SCL: Selective Contrastive Learning for Data-driven Zero-shot Relation Extraction

Ning Pang[1], **Xiang Zhao**[2*], **Weixin Zeng**[2], **Zhen Tan**[1], **Weidong Xiao**[1]
[1]National Key Laboratory of Information Systems Engineering, China
[2]Laboratory for Big Data and Decision,
National University of Defense Technology, China
{pangning14, xiangzhao, zengweixin13, tanzhen08a, wdxiao}@nudt.edu.cn

Abstract

Relation extraction has evolved from supervised relation extraction to zero-shot setting due to the continuous emergence of newly generated relations. Some pioneering works handle zero-shot relation extraction by reformulating it into proxy tasks, such as reading comprehension and textual entailment. Nonetheless, the divergence in proxy task formulations from relation extraction hinders the acquisition of informative semantic representations, leading to subpar performance. Therefore, in this paper, we take a data-driven view to handle zero-shot relation extraction under a three-step paradigm, including encoder training, relation clustering, and summarization. Specifically, to train a discriminative relational encoder, we propose a novel selective contrastive learning framework, namely, SCL, where selective importance scores are assigned to distinguish the importance of different negative contrastive instances. During testing, the prompt-based encoder is employed to map test samples into representation vectors, which are then clustered into several groups. Typical samples closest to the cluster centroid are selected for summarization to generate the predicted relation for all samples in the cluster. Moreover, we design a simple non-parametric threshold plugin to reduce false-positive errors in inference on unseen relation representations. Our experiments demonstrate that SCL outperforms the current state-of-the-art method by over 3% across all metrics.

1 Introduction

Relation extraction (RE) aims to infer the relation between a pair of entities from text, which is a vital step for automatic knowledge graph construction (Zeng et al., 2014). Recent supervised efforts define relation extraction (Zeng et al., 2014, 2015; Zhou et al., 2016) as a multi-class classification task, and select the most appropriate relation label from a pre-defined set of relations for the target entity pair. However, in practice, the training data often fails to cover all relations, and thus supervised RE methods may not be well suited for recognizing unobserved relations during training.

To eliminate labor-intensive annotation for emergent relations, open relation extraction (OpenRE) was first proposed to discover fresh relations by clustering without using any prior knowledge of scope and distribution (Banko et al., 2007; Bollegala et al., 2010). However, OpenRE approaches (Hu et al., 2020; Liu et al., 2022) fail to take advantage of relational knowledge in previously accumulated relations. Confronted with this limitation, zero-shot relation extraction (ZSRE) has come as a remedy to train a model on historical relations with annotated instances and generalize it to extract relations that have never been seen during training (Levy et al., 2017). Since training and testing relations are disjoint under the zero-shot setting, ZSRE models have no access to supervised signals for test relations. Therefore, zero-shot learning represents a long-standing challenge in transferring knowledge from training data to test data (Levy et al., 2017).

To meet the challenge of knowledge transfer, some previous research handles ZSRE by formulating it into a proxy task formulation, such as reading comprehension (Levy et al., 2017), textual entailment (Obamuyide and Vlachos, 2018), and attribute representation matching (Chen and Li, 2021). Since they are required to define answering templates or relation descriptions, searching for suitable and effective options imposes a burden. Additionally, these methods are based on the

[*]Corresponding author.

Transactions of the Association for Computational Linguistics, vol. 12, pp. 1720–1735, 2024. https://doi.org/10.1162/tacl_a_00721
Action Editor: Dan Goldwasser. Submission batch: 3/2024; Revision batch: 7/2024; Published 12/2024.

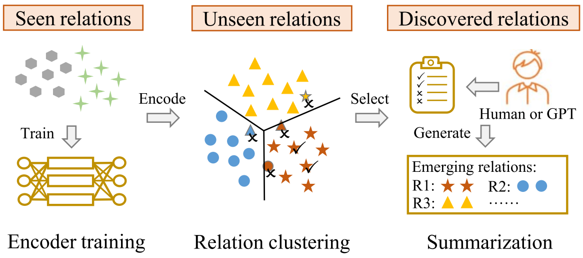


Figure 1: The data-driven paradigm of ZSRE, including seen relation training, unseen relation clustering, and emergent relation discovery.

assumption that humans know what new relations need to be extracted in advance, which is easy to violate in real applications since new relations are always summarized from corresponding emergent instances. That is, we contend that ZSRE models are expected to be data-driven rather than human-guided. As a pioneering work, Wang et al. (2022) have adopted this idea, but fail to comprehensively define how to perform data-driven ZSRE.

In this paper, we further formulate the data-driven paradigm of ZSRE, the workflow of which is depicted in Figure 1. In this formulation, we first train a relational encoder on instances of seen relations and employ it to map test samples into an embedding space. The embeddings of these test samples are clustered into several groups, and high-quality samples within each cluster are chosen for summarization by ChatGPT (OpenAI, 2022), generating a new relation name assigned to the corresponding cluster as the predicted label.

Therefore, the key to solving data-driven ZSRE lies in learning an effective projection function that can map instances of the same relation nearby and instances of different relations far apart in the embedding space. Considering the success of contrastive learning in capturing discriminative representations (Chen et al., 2020; Gao et al., 2021b), Wang et al. (2022) harness an instance-level self-supervised contrastive method to train the relational encoder to better learn subtle differences between instances. However, there are still three drawbacks in existing methods when addressing the aforementioned paradigm of ZSRE:

- **Drawback 1**: Most existing RE models adopt the entity-marker encoder to capture the semantics of an instance, which falls short of employing prompts to elicit relational knowledge embedded within the pre-trained language models.
- **Drawback 2**: Although contrastive learning has been successfully applied to RE, previous work fails to distinguish different contrastive negatives and further improve the generalization ability in encoding unseen relations.
- **Drawback 3**: during inference, test samples are partitioned into several clusters, where each cluster will inevitably contain some false positive samples; however, there have been few research attempts to filter out these false positives.

To address the drawbacks, we design a novel selective contrastive learning framework, SCL, to handle data-driven ZSRE. (1) **At the training stage**, we employ prompt learning (Gao et al., 2021a) and adapt it to the task of ZSRE. By bridging the gap between pre-training objective and downstream task, prompt-tuning can take full advantage of knowledge in pre-trained language models. Additionally, to learn better separation of relations, we propose to perform in-batch selective contrastive learning. For a batch of instances, we augment them with a different prompt template to construct another view as positive contrastive instances. It is noteworthy that we assign selective importance scores for various contrastive instances of an anchor, where we dynamically emphasize hard negatives since they are more useful for learning discriminative representations. (2) **At the test stage**, by following Wang et al. (2022), we use the well-learned model to project all test samples into the representation space and separate them into several clusters by clustering algorithms (Hartigan and Wong, 1979; Malzer and Baum, 2020). High-quality samples around the centroid of each cluster are selected for summarization as the prediction for all instances. Based on the observation that in each cluster, false positives always have a greater distance to their centroid, we propose a post-processing method to detect these false-positive instances. Specifically, we design a simple but effective threshold criterion to determine if a test instance is false positive or not. Previous ZSRE works have never investigated this non-parametric method to improve performance, indicating that the seemingly simple idea is non-trivial.

Contribution. In this paper, we re-investigate the task of ZSRE, and the contributions are at least threefold:

- We propose to jointly train an encoder via prompt learning and selective contrastive learning, which can generate more compact relation representations for better separation of relations.
- A post-processing method is designed to filter out mispredicted instances in each cluster. With a non-parametric threshold criterion, we can detect false positives to improve the precision of our proposed ZSRE model.
- To validate the effectiveness of the proposed solution, we conduct extensive experiments on three real-world datasets, and the superiority of SCL in effectiveness is confirmed throughout the comparison with current ZSRE methods.

2 Related Work

In this section, we review the related work from three perspective: zero-shot relation extraction, relational representation learning, and contrastive learning.

2.1 Zero-shot Relation Extraction

Relation extraction (RE) is defined as determining the relation between two entities given a sentence. To get rid of the cost of annotating training instances for fresh relations, open relation extraction (OpenRE) is proposed (Banko et al., 2007; Bollegala et al., 2010). OpenRE aims to learn a general model of how relations are expressed in a particular language, enabling the extraction of a large set of relational triplets without the need for any human input. Based on human-selected features or automatic features from linguistic parsing techniques, clustering algorithms are employed to cluster instances that describe a particular relation (Bollegala et al., 2010; Liu et al., 2022).

However, since these OpenRE methods fail to take advantage of relational knowledge in previously accumulated relations, recent research efforts resort to exploring ZSRE. Since there are no training instances for test relations, existing methods depend on annotating auxiliary information for input and converting RE into proxy tasks, such as reading comprehension (Levy et al., 2017) and sentence entailment (Obamuyide and Vlachos, 2018).

To make use of knowledge in the pre-trained BERT model directly, Chen and Li (2021) propose a ZS-BERT model to handle ZSRE via attribute representation learning. The representations for instances and relation descriptions can be derived from the BERT model, and ZS-BERT learns to match instance representations with description representations. Given the advances of prompt learning, Xu et al. (2023) design multiple prompt templates for an instance and derive multiple representations. These representations are then fused via an attention mechanism and matched with the representations of relation descriptions.

Similar to OpenRE, we utilize the idea of clustering for discovering novel relations. However, our method differs in that it leverages the historic relational knowledge to aid in distinguishing unknown relations. Unlike existing ZSRE methods, our approach refrains from assuming a priori knowledge of which unknown relations to extract. Instead, we embrace a data-driven paradigm for discovering emerging relations.

2.2 Relational Representation Learning

Deep learning has sparked interest in utilizing neural networks for relation extraction, where a key step involves acquiring effective relational representations from raw text. Early efforts focused on employing various neural networks, including CNN (Zeng et al., 2014) and LSTM (Miwa and Bansal, 2016), to automatically learn the representations of relations. To improve the relational representations, universal schema (Riedel et al., 2013; Verga et al., 2016) jointly embed knowledge bases and textual patterns. During ZSRE, emerging relations do not currently exist within the existing knowledge bases. Consequently, it is not feasible to employ universal schema techniques for jointly learning the representations of these relations that need to be extracted.

With the advance of language models, it becomes standard to fine-tune a BERT-like model with additional layers for downstream tasks. To bridge the gap between pre-training and fine-tuning, the prompt-tuning paradigm is proposed by formulating downstream tasks as a cloze-style task. Fueled by the birth of GPT (Brown et al., 2020), prompt learning is popular among a wide range of natural language processing tasks, such as text classification (Schick et al., 2020) and entity typing (Ding et al., 2022).

Prompt learning is also investigated in the task of RE, which has achieved significant performance improvements over previous methods. Among these approaches, Han et al. (2021) propose a PTR model to creatively apply logic rules and decompose the prompt into several subprompts. To further utilize external knowledge, a KnowPrompt model is proposed to inject entity and relation knowledge into pre-trained language models (Chen et al., 2022). Nevertheless, existing prompt-based RE approaches are tailored for supervised or few-shot relation extraction. In this paper, we adapt prompt learning to learn discriminative relational representations for ZSRE.

2.3 Contrastive Learning

In the field of computer vision, contrastive learning has attracted a lot of attention (He et al., 2020; Chen et al., 2020). The underlying idea of contrastive learning is to pull data points from the same class together and push non-neighboring data points away. Intrinsically, contrastive learning enables instance representations from the neural model to be compact and better separated.

Recently, contrastive learning is also employed to pre-train language models in the field of natural language processing. Inspired by SimCLR (Chen et al., 2020), Gao et al. (2021b) propose a framework named SimCSE for sentence representation learning where positive contrastive pairs are constructed with the use of two independently sampled dropout masks. As a more comprehensive study, ConSERT investigates different data augmentation strategies for learning sentence representations by contrastive learning (Yan et al., 2021). Different from the aforementioned contrastive learning methods, we augment data with different prompt templates to construct contrastive positives. Furthermore, instead of considering each contrastive sample equally important, we assign varying degrees of importance to different negative contrastive samples.

3 Preliminary

This section defines the task of data-driven ZSRE, and then overviews our proposed SCL method.

3.1 Task Definition

Given a piece of text, also known as an *instance* x mentioning a pair of entities (e_h, e_t), relation extraction aims at recognizing the semantic relation r between head entity e_h and tail entity e_t based on the contextual clues.

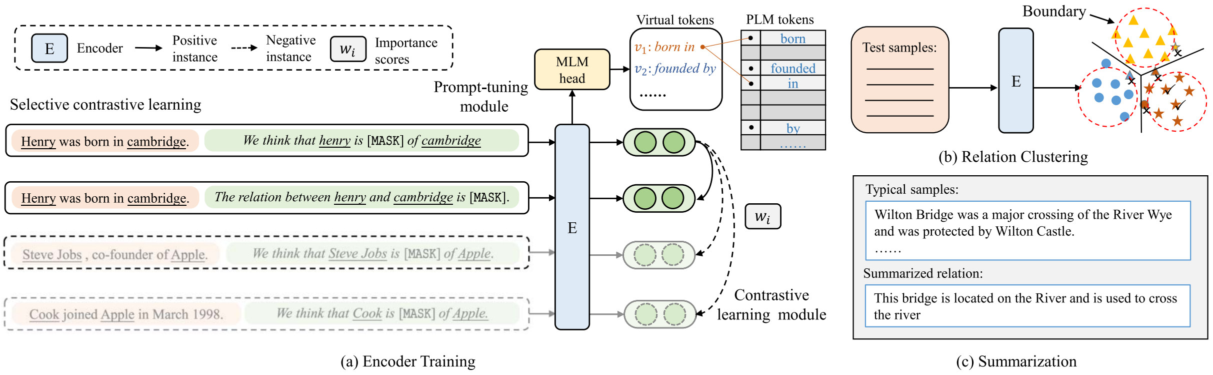
In ZSRE, there are a set of training texts $\mathcal{D} = \{(x_i, r_i)\}_{i=1}^{D}$ (i.e., instances with annotated relation r_i) and a target set of test texts $\mathcal{T} = \{x'_j\}_{j=1}^{T}$, where D (resp. T) is the number of instances in $\mathcal{D}$ (resp. $\mathcal{T}$). $\mathcal{D}$ covers a set of seen relations $\mathcal{R}^s = \{r_1^s, \cdots, r_n^s\}$, while $\mathcal{T}$ covers another set of unseen relations $\mathcal{R}^u = \{r_1^u, \cdots, r_m^u\}$, where n (resp. m) is the cardinality of $\mathcal{R}^s$ (resp. $\mathcal{R}^u$). It should be noted that these two sets of relations are disjoint, i.e., $\mathcal{R}^s \cap \mathcal{R}^u = \emptyset$.

As introduced above, we formally define the approach of solving the ZSRE task in three steps. At the first step, our goal is to train a relation extraction model on the training data $\mathcal{D}$, which consists of an encoder $\mathcal{E}$ and a classifier g, i.e., $g(\mathcal{E}(x_i)) \rightarrow r \in \mathcal{R}^s$. At the second stage, we harness the well-trained encoder $\mathcal{E}$ to map the test set $\mathcal{T} = \{x'_j\}_{j=1}^{T}$ into an embedding space, denoted as $\mathcal{E}(\mathcal{T}) \rightarrow \{\mathbf{r}'_j\}_{j=1}^{T}$. These instance embeddings are clustered into m groups by a cluster algorithm, i.e., $\texttt{Cluster}(\{\mathbf{r}'_j\}_{j=1}^{T}) \rightarrow \{\mathcal{C}_1, \cdots, \mathcal{C}_m\}$. In the third phase, we select some typical samples closest to the centroid of each $\mathcal{C}_i$ for summarization by ChatGPT to generate a relation name, denoted as $\texttt{Summarize}(\mathcal{C}_i) \rightarrow r'$. This generated relation name is then assigned as the label for all instances within that cluster $\mathcal{C}_i$.

3.2 Overview

We propose a new selective contrastive learning framework SCL to solve relation extraction in a zero-shot setting. As depicted in Figure 2, it comprises three key components: A prompt-tuning module, a contrastive learning module, and a relation inference module.

During *training*, we employ a pre-trained language model as the backbone relational encoder. Given a batch of instances concerning two entities, we augment them with two distinct prompt templates and feed them to the language model to generate instance representations. These representations are then sent into the prompt-tuning module and contrastive learning module. For the prompt-tuning module, we expand the language model with a set of extra tokens to represent the seen relations. Prompt-tuning aims to find the most appropriate token to fill into the masked position in the prompt template. For the contrastive learning module, the goal is to find the counterpart of


E
Encoder
Positive instance
Negative instance
w_i
Importance scores
Selective contrastive learning
Prompt-tuning module
MLM head
Virtual tokens
v_1: born in
v_2: founded by
......
PLM tokens
born
founded
in
by
......
Henry was born in cambridge.
We think that henry is [MASK] of cambridge
Henry was born in cambridge.
The relation between henry and cambridge is [MASK].
Steve Jobs , co-founder of Apple.
We think that Steve Jobs is [MASK] of Apple.
Cook joined Apple in March 1998.
We think that Cook is [MASK] of Apple.
E
w_i
Contrastive learning module
Test samples:
E
Boundary
Typical samples:
Wilton Bridge was a major crossing of the River Wye and was protected by Wilton Castle.
......
Summarized relation:
This bridge is located on the River and is used to cross the river

(a) Encoder Training

(b) Relation Clustering

(c) Summarization

Figure 2: The workflow of our SCL when handling zero-shot relation extraction.

each instance between two views. To highlight the role of hard contrastive pairs, we propose to dynamically emphasize these hard pairs, which facilitates the improved discrimination of relations in the representation space. At the *inference* stage, we first employ the well-learned relational encoder to generate instance representations of unseen relations. These representations can be clustered into m groups by clustering algorithms. For each cluster, the top-k samples closest to the centroids are selected for summarization to discover emergent relations as predicted labels. Besides, we observe that false positives tend to be distributed on the boundary of each cluster due to subtle differences between various relations. Therefore, for a target relation, we reassign a smaller cluster boundary to exclude the out-of-distribution instances and reduce the false-positive risk.

4 Methodology

This section introduces our proposed method, namely, SCL, to accomplish zero-shot relation extraction. In the following, we describe our prompt-based encoder in Section 4.1, prompt-tuning module in Section 4.2, contrastive learning module in Section 4.3, and relation inference module in Section 4.4.

4.1 Prompt-based Encoder

In this paper, we employ a pre-trained language model as our backbone relational encoder, denoted as $\mathcal{E}$. Since prompt-tuning shows a powerful ability to facilitate the acquisition of relational knowledge embedded within language models (Chen et al., 2022; Zhang et al., 2022), we adopt the prompt-based encoder and adapt it to ZSRE in this work.

To assist prompt-tuning, we need to formulate RE as a cloze-style masked language prediction task (Han et al., 2021; Chen et al., 2022). To achieve this goal, we augment data samples with an appropriate template $T(\cdot)$ to prompt the instance x, where a [MASK] token is necessarily held in the prompt template. We manually define the prompt template in our model as: [1]

$$T(e_h, e_t) = \text{``We think that } e_h \text{ is } \texttt{[MASK]} \text{ of } e_t\text{''}, \tag{1}$$

where e_h and e_t are head and tail entity mentions. Sequentially, we equip s with the template and generate the prompt input as:

$$\tilde{x}(e_h, e_t) = \text{``}\texttt{[CLS]}\ x\ \texttt{[SEP]}\ T(e_h, e_t)\ \texttt{[SEP]}\text{''}. \tag{2}$$

By feeding $\tilde{x}(e_h, e_t)$ into the pre-trained language model, we can obtain the hidden vector corresponding to the [MASK] position:

$$\mathbf{r} = \mathcal{E}(\tilde{x}(e_h, e_t)). \tag{3}$$

In the following, we use the notation $\mathbf{r}$, which is the hidden vector associated with the [MASK] position, to represent relational representation of an instance x for prompt-tuning and contrastive learning.

4.2 Prompt-tuning Module

In the prompt-tuning module, the language model $\mathcal{E}$ is tasked with predicting which word is appropriate to fill in the [MASK] position for relation

[1] Other templates can also be used in augmentation. Since we do not focus on prompt engineering, we only manually define one template for instance augmentation.

extraction. To predict the target relation based on the relational representation **r** from Equation (3), relational tokens are introduced to represent the relation labels to be detected.

To eliminate labor-intensive verbalizer engineering, we expand $\mathcal{E}$ with a set of learnable virtual relation tokens $\mathcal{V}$ to fully represent the corresponding seen relations $\mathcal{R}^s$. Instead of using a regular verbalizer that maps each relation label to a single label word in the vocabulary, we assume that each virtual token $v_i \in \mathcal{V}$ can describe the implicit semantics of relation $r_i \in \mathcal{R}^s$. To leverage the semantic information in relation r_i, we initialize the virtual token embeddings with the average token embeddings of a relation name. For example, the virtual token v_1 which corresponds to relation ''*born in*'' is initialized as:

$$\mathbf{v}_1 = (\mathbf{Emb}_{born} + \mathbf{Emb}_{in})/2, \tag{4}$$

where $\mathbf{Emb}_{born}$ (resp. $\mathbf{Emb}_{in}$) is the embeddings of token ''born'' (resp. ''in'') in the embedding layer of language model.

The MLM head layer in the language model is used to recover the `[MASK]` token from the set of relation tokens based on the inner product similarity:

$$p(r_i \mid x) = \frac{\exp(\mathbf{v}_i^\top \mathbf{r})}{\sum_{r_j \in \mathcal{R}^s} \exp(\mathbf{v}_j^\top \mathbf{r})}. \tag{5}$$

The MLM head layer corresponds to the relation classifier g mentioned in Section 3.

With access to the training set $\mathcal{D} = \{(x_i, r_i)\}_{i=1}^{D}$, the language model $\mathcal{E}$ is optimized by minimizing the cross-entropy loss:

$$\mathcal{L}_{CE} = \frac{1}{D} \sum_{x_i \in \mathcal{D}} \log p(r_i \mid x_i). \tag{6}$$

4.3 Contrastive Learning Module

In the contrastive learning module, given a batch of instances $\{x_i\}_{i=1}^{B}$, where B is the batch size, following Gao et al. (2021b) and Wang et al. (2022), we need to construct an augmented view for instance-level contrastive learning. Diverging from the prior approaches (Gao et al., 2021b; Wang et al., 2022) that utilize dropout for constructing augmented views, we construct positive contrastive instances by concatenating different prompt templates for each individual instance. Take an instance x concerning (e_h, e_t), for example; its original view is shown in Equation (2), and its augmented view is

$$\hat{x} = \text{``}\texttt{[CLS]}\ x\ \texttt{[SEP]}\ \hat{T}(e_h, e_t)\ \texttt{[SEP]}\text{''}, \tag{7}$$

where $\hat{T}(e_h, e_t)$ is another template manually defined as:

$$\hat{T}(e_h, e_t) = \text{``The relation between } e_h \text{ and } e_t \text{ is } \texttt{[MASK]}\text{''}. \tag{8}$$

By augmenting the batch of instances $\{x_i\}_{i=1}^{B}$ with different prompt templates, we can derive two views of instance representations $\{\mathbf{r}_i\}_{i=1}^{B}$ and $\{\hat{\mathbf{r}}_i\}_{i=1}^{B}$ by Equation (3).

The original instance-level contrastive learning (Gao et al., 2021b; Wang et al., 2022) defines the contrastive learning loss by taking a temperature-scaled cross-entropy objective with in-batch negatives as:

$$\mathcal{L}_{CL} = -\log \frac{\exp(\texttt{sim}(\mathbf{r}_i, \hat{\mathbf{r}}_i)/\tau)}{\sum_{j=1}^{B} \exp(\texttt{sim}(\mathbf{r}_i, \hat{\mathbf{r}}_j)/\tau)}, \tag{9}$$

where τ is the adjustable temperature parameter, B is the batch size, and $\texttt{sim}(\mathbf{r}_1, \mathbf{r}_2) = \frac{\mathbf{r}_1^\top \mathbf{r}_2}{\|\mathbf{r}_1\| \cdot \|\mathbf{r}_2\|}$ is the cosine similarity. In contrast to the original contrastive learning approach, our selective contrastive learning not only takes into account the label signals but also selectively assigns different importance scores to negative samples.

Specifically, for an anchor representation $\mathbf{r}_i$, we only take its counterpart augmented view $\hat{\mathbf{r}}_i$ as the positive contrastive instance. All instances with a different relation in the augmented set are taken as contrastive negatives, denoted as $\{\hat{\mathbf{r}}_k^-\}_{k=1}^{N}$. Mathematically, we measure the importance of a negative contrastive instance by calculating its distance to the anchor, where a smaller distance indicates a harder negative contrastive instance. Therefore, we define the selective importance score assigned to a contrastive instance as:

$$w_{ij} = \frac{N \cdot \exp(-\|\mathbf{r}_i - \hat{\mathbf{r}}_j^-\|)}{\sum_{k=1}^{N} \exp(-\|\mathbf{r}_i - \hat{\mathbf{r}}_k^-\|)}, \tag{10}$$

where $\|\mathbf{r}_i - \hat{\mathbf{r}}_k^-\|$ is the Euclidean distance between $\mathbf{r}_i$ and $\hat{\mathbf{r}}_k^-$. After obtaining the importance scores, we define our selective contrastive loss as:

$$\mathcal{L}_{SCL} = -\log \frac{\exp(\texttt{sim}(\mathbf{r}_i, \hat{\mathbf{r}}_i)/\tau)}{\sum_{j=1}^{N} w_{ij} \cdot \exp(\texttt{sim}(\mathbf{r}_i, \hat{\mathbf{r}}_j^-)/\tau)}. \tag{11}$$

More details about the definitions of our proposed importance score and selective contrastive loss can be seen in Appendix A.

Finally, we optimize the ZSRE model, especially the encoder $\mathcal{E}$ with a joint loss:

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}_{CE} + \lambda_2 \cdot \mathcal{L}_{SCL}, \tag{12}$$

where λ_1 and λ_2 are two weights for different constitute losses.

4.4 Relation Inference Module

At the test phase, we send incoming instances $\mathcal{T} = \{x'_j\}_{j=1}^{T}$ of unseen relations into the well-learned encoder $\mathcal{E}$ to generate their representations $\{\mathbf{r}'_j\}_{j=1}^{T}$. When the number of unseen relations m (defined in Section 3) is unknown, we harness the density-based HDBSCAN algorithm (Malzer and Baum, 2020) to automatically determine the number of clusters and assign samples to corresponding clusters. When the value of m is known in advance, following Wang et al. (2022), we apply the partition-based K-means algorithm to cluster the representations into K groups, where $K = m$. The clustering process can be denoted as $\texttt{Cluster}(\{\mathbf{r}'_j\}_{j=1}^{T}) \rightarrow \{\mathcal{C}_1, \cdots, \mathcal{C}_m\}$, where each cluster $\mathcal{C}_i$ contains a set of test samples.

Since the relation name of each cluster is unknown, we resort to a ChatGPT annotator to summarize a relation r' from typical samples and assign it to the cluster $\mathcal{C}_i$. In our implementation, we select the sample closest to the centroid of each cluster as a typical sample for summarization. Specifically, we generate the relation description by inputting the prompt ''*Based on the text that* $x(e_h, e_t)$*, please summarize the relation between* e_h *and* e_t." into ChatGPT. More details about the generated results can be found in Section 5.7. Test samples within $\mathcal{C}_i$ will be assigned r' as the predicted relation. For the evaluation purpose, we manually map the generated relation name r' onto the most appropriate relation label r^u in the test relation set $\mathcal{R}^u$.

By visualizing in a 2-D space, we find that many false-positive samples are distributed on the edge of a cluster. For the target-predicted relation, these false positives are considered out-of-distribution (OOD) samples. Therefore, the performance of the ZSRE model can be significantly improved by detecting and removing these OOD samples.

When employing the HDBSCAN algorithm, detected noisy points can be considered OOD samples. For the K-means algorithm, we introduce a simple yet effective false-positive detection method with an OOD boundary. Specifically, for an instance in a cluster, we compute its distance to the centroid, and use a simple threshold criterion to determine if it is OOD or not. For i-th cluster $\mathcal{C}_i = \{\bar{\mathbf{r}}_1, \bar{\mathbf{r}}_2, \cdots, \bar{\mathbf{r}}_{|\mathcal{C}|}\}$ where $\bar{\mathbf{r}} = \frac{\mathbf{r}'}{\|\mathbf{r}'\|}$, the cluster centroid is computed as the average of all normalized instance embeddings:

$$\bar{\mathbf{r}}^* = \frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} \bar{\mathbf{r}}_i. \tag{13}$$

The instance-level Euclidean distance to the centroid is computed by:

$$d(\bar{\mathbf{r}}_i, \bar{\mathbf{r}}^*) = \|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}^*\|. \tag{14}$$

The radius of the relation cluster is defined as:

$$d^* = \max_i \{d(\bar{\mathbf{r}}_i, \bar{\mathbf{r}}^*)\}. \tag{15}$$

The decision function for an OOD sample is given by:

$$G(\bar{\mathbf{r}}_i) = \mathbf{1}\{d(\bar{\mathbf{r}}_i, \bar{\mathbf{r}}^*) > \delta \cdot d^*\}, \tag{16}$$

where δ is a hyper-parameter that determines the threshold of OOD boundary and $\mathbf{1}\{\cdot\}$ is an indicator function. The δ is set manually by searching on the validation set, with a detailed discussion provided in Section 5.5.

Remark. Despite the HDBSCAN algorithm not requiring priori knowledge of the number of unseen relations, we opt to consider the K-means algorithm as an alternative method for relation inference. This choice stems from the tendency of HDBSCAN clustering to yield a greater number of clusters than the count of unseen relations in the test data, potentially leading to higher test performance, which may not be entirely fair in comparison to other methods. When using the HDBSCAN algorithm for relation inference, noise points might be assigned to the nearest cluster if OOD sample detection is not applied.

5 Experiments

In this section, we first describe the experimental setup and then present the experiment results with an in-depth analysis.

5.1 Benchmarks and Evaluation Metrics

Benchmarks. Experiments are conducted on three benchmark relation extraction datasets:

FewRel (Gao et al., 2019) is a manually annotated dataset built on texts from Wikipedia, which contains 80 relations with 700 instances per relation. We randomly select 40 relations as seen relations for training, while selecting m relations from the remaining 40 as unseen relations for testing where m varies in $\{20, 30, 35, 40\}$. When conducting testing with 20 relations, the remaining 20 relations can be utilized as a validation set.

Wiki-ZSL (Chen and Li, 2021) is a distantly supervised dataset which was derived by aligning Wiki-KB with texts from Wikipedia. This dataset consists of 113 relations and a total of 94,383 instances. To assist ZSRE, we randomly select the entire samples of 73 relations as training data, while using the remaining 40 relations for testing. We vary the number of unseen relations in $\{20, 30, 35, 40\}$ to evaluate ZSRE models under different settings. Similar to FewRel, when 20 unseen relations are used for testing, the remaining 20 relations can be utilized as a validation set.

TACRED (Zhang et al., 2017) contains 42 relations and was constructed via crowd-sourcing. In our experiments, we use a revised version of TACRED (Cui et al., 2021), which contains 40 relations. Additionally, to reduce the impact of sample imbalance on test performance, we limit the number of instances per relation to 1,000. We keep 20 random relations for training and the remaining 20 relations for testing. We vary the number of unseen relations m in $\{15, 20\}$ for TACRED. When 15 unseen relations are used for testing, the remaining 5 relations can serve as a validation set.

Evaluation Metrics. The data-driven ZSRE in our work aims to assist humans in discovering novel relations. Since the clustering algorithm can only generate one virtual label for each cluster, it is necessary to assign a real relation name to each cluster for evaluation, which needs to compare predicted labels with ground truths. To convert the pseudo labels predicted by clustering into real relation labels, we select typical samples near the clustering centroids and employ a ChatGPT-based annotation method to generate relation names. By setting the number of typical samples to 1, we ensure that the workload for summarization would be similar to or even smaller than previous methods (Obamuyide and Vlachos, 2018; Chen and Li, 2021). After that, for automatic evaluation purposes, we manually select the most appropriate pre-defined relation label in the test set for each cluster according to the corresponding generated relation name.

For evaluation metrics, we utilize the widely used accuracy and standard F1 score for assessing the performance of ZSRE models. To be specific, for each cluster, there are four types of predictions: true positives, true negatives, false positives, and false negatives. If the predicted relation of a test sample matches its ground truth, it is a true prediction; otherwise, it is deemed false. Samples within the OOD boundary are considered positives, while those outside are considered negatives. Accuracy and F1 score can be calculated based on the counts of different predictions. Besides, since our proposal is a cluster-based method, we also employ B^{3}F1 score and normalized mutual information (NMI) to evaluate the effectiveness of model clustering (Wang et al., 2022).

5.2 Model Settings and Baselines

Model Settings. In experiments, we implement SCL and competing methods based on the *Transformers* package[2] (Wolf et al., 2020), where we employ the base version of the pre-trained BERT model (Devlin et al., 2019) as the backbone encoder for them. When the minimum value of m is used in testing, the remaining unseen relations can serve as validation data for hyper-parameter search. When a hyper-parameter combination is found to perform well across all three datasets, we fix these hyper-parameters and use them to test other settings (i.e., test with more unseen relations) without further fine-tuning. This better demonstrates the insensitivity of our method to parameters across different settings.

Specifically, we set the maximum sentence length to 120 for both the FewRel and Wiki-ZSL datasets, while for the TACRED dataset, we set it to 200. The training epoch is set to 4 and we use an Adam optimizer (Kingma and Ba, 2015) with a batch size of 64. The learning rate is set to 1e-5 with a 0.1 weight decay. The scaled temperature τ in contrastive loss is set to 0.05, and we set λ_1 to 1 and λ_2 to 0.2 across all datasets. The δ is set to 0.75. When the number of unseen relations is assumed to be known in advance, we

[2] https://github.com/huggingface/transformers.

Methods	$m = 20$				$m = 30$				$m = 35$				$m = 40$			
	NMI	B^3F1	Acc	F1	NMI	B^3F1	Acc	F1	NMI	B^3F1	Acc	F1	NMI	B^3F1	Acc	F1
CIM	—	—	52.42	45.86	—	—	41.56	34.24	—	—	36.43	32.85	—	—	35.61	30.72
QARE	—	—	55.76	52.49	—	—	45.83	42.75	—	—	43.28	40.71	—	—	40.63	37.95
MTB	82.75	78.76	79.55	75.83	76.82	60.26	68.26	63.58	75.92	58.41	64.58	62.42	76.90	55.87	65.86	64.29
RCL	80.60	72.82	77.21	73.31	77.52	63.53	70.49	65.95	77.50	63.08	71.40	68.30	77.53	63.26	72.41	70.62
ZS-BERT	71.57	60.84	67.84	65.35	63.76	45.78	52.62	48.97	69.22	47.14	55.86	49.35	65.43	43.86	53.55	47.88
RelationPrompt	—	—	76.45	74.35	—	—	67.13	65.89	—	—	64.07	62.37	—	—	64.72	61.88
MultiPrompt	78.96	75.48	76.94	73.49	77.21	64.56	67.82	63.89	77.84	57.15	61.58	60.43	76.92	59.89	63.77	61.97
SCL +K-means	**86.66**	**80.56**	85.31	83.39	**83.90**	**73.60**	**78.50**	**76.73**	**82.33**	**69.75**	73.93	72.13	**81.99**	**67.97**	75.28	**74.18**
w/ OOD	95.21	91.64	88.90	85.12	90.24	81.71	82.48	78.01	89.85	80.88	80.54	76.71	89.70	80.05	82.12	79.21
SCL +HDBSCAN	77.56	64.23	**86.83**	**84.94**	76.18	61.30	77.46	74.31	76.12	61.33	**76.54**	**73.39**	76.16	60.13	**76.28**	73.08
w/ OOD	84.73	76.77	88.11	88.15	82.88	71.91	78.16	77.29	82.92	71.64	77.48	77.15	83.25	71.33	77.39	77.02

Table 1: Performance (%) on FewRel dataset.

Methods	$m = 20$				$m = 30$				$m = 35$				$m = 40$			
	NMI	B^3F1	Acc	F1	NMI	B^3F1	Acc	F1	NMI	B^3F1	Acc	F1	NMI	B^3F1	Acc	F1
CIM	—	—	43.37	40.75	—	—	36.52	32.18	—	—	32.49	29.63	—	—	31.28	26.58
QARE	—	—	47.95	47.08	—	—	38.97	35.31	—	—	34.05	30.84	—	—	35.83	32.49
MTB	70.42	61.39	70.18	68.61	70.66	56.27	68.55	66.41	67.52	52.20	65.31	62.14	65.80	51.58	62.76	59.44
RCL	68.65	57.28	68.75	65.15	69.09	54.83	67.35	63.46	69.14	55.31	68.94	63.69	69.44	53.28	65.43	61.36
ZS-BERT	62.45	50.72	55.24	53.80	54.71	41.08	46.34	43.16	54.96	42.58	48.50	44.70	48.25	38.42	40.73	42.15
RelationPrompt	—	—	67.30	65.85	—	—	62.40	60.07	—	—	57.48	55.10	—	—	54.25	52.90
MultiPrompt	65.48	56.30	60.38	55.80	64.10	54.06	57.68	53.03	60.49	51.20	54.66	53.78	56.48	45.28	50.75	46.40
SCL +K-means	**74.36**	**65.24**	74.13	70.88	**74.72**	**62.61**	70.69	67.35	**75.14**	**62.59**	71.24	66.02	**74.47**	**59.74**	71.56	65.18
w/ OOD	85.63	78.44	83.05	74.13	87.48	80.16	84.59	73.26	85.57	77.06	82.92	70.86	86.70	78.12	84.22	74.73
SCL +HDBSCAN	68.34	48.26	**81.63**	**80.88**	66.66	42.93	**74.32**	**71.37**	66.58	33.50	**75.75**	**73.71**	66.58	28.09	**76.51**	**76.26**
w/ OOD	71.15	51.73	92.08	90.48	74.50	54.27	85.90	78.78	74.15	42.23	85.24	81.00	74.29	38.01	86.72	82.02

Table 2: Performance (%) on Wiki-ZSL dataset.

employ the K-means algorithm for inference, where the hyper-parameter K is set to m for fair comparison with competing methods. Since OOD detection in the relation inference module is a plugin strategy, we analyze it separately in Section 5.5. This strategy is not employed in the experiments conducted in the remaining sections.

Baselines. In experiments, we compare our proposal with a variety of strong baselines. Among the baselines, several approaches transform the task of ZSRE into alternative task formulations. These approaches include a text entailment-based CIM model (Obamuyide and Vlachos, 2018), and a reading comprehension-based QARE model (Levy et al., 2017). Furthermore, two representation-based methods, namely, MTB (Soares et al., 2019) and RCL (Wang et al., 2022), are also incorporated for comparative analysis. These methods learn discriminative representations for relations and employ the strategy of predicting relations through clustering. ZS-BERT (Chen and Li, 2021) is chosen as a competitor that predicts the target relation by identifying the closest description to a given instance. Additionally, we introduce prompt-based techniques as competing methods, leveraging internal knowledge within pre-trained language models for ZSRE. Specifically, these techniques encompass RelationPrompt (Chia et al., 2022) and MultiPrompt (Xu et al., 2023). All experiments were conducted five times, and average results are reported.

5.3 Overall Performance

RQ1: *Does SCL perform better in zero-shot relation extraction than competing methods?*

To answer **RQ1**, we show the overall results of SCL and competing methods on three datasets in Tables 1, 2, and 3, where we test the models with different numbers of unseen relations. Considering both scenarios where the number of unseen relations is known and unknown, we evaluate the performance of our proposed SCL using both the K-means and HDBSCAN algorithms for inference, and showcases the results with a gray shade. Higher evaluation metric values correspond to superior model performance. The optimal result is bolded and the second-best result is indicated with an underline. Additionally, to validate the idea of excluding false positives, we integrate the

Methods	$m = 15$				$m = 20$			
	NMI	B^3F1	Acc	F1	NMI	B^3F1	Acc	F1
CIM	—	—	52.88	38.60	—	—	48.61	35.44
QARE	—	—	55.42	41.09	—	—	52.75	38.20
MTB	68.58	60.72	76.14	63.02	63.24	56.80	71.95	55.61
RCL	70.40	62.89	79.74	66.34	71.46	60.74	78.28	57.25
ZS-BERT	67.34	58.45	73.08	61.25	60.04	52.60	67.59	51.75
RelationPrompt	—	—	74.85	63.60	—	—	68.32	56.04
MultiPrompt	72.08	65.28	80.60	65.46	67.35	54.53	75.48	61.85
SCL +K-means	**78.49**	71.67	85.70	68.81	**81.29**	**77.67**	**83.29**	63.51
w/ OOD	90.83	88.83	87.90	68.31	89.25	86.58	88.54	65.52
SCL +HDBSCAN	77.62	**72.90**	**86.78**	**73.91**	65.23	46.59	83.18	**67.37**
w/ OOD	86.84	84.83	94.10	80.91	75.20	62.14	90.12	75.33

Table 3: Performance (%) on TACRED dataset.

OOD detection plugin into SCL+K-means and SCL+HDBSCAN and evaluate the performance (the ‘‘w/ OOD’’ lines).

From the results in Tables 1, 2, and 3, it can be observed that: (1) CIM, QARE, and ZS-BERT obtain inferior performance since the disparity in task formulations hinders relation extraction models from acquiring informative semantic representations. (2) When evaluating MTB and RCL, we employ their encoders and utilize the K-means algorithm for relation inference. However, the inferior performance compared to our approach indicates that these encoders are unable to capture more discriminative representations of relations. (3) RelationPrompt performs ZSRE in a generative approach and exhibits the poorest performance among all prompt-based methods due to the intrinsic difficulty of its task formulation. Owing to the utilization of multi-prompt learning, MultiPrompt, which employs a nearest-neighbor search to identify the relation, achieves promising performance among baselines. (4) Our proposed SCL with both K-means and HDBSCAN algorithms consistently surpasses the performance of all comparative methods under different settings. When using the HDBSCAN algorithm for relation prediction, it typically achieves optimal performance on F1 and accuracy, but it tends to perform poorly on the NMI and B^3F1 metrics. This is because the HDBSCAN algorithm often generates more clusters than the actual number of unseen relations in the test data. On the FewRel, Wiki-ZSL, and TACRED datasets, HDBSCAN algorithm usually generates over 100, 200, and 40 clusters, respectively, much more than the number of unseen relations. Therefore, it achieves higher accuracy and F1 scores than the K-means algorithm. However, this clustering distribution does not align well with the true data distribution, resulting in lower NMI and B^3F1 scores. (5) When SCL is combined with the OOD detection plugin to remove false positives, all metrics show improvement.

FewRel

Methods	$m = 30$				$m = 40$			
	NMI	B^3F1	Acc	F1	NMI	B^3F1	Acc	F1
NoCon	82.23	69.81	74.73	71.86	81.10	66.66	73.89	70.25
SelfCon	81.49	68.22	73.57	71.66	81.73	67.69	75.11	72.04
SupCon	81.43	68.16	72.61	68.27	81.23	66.59	71.60	67.85
SCL	**83.90**	**73.60**	**78.50**	**76.73**	**81.99**	**67.97**	**75.28**	**74.18**

Wiki-ZSL

Methods	$m = 30$				$m = 40$			
	NMI	B^3F1	Acc	F1	NMI	B^3F1	Acc	F1
NoCon	73.65	61.05	71.65	65.90	74.22	59.33	70.43	64.58
SelfCon	74.07	61.32	**72.07**	65.88	73.83	59.16	68.85	64.28
SupCon	72.45	59.20	69.31	65.87	73.19	58.06	69.20	61.91
SCL	**74.72**	**62.61**	70.69	**67.35**	**74.47**	**59.74**	**71.56**	**65.18**

TACRED

Methods	$m = 15$				$m = 20$			
	NMI	B^3F1	Acc	F1	NMI	B^3F1	Acc	F1
NoCon	77.81	70.60	84.57	67.49	75.96	66.53	82.58	62.73
SelfCon	76.34	69.47	85.35	68.44	75.35	65.55	82.49	61.44
SupCon	**78.58**	71.25	83.84	66.77	76.71	66.51	82.25	62.87
SCL	78.49	**71.67**	**85.70**	**68.81**	**81.29**	**77.67**	**83.29**	**63.51**

Table 4: The experimental results (%) when employing different contrastive learning methods.

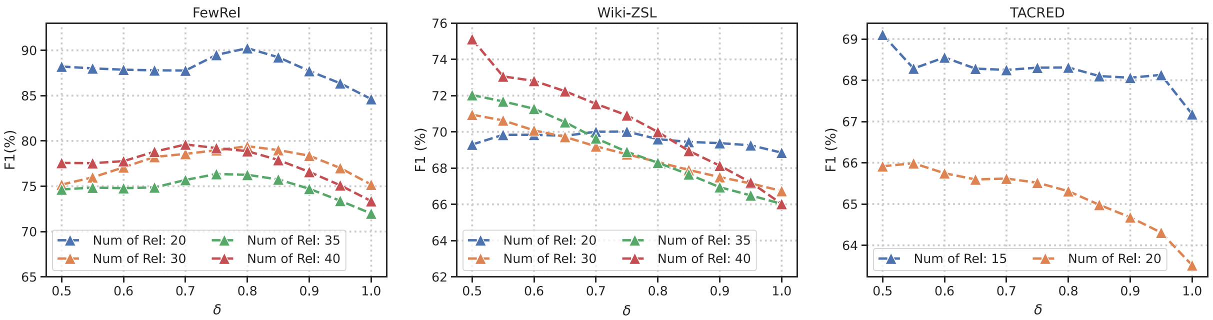
Discussion. When using the K-means algorithm for clustering, it requires prior knowledge of the number of unseen relations. Besides, when using the HDBSCAN algorithm for clustering, it tends to generate more clusters than the number of unseen relations, leading to more labeling efforts. These constitute a limitation of this study. In the following experiments, we harness the K-means algorithm for relation inference.

5.4 Analysis of Contrastive Module

RQ2: *What are the effects of different contrastive learning methods on the performance of ZSRE?*

To further validate our proposed SCL, we conduct an analysis study by replacing the SCL with three variants that combine no contrastive learning (NoCon), self-supervised contrastive learning (SelfCon) (Gao et al., 2021b), and supervised contrastive learning (SupCon) (Khosla et al., 2020). The results are shown in Table 4, where the highest values are represented in bold.

We can conclude from the results in Table 4 that: (1) By comparing SelfCon to NoCon, we observe that almost all indicators of the model have been improved. The improvements signify that additional contrastive learning loss can enhance the representation of instances of unseen relations. (2) Interestingly, compared with SelfCon, SupCon experiences some performance degradation. The underlying cause may be that SupCon on seen relations deteriorates the model's



(a)

(b)

(c)

Figure 3: The F1 scores with the variations of the proportion of preserved samples on three datasets.

ability to generalize to unseen relations. (3) Our proposed SCL removes redundant negative samples and distinguishes true negative examples with different difficulty. These designs can further assist the model in capturing differences between similar samples, thereby enhancing its representation capability for unseen relations. Therefore, the proposed SCL achieves the best performance in nearly all settings.

5.5 Analysis of OOD Detection

RQ3: *What are the effects of different OOD boundaries on the performance of ZSRE?*

To reduce the risk of false-positives, we propose a non-parametric OOD detection method in Section 4.4 to filter out certain false-positive samples. The hyper-parameter δ controls the size of the OOD boundary of each cluster, with samples distributed outside this boundary predicted as false positives. A smaller value of δ corresponds to a smaller cluster radius and excludes more samples. We vary the value of δ in $\{0.50, 0.55, 0.60, \cdots, 0.85, 0.90, 0.95\}$ and observe the changes in the F1 scores of SCL.

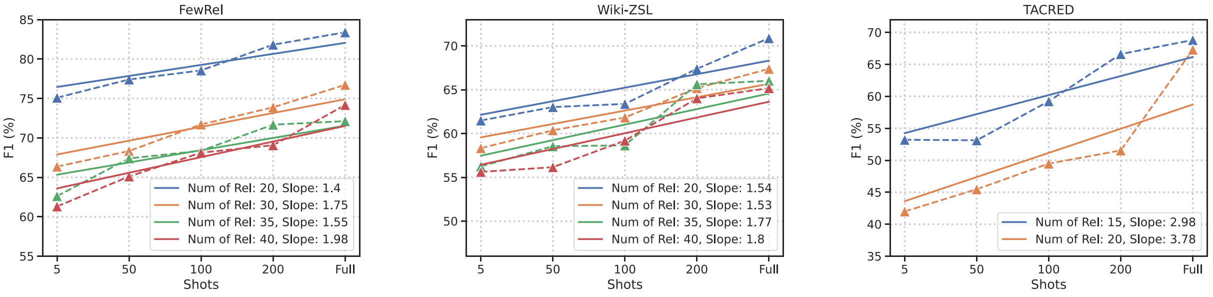
Figure 3 shows that as the value of δ increases, the overall F1 scores decrease for the `Wiki-ZSL` and `TACRED` datasets. This is because larger values of δ are less effective at excluding more false positives, leading to lower precision scores. On the `FewRel` dataset, the F1 scores initially increase and then decreases under four settings. This trend is due to the fact that, although false positives are excluded, some true positive samples that are farther from the cluster centers are also removed, which lowers recall scores. Therefore, it is important to maintain a balance between precision and recall. In our implementation, we found that setting the value of δ to be 0.75 obviously improves the F1 scores across all three datasets.

5.6 Analysis of Data Size

RQ4: *What are the effects of different data sizes of each training relation on the performance of ZSRE?*

This section aims to analyze whether utilizing the entire labeled data for training yields the best generalization capability for separating unseen relations. Therefore, on three datasets, we set the training sample quantity for each seen relation to $\{5, 50, 100, 200, \text{Full}\}$ and observe the performance variations of SCL. The results are presented in Figure 4, where we utilize the Least Squares Method (LSM) to fit the F1 scores to illustrate the trend of performance changes.

From the results, we have the following observations: (1) Based on the linear regression analysis using LSM, as the number of training samples increases, the overall performance of the model exhibits an upward trend. This observation suggests that effectively utilizing the accumulated labeled data of seen relations for training the model plays a crucial role in facilitating the discovery of new relations. (2) Compared to the testing scenarios with 20 unseen relations on the `FewRel` and `Wiki-ZSL` datasets, as well as 15 unseen relations on the `TACRED` dataset, the rate of performance improvement for the model, indicated by the slope of the fitted lines, is higher for the remaining unseen relations in terms of their quantity. This is because as the number of unseen relations increases, the difficulty of testing also intensifies. In such cases, an ample amount of training data becomes particularly vital for the model to learn sufficient semantic information.



(a) (b) (c)

Figure 4: The F1 score variations under different training shots. The triangle markers present the metric values under different settings. Bold lines are the results fitted by LSM.

	Typical samples	ChatGPT annotations	Gold relation label
Case I	*Top-1*: **he scored the most goals in** $[e_h]$ **azerbaijani premier league** $[/e_h]$ **with inter baku in** $[e_t]$ **2007 - 08** $[/e_t]$ **season.** (Distance: 0.4153)	It indicates the specific time or timeframe when the event of winning the prize occurred.	sports season of competition or league
	Top-2: in total, obradović won with panathinaikos, 11 greek league championships, 7 greek cups and 5 $[e_h]$ euroleague $[/e_h]$ titles $[e_t]$ 2011 $[/e_t]$. (Distance: 0.4163)		
	Top-3: e helped the newly promoted $[e_h]$ chinese super league $[/e_h]$ side to the third place in $[e_t]$ 2009 $[/e_t]$, and consistently ranked amongst the competition 's top scorers in the following campaigns. (Distance: 0.4250)		
Case II	*Top-1*: $[e_h]$ **nydalen** $[/e_h]$ **is a rapid transit station on the** $[e_t]$ **ring line** $[/e_t]$ **of the oslo metro.** (Distance: 0.1862)	The association between a specific location and its position within the transportation network	connecting line
	Top-2: $[e_h]$ sinsen $[/e_h]$ is a rapid transit station on the $[e_t]$ ring line $[/e_t]$ of the oslo metro. (Distance: 0.1994)		
	Top-3: the building is directly connected to the $[e_h]$ mcgill station$[/e_h]$ of the montreal metro s $[e_t]$ green line $[/e_t]$. (Distance: 0.2051)		

Table 5: Study cases from FewRel and Wiki-ZSL to show how to generate emergent relation names. In each sample, e_h denotes the head entity while e_t denotes the tail entity.

5.7 Emergent Relation Generation

RQ5: *How can we convert the virtual labels of each cluster into real semantic relation labels?*

Unlike previous zero-shot relation extraction methods that use a predefined set of relations, we derive new relation concepts by clustering samples and summarizing them. Specifically, we select the *Top-1* closest sample to the centroid of each cluster and generate new relations using both ChatGPT (OpenAI, 2022) and human annotation methods. To demonstrate the process of generating relations, we select two cases from the FewRel (Case I) and Wiki-ZSL (Case II) under the setting of 20 unseen relations to showcase the new relations summarized by ChatGPT and gold relation labels in Table 5.

When employing the ChatGPT to summarize, we invoke the API[3] and input the prompt '*'Based on the text that $x(e_h, e_t)$, please summarize the relation between e_h and e_t.*" to generate a relation description. If mentions of entities e_h and e_t exist in the description, we replace these mentions with their entity types to make the definition more general. According to the generated relation name, we manually map it onto a gold relation label in the test set.

In Table 5, except for the Top-1 sample for summarization, we also show two other typical samples to illustrate whether the generated relation name or description can define the underlying semantic information in them. Through observation, we have found that the relation annotated by ChatGPT exhibits semantic correlations or even equivalence with the gold relation label. Besides, the relations from ChatGPT can cover the relational semantics of other typical samples.

6 Conclusion

In this paper, we re-investigate the task of zero-shot relation extraction (ZSRE) and propose a training method SCL to transfer relational knowledge learned from seen relations to unseen relations. We formally define a three-step paradigm to perform data-driven relation extraction under a zero-shot setting, including encoder training, relation clustering, and summarization. Specifically, to train a discriminative relational encoder, we

[3] https://openai.com/blog/chatgpt.

propose a selective contrastive learning approach based on prompt-tuning, where false negative examples are removed and different importance scores are assigned to emphasize the importance of different negative samples. During the testing phase, we cluster the encoded test samples. To convert the virtual labels of the clustering results into relation labels, we select typical samples and summarize relation names from them.

Acknowledgments

This work was partially supported by National Key R&D Program of China (No. 2022YFB3103600), NSFC (Nos. U23A20296, 62272469, 62302513, 72371245), and The Science and Technology Innovation Program of Hunan Province (No. 2023RC1007).

References

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*, pages 2670–2676.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2010. Relational duality: Unsupervised extraction of semantic relations between entities on the web. In *WWW*, pages 151–160. ACM. https://doi.org/10.1145/1772690.1772707

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Chih-Yao Chen and Cheng-Te Li. 2021. ZS-BERT: Towards zero-shot relation extraction with attribute representation learning. In *NAACL-HLT*, pages 3470–3479. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.272

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *WWW*, pages 2778–2788. ACM. https://doi.org/10.1145/3485447.3511998

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. Relationprompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. In *ACL (Findings)*, pages 45–57. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-acl.5

Li Cui, Deqing Yang, Jiaxin Yu, Chengwei Hu, Jiayang Cheng, Jingjie Yi, and Yanghua Xiao. 2021. Refining sample embeddings with relation prototypes to enhance continual relation extraction. In *ACL/IJCNLP (1)*, pages 232–243. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.20

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Xiaobin Wang, Pengjun Xie, Haitao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2022. Prompt-learning for fine-grained entity typing. In *EMNLP (Findings)*, pages 6888–6901. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-emnlp.512

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. Making pre-trained language models better few-shot learners. In *ACL/IJCNLP (1)*, pages 3816–3830. Association for Computational Linguistics.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. Fewrel

2.0: Towards more challenging few-shot relation classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP* 2019, Hong Kong, China, November 3–7, 2019, pages 6249–6254. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP (1)*, pages 6894–6910. Association for Computational Linguistics.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. PTR: Prompt tuning with rules for text classification. *CoRR*, abs/2105.11259.

J. A. Hartigan and M. A. Wong. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108. https://doi.org/10.2307/2346830

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735. Computer Vision Foundation / IEEE. https://doi.org/10.1109/CVPR42600.2020.00975

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip S. Yu. 2020. Selfore: Self-supervised relational feature learning for open relation extraction. In *EMNLP (1)*, pages 3673–3682. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.299

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *NeurIPS*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *CoNLL*, pages 333–342. Association for Computational Linguistics. https://doi.org/10.18653/v1/K17-1034

Shuliang Liu, Xuming Hu, Chenwei Zhang, Shuang Li, Lijie Wen, and Philip S. Yu. 2022. Hiure: Hierarchical exemplar contrastive learning for unsupervised relation extraction. In *NAACL-HLT*, pages 5970–5980. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.437

Claudia Malzer and Marcus Baum. 2020. A hybrid approach to hierarchical density-based cluster selection. In *MFI*, pages 223–228. IEEE. https://doi.org/10.1109/MFI49285.2020.9235263

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *ACL (1)*. The Association for Computer Linguistics. https://doi.org/10.18653/v1/P16-1105

Abiola Obamuyide and Andreas Vlachos. 2018. Zero-shot relation classification as textual entailment. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 72–78. https://doi.org/10.18653/v1/W18-5511

OpenAI. 2022. Introducing chatgpt. https://openai.com/blog/chatgpt

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *HLT-NAACL*, pages 74–84. The Association for Computational Linguistics.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *COLING*, pages 5569–5578. International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.488

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *ACL (1)*, pages 2895–2905. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1279

Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. 2016. Multilingual relation extraction using compositional universal schema. In *HLT-NAACL*, pages 886–896. The Association for Computational Linguistics. https://doi.org/10.18653/v1/N16-1103

Shusen Wang, Bosen Zhang, Yajing Xu, Yanan Wu, and Bo Xiao. 2022. RCL: Relation contrastive learning for zero-shot relation extraction. In *NAACL-HLT (Findings)*, pages 2456–2468. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-naacl.188

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP (Demos)*, pages 38–45. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Liang Xu, Chun Zhang, Ning Zhang, and Xue-Shou Tian. 2023. Zero-shot relation extraction model via multi-template fusion in prompt. *Journal of Computer Applications*, pages 1–10.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. In *ACL/IJCNLP (1)*, pages 5065–5075. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.393

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP* 2015, Lisbon, Portugal, September 17–21, 2015, pages 1753–1762. The Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1203

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23–29, 2014, Dublin, Ireland*, pages 2335–2344. ACL.

Han Zhang, Bin Liang, Min Yang, Hui Wang, and Ruifeng Xu. 2022. Prompt-based prototypical framework for continual relation extraction. *EEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2801–2813. https://doi.org/10.1109/TASLP.2022.3199655

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *EMNLP*, pages 35–45. Association for Computational Linguistics. https://doi.org/10.18653/v1/D17-1004

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7–12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics. https://doi.org/10.18653/v1/P16-2034

A Explanation on the Definitions of Importance Score and Selective Contrastive Loss

In the original definition of contrastive loss in Equation (9), all negative instances are of equal importance. In this study, we posit that hard negatives play a more important role in representation learning, thereby distinguishing the importance of different contrastive pairs.

Therefore, we propose to use the Euclidean distance to assess the importance score of each negative contrastive instance:

$$w_{ij} = \frac{N \cdot \exp(-\|\mathbf{r}_i - \hat{\mathbf{r}}_j^-\|)}{\sum_{k=1}^{N} \exp(-\|\mathbf{r}_i - \hat{\mathbf{r}}_k^-\|)}. \quad (17)$$

Based on the distances between negative contrastive instances and the anchor instance, we can assess the difficulty of pushing each negative contrastive instance away from the anchor. The closer the distance, the more difficult the negative sample, indicating its higher importance in the contrastive loss. It is noteworthy that the importance score here is computed based on the relation representation of the instances rather than hyperparameters or learnable parameters.

After evaluating the importance scores of different negative instances, we can then obtain the selective contrastive learning loss:

$$\mathcal{L}_{SCL} = -\log \frac{\exp(\mathtt{sim}(\mathbf{r}_i, \hat{\mathbf{r}}_i)/\tau)}{\sum_{j=1}^{N} w_{ij} \cdot \exp(\mathtt{sim}(\mathbf{r}_i, \hat{\mathbf{r}}_j^-)/\tau)}. \tag{18}$$

Substituting Equation (2) into Equation (3) will yield:

$$\mathcal{L}_{SCL} = \\ -\log \frac{\sum_{k=1}^{N} \exp(\mathtt{sim}(\mathbf{r}_i, \hat{\mathbf{r}}_i)/\tau - \|\mathbf{r}_i - \hat{\mathbf{r}}_k^-\|)}{\sum_{j=1}^{N} N \cdot \exp(\mathtt{sim}(\mathbf{r}_i, \hat{\mathbf{r}}_j^-)/\tau - \|\mathbf{r}_i - \hat{\mathbf{r}}_j^-\|)}. \tag{19}$$

In this view, the selective contrastive loss seems to be a combination of two similarities. We did not directly employ Equation (19) to introduce the loss $\mathcal{L}_{SCL}$ because this form is less conducive to understanding its meaning. Therefore, we compute the importance score and selective contrastive loss via Equations (17) and (18), respectively, to better illustrate how we assess the importance score of a negative contrastive instance and how this importance score is reflected in the loss.