59.7±1.7	38.6±1.1
	`HiURE` w/o ExemNCE	47.3±1.1	51.2±1.2	43.9±0.9	56.4±1.0	50.3±1.2	64.2±1.4	36.9±1.0
	`HiURE` w/o HPC	48.4±0.9	50.3±0.8	46.7±1.2	58.1±1.1	51.8±1.4	66.2±1.5	37.8±0.8
	`HiURE` w. 10 clusters	55.8±0.4	57.8±0.3	54.0±0.5	59.7±0.6	57.6±0.5	61.9±0.6	40.5±0.4
	`HiURE`	**56.7±0.4**	58.4±0.5	55.0±0.3	**61.3±0.5**	59.5±0.6	63.1±0.4	**42.2±0.5**

Table 1: Quantitative performance evaluation on two datasets.

Tran et al. (2020). Firstly, the development and test sets contain lots of wrong/noisy labeled instances, where we found that more than 40 out of 100 randomly selected sentences were given the wrong relations. Secondly, the development and test sets are part of the training set. Even under the setting of unsupervised relation extraction, this is still not conducive to reflect the performance of models on unseen data. Therefore, we follow Tran et al. (2020) and additionally evaluate all models on the test set of TACRED (Zhang et al., 2017), a large-scale crowd-sourced relation extraction dataset with 42 relation types (including *no_relation*) and 18,659 relation mentions in the test set.

3.2 Baselines

We use standard unsupervised evaluation metrics for comparisons with other eight baseline algorithms: 1) **rel-LDA** (Yao et al., 2011), generative model that considers the unsupervised relation extraction as a topic model. We choose the full rel-LDA with a total number of 8 features for comparison. 2) **MARCH**(Marcheggiani and Titov, 2016) proposed a discretestate variational autoencoder (VAE) to tackle URE. 3) **UIE** (Simon et al., 2019) trains a discriminative RE model on unlabeled instances by forcing the model to predict each relation with confidence and encourages the number of each relation to be predicted on average, where two base models (UIE-PCNN and UIE-BERT) are considered. 4) **SelfORE** (Hu et al., 2020) is a self-supervised framework that clusters self-supervised signals generated by BERT adaptively and bootstraps these signals iteratively by relation classification. 5) **EType** (Tran et al., 2020) uses one-hot vector of the entity type pair to ascertain the important features in URE. 6) **MORE** (Wang et al., 2021) utilizes deep metric learning to obtain rich supervision signals from labeled data and drive the neural model to learn semantic relational representation directly. 7) **OHRE** (Zhang et al., 2021) proposed a dynamic hierarchical triplet objective and hierarchical curriculum training paradigm for open relation extraction. 8) **EIURE** (Liu et al., 2021) is the state-of-the-art method that intervenes on the context and entities respectively to obtain the underlying causal effects of them.

3.3 Results

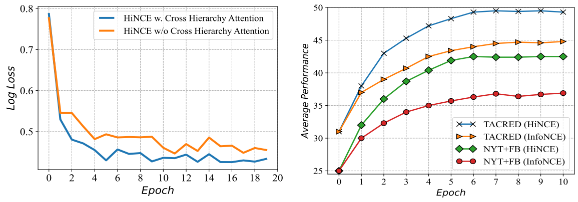
Since most baseline methods adopted the setting by clustering all samples into 10 relation classes (Simon et al., 2019; Hu et al., 2020; Tran et al., 2020; Liu et al., 2021), we adjust the $p_{\perp}$ in Algorithm 1 to get the same results for fair comparison, and name this setting HiURE w. 10 clusters. Although 10 relation classes are lower than the number of true relation types in the dataset, it still reveals important insights about models' ability to tackle skewed distribution.

Table 1 demonstrates the average performance and standard deviation of the three runs of our model in comparison with the baselines on NYT+FB and TACRED. We can observe that EIURE achieves the best performance among all the baselines, which is considered as the previous state-of-the-art method. The proposed HiURE outperforms all baseline models consistently on B^3 F1, V-measure F1, and ARI. HiURE on average achieves 3.4% higher in B^3 F1, 2.9% higher in V-measure F1, and 3.9% higher in ARI on two datasets when comparing with EIURE. The standard deviation of HiURE is particularly lower than other baseline methods, which validates its robustness. Furthermore, the performance of HiURE on TACRED exceeds all the baseline methods by at least 2.1%. These performance gains are likely from both 1) higher-quality manually labeled samples in TACRED and 2) an improved discriminative power of HiURE considering the variation and semantic shift from NYT+FB to TACRED.

Effectiveness of HPC. HPC considers all data points and uses their mutual similarity to find the most suitable points as exemplars for each cluster, these exemplars could update the instances in their own clusters and transfer the relational features from high-level relations to base-level through the cross hierarchy attention. From Table 1, HiURE w/o HPC, which uses k-means instead of the proposed hierarchical clustering, gives 4.7% less performance in average over all metrics when comparing with HiURE.

To intuitively show how tree-structured hierarchical exemplars helps learn better contextualized relational features for URE, we visualize the feature space $\mathbb{R}^{4 \cdot h_R}$ after dimension reduction using t-SNE (Maaten and Hinton, 2008) in Appendix C.

Effectiveness of Cross Hierarchy Attention. In order to explore how CHA helps data points to obtain the semantics of exemplars as training signals in HPC, Figure 3(a) illustrates the log loss values of HiNCE during the training epochs. Based on the loss curve, using Cross Hierarchy Attention leads to consistently lowered loss value, which implies that it provides high-quality signals to help train a better relational clustering model.



(a) Effect of CHA (b) Effect of HiNCE

Figure 3: Effect of Cross Hierarchy Attention on NLL loss on NYT+FB dataset (left) and HiNCE on average performance of two datasets (right) while training.

Considering that our exemplars correspond to specific data points and relations, we further show the hierarchical relations the model derived from the dataset. From Figure 4, we can observe a three-layer exemplars structure the model derives from the NYT+FB dataset without any prior knowledge. The high-level relations and base-level relations belonging to an original cluster convey similar relation categories, which demonstrates the rationality of exemplars in relational feature clustering. As the number of exemplars between different layers increases, some exemplars are adaptively replaced with more fine-grained ones in the sub-layer.

Effectiveness of HiNCE. The main purpose of HiNCE is to leverage exemplar-wise contrastive learning in addition to instance-wise. HiNCE avoids the pitfall where many instance-wise negative pairs share similar semantics but are undesirably pushed apart in the feature space. We first conduct an ablation study to demonstrate the effectiveness of this module. From Table 1, HiURE w/o HiNCE gives us 6.3% less performance averaged over all metrics. Then we report the average performance on B^3 F1, V-measure F1, and ARI on the two datasets changing with epochs, which reflects the quality and purity of the clusters generated by HiURE. From Figure 3(b), compared to InfoNCE alone, training on HiNCE can improve the performance as training epochs increase, indicating that better representations are obtained to form more semantically meaningful clusters.

Hyperparameter Analysis. We have explicitly introduced two hyperparameters P in the encoder and L in the HPC algorithm. We first study the number of $[Span]$ words P which affects the fixed-length of relation representation in Eq. 2 by changing P from 1 to 4 and report the average performance on NYT+FB and TACRED. From Table 2, the fluctuation results indicate that both information deficiency and redundancy of relation representations will affect the model's performance. Using

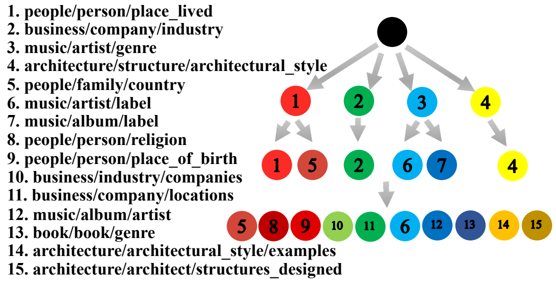


Figure 4: Relation hierarchy derived from the feature space on the NYT+FB dataset.

Dataset / P	1	2	3	4	5
NYT+FB	40.9	**42.5**	41.3	40.6	39.2
TACRED	51.2	**52.4**	51.4	50.6	49.8
Dataset / L	2	3	4	5	3+M
NYT+FB	38.6	**42.5**	40.9	39.2	**42.5**
TACRED	48.8	**52.4**	50.4	49.6	52.1

Table 2: Average performance with different number of P and L on NYT+FB and TACRED.

short $[Span]$ words will introduce less-information relational features so that is hard to transfer representations from a large scale of sentences, while long $[Span]$ words will cause high computational complexity and lead to information redundancy.

Then, we study the level of L hierarchical layers as well as the way of building preference sequence to form them, so as to discover the most suitable tree-structured hierarchical relations for the data distribution. We change L from 2 to 5 with fixed top preference $p_\top$ and bottom preference $p_\perp$ to get the effect of L and report the average performance in Table 2. The fluctuation here implies that fewer layers fail to transfer more information while more layers may cause exemplar-level information conflicts between different coarse-grained layers. (Moiane and Machado, 2018) has shown that the minimum and median value of similarity matrix are best preferences for propagation clustering, so we manually adjust the preference sequence between them multiple times with $L = 3$ and get the average results as `3+M` to compare with the automatically generated ones by Line 3-4 in HPC. The results show that the bottom layer is not so sensitive to the preference sequence as long as it is reasonable, which proves the practicability and effectiveness of the equation in Line 4.

4 Related Work

Unsupervised relation extraction has received attention recently (Simon et al., 2019; Tran et al., 2020; Hu et al., 2020), due to the ability to discover relational knowledge without access to annotations or external resources. Unsupervised models either 1) cluster the relation features extracted from the sentence, or 2) make more assumptions as learning signals to discover better relational representations.

Among clustering models, an important milestone is the self-supervised learning approach (Wiles et al., 2018; Caron et al., 2018; Hu et al., 2020), assuming the cluster assignments as pseudo-labels and a classification objective is optimized. However, these works heavily rely on a frequently re-initialized linear classification layer which interferes with representation learning. Zhan et al. (2020) proposes Online Deep Clustering that performs clustering and network update simultaneously rather than alternatingly to tackle this concern, however, the noisy pseudo labels still affect feature clustering when updating the network.

Inspired by the success of contrastive learning in computer vision tasks (He et al., 2020; Li et al., 2020; Caron et al., 2020), instance-wise contrastive learning in information extraction tasks (Peng et al., 2020), and large pre-trained language models that show great potential to encode meaningful semantics for various downstream tasks (Devlin et al., 2019; Soares et al., 2019), we proposed a hierarchical exemplar contrastive learning schema for unsupervised relation extraction. It has the advantages of supervised learning to capture high-level semantics in the relational features instead of exploiting base-level sentence differences to strengthen discriminative power and also keeps the advantage of unsupervised learning to handle the cases where the number of relations is unknown in advance.

5 Conclusion

In this paper, we propose a contrastive learning framework model `HiURE` for unsupervised relation extraction. Different from conventional self-supervised models which either endure gradual drift or perform instance-wise contrastive learning without considering hierarchical relation structure, our model leverages HPC to obtain hierarchical exemplars from relational feature space and further utilizes exemplars to hierarchically update relational features of sentences and is optimized by performing both instance and exemplar-wise contrastive learning through HiNCE and propagation clustering iteratively. Experiments on two public datasets show the effectiveness of `HiURE` over competitive baselines.

References

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document core f erencing using the vector space model. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 79–85.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33.

Francesco Corcoglioniti, Mauro Dragoni, Marco Rospocher, and Alessio Palmero Aprosio. 2016. Knowledge extraction for information retrieval. In *European Semantic Web Conference*, pages 317–333. Springer.

James R Curran, Tara Murphy, and Bernhard Scholz. 2007. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, volume 6, pages 172–180. Bali.

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

Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science*, 315(5814):972–976.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip Yu. 2020. SelfORE: Self-supervised relational feature learning for open relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3673–3682, Online. Association for Computational Linguistics.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. 2020. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*.

Fangchao Liu, Lingyong Yan, Hongyu Lin, Xianpei Han, and Le Sun. 2021. Element intervention for open relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 4683–4693, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478.

Diego Marcheggiani and Ivan Titov. 2016. Discrete-state variational autoencoders for joint discovery and factorization of relations. *Transactions of the Association for Computational Linguistics*, 4:231–244.

André Fenias Moiane and Álvaro Muriel Lima Machado. 2018. Evaluation of the clustering performance of affinity propagation algorithm considering the influence of preference parameter and damping factor. *Boletim de Ciências Geodésicas*, 24:426–441.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from context or names? an empirical study on neural relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3661–3672.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420.

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

Étienne Simon, Vincent Guigue, and Benjamin Piwowarski. 2019. Unsupervised information extraction: Regularizing discriminative approaches with relation distribution losses. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1378–1387.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.

Thy Thy Tran, Phong Le, and Sophia Ananiadou. 2020. Revisiting unsupervised relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7498–7505, Online. Association for Computational Linguistics.

Yutong Wang, Renze Lou, Kai Zhang, Mao Yan Chen, and Yujiu Yang. 2021. More: A metric learning based framework for open-domain relation extraction. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7698–7702. IEEE.

Olivia Wiles, A Koepke, and Andrew Zisserman. 2018. Self-supervised learning of a facial attribute embedding from video. *arXiv preprint arXiv:1808.06882*.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466. Association for Computational Linguistics.

Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. 2020. Online deep clustering for unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6688–6697.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Kai Zhang, Yuan Yao, Ruobing Xie, Xu Han, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2021. Open hierarchical relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5682–5693.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.

A Implementation Details

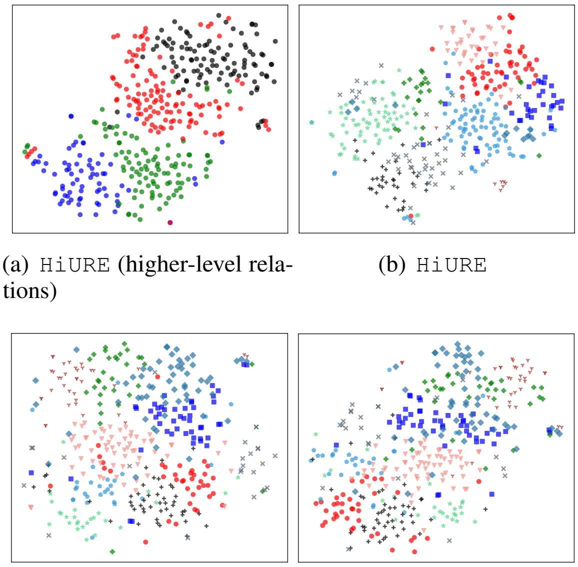
In the encoder phase, we set the number P of randomly selected words in the $[Span]$ to 2, the reason of which is illustrated in parameter analysis. Therefore the output entity-level features $\mathbf{h}_i$ and $\mathbf{h}_i'$ possess the dimension of $4 \cdot b_R$, where $b_R = 768$. We use the pretrained `BERT-Base-Cased` model to initialize both the Momentum Encoder and Propulsion Encoder respectively, and use AdamW (Loshchilov and Hutter, 2017) to optimize the loss. The encoder is trained for 20 epochs with $1e{-}5$ learning rate. In the HPC phase, we set the numbers of layers L to 3 after parameter analysis and the maximum iterations at Line 8 to 400 to make sure the algorithm terminates in time and make the converge condition as E^l not change for 10 iterations. We set temperature parameter $\tau = 0.02$ and momentum parameter $m = 0.999$ following (He et al., 2020) and adjust the number of negative samples J to 512 to accommodate smaller batches.

B Evaluation metrics

We follow previous works and use B^3 (Bagga and Baldwin, 1998), V-measures (Rosenberg and Hirschberg, 2007) and Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) as our end metrics. B^3 uses precision and recall to measure the correct rate of assigning data points to its cluster or clustering all points into a single class. We use V-measures (Rosenberg and Hirschberg, 2007) to calculate homogeneity and completeness, which is analogous to B^3 precision and recall. These two metrics penalize small impurities in a relatively "pure" cluster more harshly than in less pure ones. We also report the F1 value, which is the harmonic mean of Hom. and Comp. Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) measures the similarity of predicted and golden data distributions. The range of ARI is [-1,1]. The larger the value, the more consistent the clustering result is with the real situation.

C Visualization

Visualize Hierarchical Contextualized Features To further intuitively show how tree-structured hierarchical exemplars help learn better contextualized relational features on entity pairs for URE, we visualize the contextual representation space $\mathbb{R}^{(2+P)\cdot b_R}$ after dimension reduction using t-SNE (Maaten and Hinton, 2008). We randomly choose 400 relations from TACRED dataset and the visualization results are shown in Figure 5.



(a) `HiURE` (higher-level relations) (b) `HiURE`

(c) `HiURE` w/o ExemNCE (d) `HiURE` w/o HPC

Figure 5: Visualizing contextualized entity-level features after t-SNE dimension reduction on TACRED dataset.

From Figure 5 (a), we can see that `HiURE` can give proper clustering results to the high-level relational features generated by propagation clustering, and features are colored according to their clustering labels. In order to explore how our modules utilize high-level relation features to guide the clustering of base-level relations, we still use the corresponding high-level clustering relation labels as the color series, while base-level clustering relation labels with different shapes to get Figure 5 (b) (c) (d). `HiURE` in (b) learns denser clusters and discriminitaive features. However, `HiURE` w/o Hierarchical ExemNCE in (c) is difficult to obtain the semantics of the sentences without exemplar-wise information, which makes the clustering results loose and error-prone. When Hierarchical Propagation Clustering is not applied as (d), k-means is adopted to perform clustering on the high-level relational features, which could not use exemplars to update relational features or mutual similarity between feature points. On that occasion, `HiURE` w/o HPC gives the results where the points between clusters are more likely to be mixed. The outcomes revealed above prove the effectiveness of `HiURE` to obtain the semantics of sentences while distinguishing between similar and dissimilar sentences.