UOREX: Towards Uncertainty-Aware Open Relation Extraction

Rebii Jamal[1], Mounir Ourekouch[1,2], Mohammed Erradi[1,2,3]
[1]CID Development, Morocco
[2]University Mohammed VI Polytechnic, Morocco
[3]ENSIAS, University Mohamed V of Rabat, Morocco
rebiijamal1@gmail.com; mounir.ourekouch@um6p.ma;
mohammed.erradi@ensias.um5.ac.ma

Abstract

Open relation extraction (OpenRE) aims to identify relational facts within open-domain corpora without relying on predefined relation types. A significant limitation of current state-of-the-art OpenRE approaches is their inability to accurately self-assess their performance. Which is caused by the reliance on pseudo-labels, that treats all points within a cluster equally, regardless of their actual relative position according to the cluster center. This leads to models that are often overconfident in their incorrect predictions, significantly undermining their reliability. In this paper, we introduce an approach that addresses this challenge by effectively modeling a part of the epistemic uncertainty within OpenRE. Instead of using pseudo-labels that mask uncertainty, our approach is built to train a classifier directly with the clustering distribution. Our experimental results across various datasets demonstrate that the suggested approach improves the reliability of OpenRE by preventing overconfident errors. Furthermore, we show that by improving the reliability of the predictions, UOREX operates more efficiently in a generative active learning context where an LLM is the oracle, doubling the performance gain compared to the state-of-the-art.

1 Introduction

The extraction of relations between entities from unstructured text is an essential component of knowledge graph construction (Church and Bian, 2021; Mirtaheri, 2021). These graphs enable numerous downstream applications, such as web search (Xiong et al., 2017), question-answering (Yu et al., 2017), and more recently, Retrieval Augmented Generation (RAG) based on LLMs (Pan et al., 2024; Ren et al., 2023; Loconte et al., 2023). Traditionally, relation extraction has been limited to identifying predefined relations, leading to either the misclassification of new relations—adding inaccuracies to the knowledge graph—or labeling them as "unknown" (Zhao et al., 2023), which avoids misinformation but still overlooks valuable new connections. To address this limitation, open relation extraction has emerged as an essential research area for discovering and extracting new relation types from open-domain text.

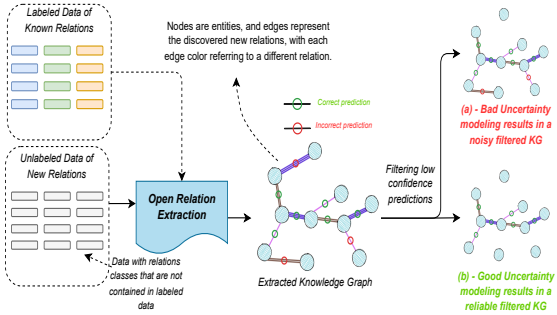

Figure 1: Comparison of the quality of Knowledge Graphs after filtering low-confidence predictions. In Case (a), where the OpenRE approach has poor uncertainty modeling, the filtered KG remains noisy due to overconfident errors. In Case (b), with good uncertainty modeling, the filtered KG is more reliable.

Most recent OpenRE research has focused on unsupervised relation discovery (URD) (Yao et al., 2011; Shinyama and Sekine, 2006; Simon et al., 2019), that aims to discover new relations **classes** in an open-domain corpora. Initial breakthroughs in this area were made by (Hu et al., 2020), followed by weakly-supervised or semi-supervised approaches like (Zhao et al., 2021), (Wu et al., 2019), (Wang et al., 2022a) and (Hogan et al., 2023). These methods employ a set of known relations as a starting point to discover new, unknown relations from documents. A common theme across these approaches is that they use clustering in a smaller space to guide a classifier that will discover relations in a much larger space. The main issue with this method is its reliance on pseudo-labels, which treat all points within a cluster equally, regardless of their actual relative position according

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6027–6040

to the cluster center.

This leads to three challenges: (1) Overconfidence in OpenRE models, complicating their integration into real systems due to the high risk of introducing noise into knowledge graphs as illustrated in Figure 1. (2) The need for thorough post-training evaluation of OpenRE methods. Given the nature of current OpenRE approaches, they are not inherently transparent; therefore, after training, there is no clear indication of their performance because the confidence of their predictions is forced to be high. This often requires the labeling of a significant portion of the target documents, which contradicts the inherent self-supervised nature of OpenRE. (3) Inefficient use of an LLM oracle in a generative active learning context. Because these models are overly confident, it's challenging to use them effectively in generative active learning scenarios (Settles, 2009; Xiao et al., 2023; Zhu et al., 2024) where an LLM serves as the oracle. Their inability to accurately model uncertainty means that even their least confident predictions include a large amount of correct answers, making the use of the oracle less efficient.

In this paper, we propose UOREX (**U**ncertainty-Aware **O**pen **R**elation **EX**traction) an approach that falls under the URD category and addresses the previously mentioned challenges.

Specifically, our approach is semi-supervised. We start with a set of **labeled data** containing **known relations** and our goal is to discover **novel relations** in the **unlabeled data**. We encode the contextual information of sentences and then project this encoded information into a smaller space. The adaptation of this space is achieved by optimizing a clustering loss that encourages labeled points to gather around their relational centers. As a result, unlabeled data points are separated based on their expressed relations. This separation is mainly related to the space learning the underlying concept behind relations. By taking advantage of this separation, we employ soft clustering to obtain a clustering distribution. This distribution captures an uncertainty that is epistemic in nature, arising from incomplete knowledge of relational patterns. We refer to this uncertainty as **relational uncertainty** and it is primarily linked to the number of known relations and is inversely proportional to the initial separation observed in the unlabeled data. More known relations result in less relational uncertainty and greater separation. On top of the encoded space, we train two classifiers: one for discovering relations and another for refining these discoveries. The discoverer is directly guided by the reordered distribution resulting from soft clustering using Kullback-Leibler (KL) divergence, thus passing the relational uncertainty to the discoverer which mitigates overconfident predictions. The refinement of discoveries is then facilitated by the second classifier that is trained to classify both labeled and unlabeled data.

This results in a model that understands the limits of its knowledge, thus enabling self-assessment of its performance using what we call **overall confidence**, a measure of average prediction confidence. Additionally, by modeling relational uncertainty, our approach encourages the model to assign lower confidence to its incorrect predictions. This means that errors are more likely to appear among low-confidence predictions, enhancing the efficiency of integrating our method into a generative active learning context.

In summary, the contributions of this paper are summarized as follows:

- **(1)** We introduce UOREX, a reliable OpenRE framework designed to mitigate overconfident errors often present in current SOTA OpenRE approaches.
- **(2)** We reduce the need for exhaustive post-training evaluation by enabling the overall confidence to represent the quality of the training.
- **(3)** We enhance OpenRE's efficiency in generative active learning contexts, thereby improving its overall performance gain.

Preliminary results demonstrate the effectiveness of our methodology in enhancing the overall reliability of OpenRE.

2 Related Work

Open Relation Extraction

OpenRE has rapidly evolved into an essential technique for knowledge graph construction, particularly due to its capability to extract emerging relation types from text without being limited to predefined categories. Early efforts in OpenRE employed tagging-based methods (Etzioni et al., 2008; Yates et al., 2007; Fader et al., 2011), where relations were directly extracted from text as spans. However, these methods often lacked generality due to the different ways the same relation could

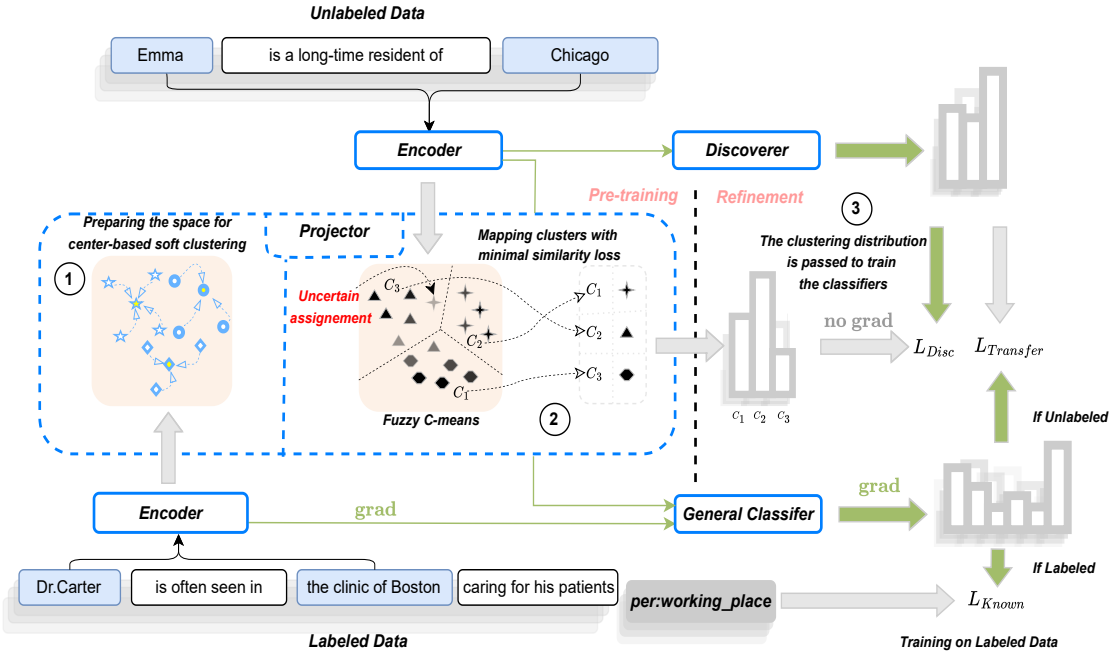

Figure 2: UOREX Architecture Overview. (1) The labeled data is used to pretrain the projection space enabling it to capture relational knowledge, (2) The soft clustering of unlabeled data produces a clustering distribution that we reorder to keep the learning consistent, (3) the clustering distribution is passed to train the discoverer, while the general classifier trains on the labeled data and the predicted discoverer distributions to refine the learning.

be expressed, and additionally, some relations are implicit and cannot be captured using spans. More recently Unsupervised Relation Discovery emerged as a new paradigm for OpenRE. This paradigm primarily relies on clustering-based methods to autonomously organize textual data into meaningful relational clusters. In this context, (Hu et al., 2020) introduced a self-supervised framework that exploits pretrained language models for adaptive clustering. More recent methodologies have leaned into the use of transferable knowledge. Prompt-based approaches like those of (Wang et al., 2022a) and (Hogan et al., 2023), along with methods such as (Zhao et al., 2021) and (Wu et al., 2019), employed predefined relational instances to train a model that can generalize to discover new, contextually relevant relations in an open-world setting. These methods leverage the relational knowledge embedded in these labeled instances to guide a clustering process, ensuring that newly discovered relations align more closely with realistic semantic contexts.

Uncertainty in Machine Learning

In machine learning (ML), accurately representing uncertainty is essential for ensuring the safety and reliability of models in various applications, enhancing their effectiveness across diverse domains (Lambrou et al., 2011; Varshney, 2016). Uncertainty in ML is typically divided into two categories: aleatoric and epistemic. Aleatoric uncertainty stems from the inherent randomness in model inputs and is irreducible. Epistemic uncertainty, on the other hand, originates from incomplete knowledge about the optimal model but can be reduced through the acquisition of more data or enhanced insights (Kiureghian and Ditlevsen, 2009). In semi-supervised learning, multiple sources of uncertainty are identified. Predictive uncertainty arises from the inherent variability in the data and the unpredictable nature of model outputs (Nguyen et al., 2019). Model uncertainty is associated with the selection of an appropriate hypothesis space, reflecting potential mismatches in model specification. Approximation uncertainty refers to the gap between the ideal hypothesis and the hypothesis estimated by the learning algorithm.

In our approach the relational uncertainty falls under the class of epistemic uncertainty, and is considered as part of the approximation uncertainty, as more labeled data reduces this uncertainty and further improves the performance of the estimated

hypothesis, note that our uncertainty is modeled by the soft clustering, the main challenge is to pass it to the rest of the components

3 Proposed approach: UOREX

The architecture of UOREX consists of four major components: **Encoder**, **Projector**, **Discoverer**, and **General Classifier**. As illustrated in Figure 2, we first employ the Encoder module to encode the contextual information of the relation using the entity pairs. The Projector then projects the encoded **labeled data** into a projection space. This projection space is then optimized for center-based clustering by herding the labeled data points close to their average relational center. Subsequently, we cluster the projected **unlabeled data** points using soft clustering. As the clustering classes of soft clustering change for each iteration, the distribution of soft assignments is reordered using a space-independent cluster similarity that maps the clusters to the previous cluster order. The Discoverer uses these reordered soft assignments to refine the **Encoder** space, initiating a cyclical optimization process. To prevent the Encoder space from collapsing in this cycle, a General Classifier is trained on the labeled data, which constrains the Encoder from merely following the descent directions of the Discoverer optimization. Additionally, the General Classifier is trained on the discoverer output distribution using a transfer loss to refine the learning. The approach results in obtaining two classifiers: one specialized in classifying new relations and a more general classifier for all existing relations.

3.1 Preliminaries

Let us define D_l as the set of labeled data points, and Y_l as the associated labels and $N_l = |D_l|$ as the count of these points. The set of known relations within D_l is denoted as R_{known}.

For unlabeled data, let D_u represent the set of unlabeled data points, with $N_u = |D_u|$, and R_{new} identifies the new relations.

The combined dataset, $D = D_u \cup D_l$, encompasses both labeled and unlabeled data points, totaling $N_T = |D|$. The space of all the entity-pair sentences is denoted by S_E, and we have $D \subseteq S_E$.

The different mappings are defined as follows: The encoder $f : S_E \rightarrow \mathbb{R}^N$ maps an entity-pair sentence into an N-dimensional encoded space. The projector head $g : \mathbb{R}^N \rightarrow \mathbb{R}^M$ projects this space into the projection space of dimension M. The decoder head $d : \mathbb{R}^M \rightarrow \mathbb{R}^N$ inverses the projection head behavior. The Discoverer, $\mu : \mathbb{R}^N \rightarrow \mathbb{R}^{|R_{\text{new}}|}$, classifies the new relations based on the encoder space. The General Classifier, $u : \mathbb{R}^N \rightarrow \mathbb{R}^{|R_{\text{new}}|+|R_{\text{known}}|}$, classifies both known and new relations.

3.2 Sentence Representation

Consider a set $s_p = \{x_p, e_p^1, e_p^2\}$ as an element from D. Here, x_p represents the sequence of tokens in a sentence, while e_p^1 and e_p^2 denote the positions of two specific entities within the sentence. These positions are defined as tuples $e_p^1 = (b_p^1, e_p^1)$ and $e_p^2 = (b_p^2, e_p^2)$, where b_p^1 and b_p^2 are the starting indices, and e_p^1 and e_p^2 are the ending indices of the entities in the token sequence.

We use BERT (Devlin et al., 2019) to encode the contextual information of each token in the sentence x_p:

$$H_p = \text{BERT}(x_p) \tag{1}$$

The result of the encoding step is H_p, a list of contextual embeddings for each token in x_p. Next, similar to (Hu et al., 2020) we apply max pooling over the range defined by the entity start and end indices. This is done to capture the most significant features of the encoded vectors that correspond to each entity:

$$t_p^0 = \text{MaxPooling}(H_p[b_p^1 : e_p^1]) \tag{2}$$

$$t_p^1 = \text{MaxPooling}(H_p[b_p^2 : e_p^2]) \tag{3}$$

Finally, these vectors are concatenated to form a single feature vector t_p for the pair of entities in sentence s_p. This vector serves as the output of the encoder for s_p and is defined as follows:

$$f(s_p) = t_p = t_p^0 \oplus t_p^1 \tag{4}$$

3.3 Projector

After the initial sentence encoding, the output is then passed through the projector. The primary function of the projector is to utilize data points from D_l to establish an appropriate space for clustering. We achieve that by minimizing the clustering loss, L_{Clust}, that encourages data points to migrate towards their respective relational centers C_{r_p}, with:

$$L_{\text{Clust}} = \frac{1}{N_l} \sum_{r_p \in R_{\text{known}}} \sum_{t_j \in r_p} \|g(t_j) - C_{r_p}\|^2 \tag{5}$$

$$C_{r_p} = \frac{1}{|\overline{r_p}|} \sum_{t_j \in \overline{r_p}} g(t_j) \tag{6}$$

Here, $\overline{r_p}$ represents the set of elements that express the relation r_p between two entities. And C_{r_p} is the center of each cluster associated with a relation r_p. To prevent the collapsing of the space, a decoder d is introduced alongside the projection head. The corresponding collapsing loss, L_{col}, is defined as:

$$L_{\text{col}} = \frac{1}{N_T} \sum_{p=1}^{N_T} (t_p - d(g(t_p)))^2 \tag{7}$$

The total loss for the projector, denoted as L_{P}, is then expressed as a weighted sum of the clustering and collapsing losses:

$$L_{\text{P}} = L_{\text{Clust}} + \gamma L_{\text{col}} \tag{8}$$

The unlabeled data points are then clustered using Fuzzy c-means clustering (Bezdek et al., 1984), with the soft assignments being represented by Y_u.

$$Y_u = \text{Fuzzy-cmeans}\left(g\left(f(D_u)\right)\right) \tag{9}$$

we note y_p^u as the soft assignment for an unlabeled data point t_p.

3.4 Discoverer

Before passing the soft assignments to the Discoverer, we reorder the clusters by assigning each to the cluster from the previous iteration with the highest similarity. This process involves calculating a similarity invariant to spatial changes, focusing solely on the comparison of the indices of data points between clusters, the similarity that satisfies these conditions is the number of shared indices between clusters:

$$S_{ij} = |C_i \cap C_j| \tag{10}$$

where C_i and C_j represent the sets of indices of elements in the i^{th} and j^{th} clusters, respectively. To maximize the overall similarity of the clusters matching, we employ the Hungarian algorithm (Kuhn, 2010), which effectively matches clusters to maximize their mutual similarity based on the calculated S matrix.

The Discoverer is then guided by the clustering distribution to optimize the encoded space towards optimal separation for relation discovery. Given an instance s_p from D_u, we aim to minimize:

$$L_{\text{Disc}}(s_p) = \text{KL}\left(y_p^u, \mu(t_p)\right) \tag{11}$$

By minimizing this loss, the Discoverer influences the encoder space, enhancing the separation between data points in the projection space based on their respective confidence levels. Consequently, less certain data points are not forcibly pushed into a cluster if the projector space does not accurately recognize their relational class, thereby avoiding overconfident errors for the discoverer. Simultaneously, this loss propagates the relational uncertainty captured in the projection space.

3.5 General classifier

The General Classifier (GC) is designed to break the optimization cycle and refine the learning process. We achieve that by employing a cross-entropy loss over the set of labeled data, using as a target distribution one hot encoded vectors y_p^l from Y_l:

$$L_{\text{Known}} = -\frac{1}{N_l} \sum^{N_l} y_p^l \log u(t_p) \tag{12}$$

This forces the descent direction found by the optimization of the Discoverer to align with the classification of labeled data. Since we assume that the optimal direction should push the model towards separating all types of relations, this approach ensures that only beneficial directions are favored. To further enhance the capabilities of the GC, a knowledge transfer loss from the Discoverer to the GC is applied. This loss, L_{Transfer}, uses the KL divergence with the target distribution $\hat{y}_p^u = 0_{|R_{\text{known}}|} \oplus y_p^u$:

$$L_{\text{Transfer}} = \text{KL}(\hat{y}_p^u, u(t_p)) \tag{13}$$

The total GC loss, L_{GC}, is then calculated as the weighted sum of these two losses, where ρ is a hyperparameter that modulates the extent of knowledge transfer within the learning process:

$$L_{\text{GC}} = L_{\text{Known}} + \rho \cdot L_{\text{Transfer}} \tag{14}$$

3.6 General Algorithm

Algorithm 1 outlines our proposed method. Initially, we pretrain the projector by minimizing the projector loss L_p with respect to its parameters (lines 7-10). To enhance the relational knowledge encapsulated within the initial cluster distributions, we train the General Classifier (GC) on the labeled data for a limited number of epochs (lines 11-14). Subsequently, we enter the main training

Algorithm 1 Algorithm for UOREX

1: **Input:**
2: Labeled Data D_l, Unlabeled Data D_u
3: Projector parameters Π
4: Discoverer and Encoder parameters Φ
5: General Classifier and Encoder parameters Γ
6: Learning rate β, Discovery factor μ_d
7: **Output:** Γ, Π, Φ
8: **for** $i = 1$ to $epoch_{pretrain}$ **do**
9: $\quad \Pi \leftarrow \Pi - \beta \nabla_{\Pi} L_p$
10: **end for**
11: **for** $i = epoch_{pretrain}$ to $epoch_{context}$ **do**
12: $\quad \Pi \leftarrow \Pi - \beta \nabla_{\Pi} L_p$
13: $\quad \Gamma \leftarrow \Gamma - \beta \nabla_{\Gamma} L_{\text{Known}}$
14: **end for**
15: **for** $i = epoch_{context}$ to *epochs* **do**
16: $\quad$ **for** $p = 1$ to d **do**
17: $\quad\quad$ Perform Fuzzy C-means clustering with random initialization
18: $\quad$ **end for**
19: $\quad$ Select the iteration with the best FPC
20: $\quad \Phi \leftarrow \Phi - \beta \mu_d \nabla_{\Phi} L_{\text{Disc}}$
21: $\quad \Gamma \leftarrow \Gamma - \beta \nabla_{\Gamma} L_{\text{GC}}$
22: $\quad \Pi \leftarrow \Pi - \beta \nabla_{\Pi} L_p$
23: **end for**

loop, where, we first search for the optimal clustering by repeating the Fuzzy C-means algorithm d times with random initialization (lines 16-18). We select the clustering distribution that achieves the highest fuzzy partition coefficient (FPC), ensuring the stability of our approach (line 19). We then optimize the Discoverer and the encoder parameters with respect to L_{Disc} , where μ_d is a tuning parameter that balances the joint effect of the GC and the Discoverer on the encoder. Concurrently, L_{GC}, which encompasses both transfer and known losses, is optimized along with the projector loss to adapt to the evolving encoder space (lines 20-22).

4 Experiments

4.1 Metrics

We evaluate the different compared approaches using the B3 measure F1 score (Bagga and Baldwin, 1998), the V-measure F1 score (Rosenberg and Hirschberg, 2007), and the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985).

Similar to (Lakshminarayanan et al., 2017), we use confidence thresholds to evaluate the uncertainty modeling part of our approach, as we think it is the clearest type of evaluation for our case, and because in practice most models are used with a confidence threshold.

4.2 Datasets

We conduct our experiments on two well-established English relation extraction datasets: FewRel and TACRED, to rigorously evaluate our proposed model for relation discovery.

TACRED
The TAC Relation Extraction Dataset (TACRED) (Zhang et al., 2017) is a large-scale, human-annotated dataset having 42 unique relations and 106,264 samples. Similar to (Zhao et al., 2021) in our setup, we exclude the *no_relation*. Our base relations set comprises 31 relations, and 10 relations as our unknown set for discovery. Finally, 15% of instances from the unknown set are randomly chosen to serve as our test set, while the remaining instances are allocated to the training set.

FewRel
We use FewRel (Han et al., 2018) as the second dataset for our experiments. FewRel is a large-scale, manually annotated dataset designed for few-shot relation classification, containing over 70,000 instances across 80 relation types. Similar to (Zhao et al., 2021) the base relations set is formed with 64 relations, and the unknown relations set is formed with 16 relations, we randomly chose 15% of the instances as a test set and we train on the rest.

4.3 Experimental Setup

To evaluate the effectiveness of our approach, we compare it with the following SOTA OpenRE methods: **SelfORE** (Hu et al., 2020), which operates under a fully unsupervised framework; and **RSN** (Wu et al., 2019), along with **RoCORE** (Zhao et al., 2021) that operates under a semi-supervised framework, and **KNoRD** (Hogan et al., 2023) a prompt-based OpenRE approach. For a fair comparison we reimplemented all the 4 approaches under the same configuration, we used a dichotomic search to calibrate their softmax temperature to all match the same data percentage above the 90% confidence threshold at the evaluation.

The hyperparameter configuration and the more detailed implementations and the specifics of our approach and the compared models are provided in the Appendix.

Threshold	Approach	TACRED			FewRel		
		$B3\_F1$	V_F1	ARI	$B3\_F1$	V_F1	ARI
$\alpha = 0$	SelfORE	0.523_{184}	0.561_{328}	0.353_{498}	0.744_{433}	0.818_{325}	0.702_{539}
	RSN	0.608_{203}	0.624_{036}	0.442_{216}	0.579_{155}	0.698_{053}	0.445_{216}
	RoCORE	$\mathbf{0.842_{116}}$	$\underline{0.877_{084}}$	0.780_{052}	0.850_{129}	$\mathbf{0.894_{083}}$	0.805_{180}
	KNoRD	$\underline{0.840_{056}}$	$\mathbf{0.883_{089}}$	$\mathbf{0.880_{075}}$	$\mathbf{0.859_{067}}$	0.878_{059}	$\mathbf{0.913_{068}}$
	UOREX	0.839_{162}	0.863_{098}	$\underline{0.806_{241}}$	$\underline{0.858_{129}}$	$\underline{0.888_{087}}$	$\underline{0.842_{140}}$
$\alpha = 50$	SelfORE	0.555_{345}	0.603_{450}	0.374_{590}	0.748_{426}	0.822_{317}	0.707_{533}
	RSN	0.608_{203}	0.624_{036}	0.442_{216}	0.579_{155}	0.698_{053}	0.445_{216}
	RoCORE	$\underline{0.869_{277}}$	$\underline{0.901_{232}}$	0.825_{371}	0.865_{188}	$\underline{0.908_{129}}$	0.822_{238}
	KNoRD	0.861_{067}	0.899_{089}	$\mathbf{0.904_{056}}$	$\underline{0.866_{059}}$	0.879_{068}	$\mathbf{0.925_{075}}$
	UOREX	$\mathbf{0.878_{129}}$	$\mathbf{0.902_{120}}$	$\underline{0.852_{187}}$	$\mathbf{0.896_{106}}$	$\mathbf{0.921_{060}}$	$\underline{0.884_{127}}$
$\alpha = 80$	SelfORE	0.622_{429}	0.660_{424}	0.423_{625}	0.787_{366}	0.856_{248}	0.758_{478}
	RSN	0.608_{203}	0.624_{036}	0.442_{216}	0.579_{155}	0.698_{053}	0.445_{216}
	RoCORE	0.902_{380}	0.921_{304}	0.872_{558}	$\underline{0.893_{207}}$	$\underline{0.933_{130}}$	0.856_{244}
	KNoRD	$\underline{0.925_{089}}$	$\underline{0.943_{056}}$	$\mathbf{0.970_{078}}$	0.883_{068}	0.867_{075}	$\underline{0.935_{059}}$
	UOREX	$\mathbf{0.940_{082}}$	$\mathbf{0.951_{077}}$	$\underline{0.948_{188}}$	$\mathbf{0.949_{067}}$	$\mathbf{0.965_{043}}$	$\mathbf{0.939_{072}}$
$\alpha = 90$	SelfORE	0.653_{459}	0.685_{380}	0.464_{688}	0.808_{342}	0.872_{211}	0.786_{457}
	RSN	0.732_{612}	0.701_{532}	0.550_{342}	0.690_{455}	0.777_{348}	0.577_{567}
	RoCORE	0.911_{554}	0.930_{465}	0.884_{874}	$\underline{0.908_{230}}$	$\underline{0.942_{140}}$	0.885_{339}
	KNoRD	$\underline{0.932_{075}}$	$\underline{0.948_{056}}$	$\underline{0.978_{089}}$	0.907_{059}	0.870_{068}	$\underline{0.948_{067}}$
	UOREX	$\mathbf{0.980_{036}}$	$\mathbf{0.981_{052}}$	$\mathbf{0.986_{095}}$	$\mathbf{0.961_{049}}$	$\mathbf{0.974_{032}}$	$\mathbf{0.952_{052}}$

Table 1: Evaluation of the approaches at varying confidence thresholds for TACRED and FewRel. Subscripts denotes standard deviation (e.g., 0.948_{188} represents 0.948 ± 0.0188). The best result for each metric is **bolded**, and the second-best is underlined.

4.4 Overall Performance

Table 1 presents the evaluation results of the compared approaches under different prediction confidence thresholds α. From these results, we can draw two conclusions:

(1) UOREX outperforms the state-of-the-art by a large margin in terms of reliability. This is displayed by how our approach avoids overconfident errors, where increasing α drastically augments the performance, achieving for $\alpha = 90$ improvements over the second best performing approach of +5.1%, +3.4%, and +0.8% on TACRED for the B3, F1 V-measure, and ARI, respectively. Similarly, for FewRel, we observe improvements of +5.8%, +3.3%, and +0.4% for the B3, F1 V-measure, and ARI, respectively, indicating that the majority of incorrect model predictions are made with low confidence.

(2) UOREX achieves competitive results in normal settings while being more reliable. Our method achieves competitive performance compared the second best performing approach, showing variations of -1.4% and -2.3% on TACRED, and -0.1% and -0.7% on FewRel for the B3 score, F1 V-measure respectively. While the ARI variation is slightly higher -9.1% on TACRED and -8.4% on FewRel, this is acceptable because ARI

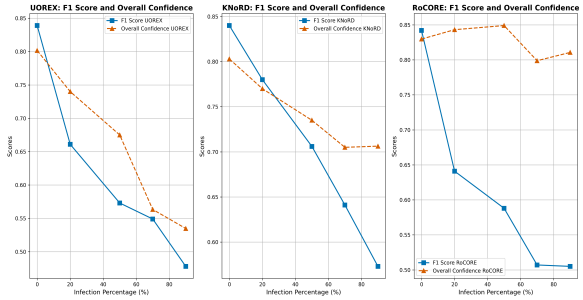

Figure 3: Self-Assessment comparison, (left) UOREX (middle) KNoRD (right) RoCORE, the red curve is the evolution of Overall confidence, the blue curve is the evolution of the B3 F1 score, the x axis values correspond to the infection percentage of the data

emphasizes exact cluster alignment and is sensitive to minor misalignments, and compared to KNoRD our approach is superior by a variation of +11.3% for the F1 V-measure on Fewrel. This demonstrates UOREX's capability to compete effectively while maintaining greater reliability.

We can also note from Table 1 the behavior of the compared approaches. Among all, RSN achieves the worst performance in terms of reliability. This is primarily because RSN infers similarity rather than class membership, which leads to very overconfident predictions, often close to 1. After recalibration, 0.9 was established as the point of separa-

Data	B3 F1 Score
Overall	0.917
UOREX Lowest 10%	0.899
KNoRD Lowest 10%	0.938
RoCORE Lowest 10%	0.948

Table 2: Performance of Oracle based on GPT-4o on overall test data and 10% lowest confident predictions for different approaches

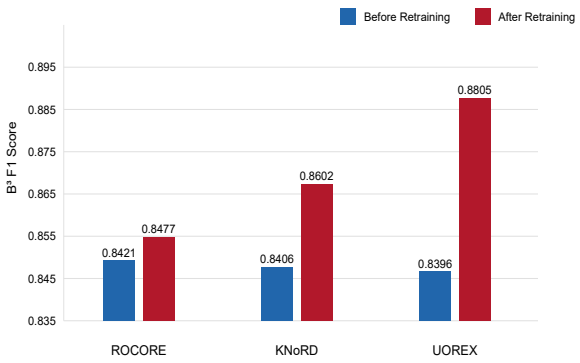

Figure 4: Performance gain comparison before and after retraining using oracle labeled data of the 10% lowest confident predictions for different approaches

tion, explaining RSN's unchanged behavior until the 0.9 threshold. For RoCORE, we observe that its performance does not significantly improve as we increase α, with only an +8.1% difference for the B3 F1 score between the first and last thresholds, in contrast, UOREX achieves a +16.8% difference. KNoRD shows a good ARI scores overall and competitive F1 scores, but as we increase α UOREX shows a greater performance.

4.5 Self-Assessment Evaluation

In this experiment, we evaluate the self-assessment capability of UOREX. To do so, we infect a percentage of the TACRED dataset sentences by replacing 60% of the words with random words, excluding the entity arguments, and evaluate the overall confidence evolution alongside the infection percentage. Figure 3 shows the results for UOREX, KNoRD and RoCORE. For RoCORE, as the infection percentage increases, the performance of the model logically declines, but the overall confidence of the predictions does not change. This results in the model having an overall prediction confidence of 81% at an infection percentage of 90%, with a B3 F1 score of 0.505. For KNoRD, we observe a slightly better self-assessment capability compared to RoCORE. Nonetheless, the overall confidence does not clearly correlate with the model's performance. The difference in overall confidence between 20% and 90% infection percentages is approximately ±0.07. This small variance makes it challenging to distinguish between correct and poor performance based solely on the overall confidence. In contrast, for UOREX, there is a clear correlation between the model performance and the overall confidence: as the infection percentage increases, the overall confidence of the predictions decreases, indicating that the model recognizes when it has not performed well, unlike KNoRD or RoCORE. This experiment further proves that UOREX is capable of reliable self-assessment of its performance, reducing the need for exhaustive post-training evaluation for OpenRE.

4.6 Evaluating the Impact of GPT-4o Oracle

Given the high cost of recent LLMs, we evaluate the efficiency of our approach by using GPT-4o (Wu et al., 2024) as an oracle within a generative active learning framework, the oracle is constructed using a straightforward adaptation of current prompting techniques to evaluate our OpenRE approach. Mainly, our oracle operates through a multi-step prompting process. It begins by aligning with the model's class indices using the most confident predictions from each class, then samples different chains of thought responses. A majority voting is then applied to select the final prediction (Wang et al., 2022b). To measure the oracle's effectiveness in combination with the different OpenRE approaches, we take the lowest 10% confident predictions for each approach and label them using the oracle aligned predictions. Table 2 summarizes the oracle's performance across the entire test dataset and the least confident predictions per approach. We then retrain each model with the enhanced dataset. The results, displayed in Figure 4, reveal that UOREX achieves a performance gain of 4.87%, which is over twice the improvement seen in KNoRD (2.33%) and more than seven times that of RoCORE (0.67%). Notably, although the oracle performs lowest on UOREX's least confident predictions (as shown in Table 2), UOREX benefits the most from this retraining phase. This suggests that UOREX's lowest-confidence predictions correspond to challenging, often incorrect examples. Consequently, despite lower oracle performance on these difficult cases, the resulting label correc-

tions yield substantial gains in model accuracy for UOREX.

5 Conclusion

In this work, we introduce a novel and reliable OpenRE approach UOREX, designed to address the prevalent issue of overconfident errors in current state-of-the-art OpenRE approaches caused by the use of pseudo-labels. UOREX improves the reliability of OpenRE, by modeling a relational uncertainty, that enables the model to understand the limits of its knowledge. As a result, UOREX avoids confident mistakes, enables self-assessment of performance using overall confidence after training, and doubles the performance gain of OpenRE models from oracles in a generative active learning context.

Experimental results on multiple datasets demonstrate that UOREX not only enhances the overall reliability of OpenRE but also achieves competitive performance. These advancements establish UOREX as a solution for more dependable OpenRE models. For future work, we intend to explore refining the uncertainty modeling using bayesian approaches for an even greater model reliability and extend the scope of the approach to work for low ressource languages. More importantly, we also plan to integrate the suggested approach within an integrated intelligent platform dedicated to support real-world applications for public sector agencies.

6 Limitations

We acknowledge several limitations in the proposed contributions. (1) UOREX relies on a large set of predefined relations. (2) UOREX utilizes BERT as its encoder, making it dependent on the performance of BERT models. This reliance can pose problems for low resource languages, where BERT may not perform optimally due to limited training data availability.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains.

James C. Bezdek, Robert Ehrlich, and William Full. 1984. Fcm: The fuzzy c-means clustering algorithm. *Computers Geosciences*, 10(2):191–203.

Kenneth Church and Yuchen Bian. 2021. Data collection vs. knowledge graph completion: What is needed to improve coverage? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6210–6215, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Commun. ACM*, 51(12):68–74.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Geoffrey E Hinton and Sam Roweis. 2003. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, volume 15, pages 857–864. MIT Press.

William Hogan, Jiacheng Li, and Jingbo Shang. 2023. Open-world semi-supervised generalized relation discovery aligned in a real-world setting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14227–14242, Singapore. Association for Computational Linguistics.

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. *J. of Classification*, 2(1):193–218.

Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112. Risk Acceptance and Risk Communication.

Harold W. Kuhn. 2010. *The Hungarian Method for the Assignment Problem*, pages 29–47. Springer Berlin Heidelberg, Berlin, Heidelberg.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Antonis Lambrou, Harris Papadopoulos, and Alex Gammerman. 2011. Reliable confidence measures for medical diagnosis with evolutionary algorithms. *IEEE Transactions on Information Technology in Biomedicine*, 15(1):93–99.

Lorenzo Loconte, Nicola Di Mauro, Robert Peharz, and Antonio Vergari. 2023. How to turn your knowledge graph embeddings into generative models. In *Advances in Neural Information Processing Systems*, volume 36, pages 77713–77744. Curran Associates, Inc.

Mehrnoosh Mirtaheri. 2021. Relational learning to capture the dynamics and sparsity of knowledge graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(18):15724–15725.

Vu-Linh Nguyen, Sébastien Destercke, and Eyke Hüllermeier. 2019. Epistemic uncertainty sampling. In *Discovery Science: 22nd International Conference, DS 2019, Split, Croatia, October 28–30, 2019, Proceedings 22*, pages 72–86. Springer.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Hongyu Ren, Mikhail Galkin, Michael Cochez, Zhaocheng Zhu, and Jure Leskovec. 2023. Neural graph reasoning: Complex logical query answering meets graph databases. *ArXiv*, abs/2303.14617.

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, Prague, Czech Republic. Association for Computational Linguistics.

Burr Settles. 2009. Active learning literature survey.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 304–311, New York City, USA. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Kush R Varshney. 2016. Engineering safety in machine learning. In *2016 Information Theory and Applications Workshop (ITA)*, pages 1–5. IEEE.

Jiaxin Wang, Lingling Zhang, Jun Liu, Xi Liang, Yujie Zhong, and Yaqiang Wu. 2022a. MatchPrompt: Prompt-based open relation extraction with semantic consistency guided clustering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7875–7888, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Huai hsin Chi, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171.

Ruidong Wu, Yuan Yao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2019. Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 219–228, Hong Kong, China. Association for Computational Linguistics.

Yiqi Wu, Xiaodan Hu, Ziming Fu, Siling Zhou, and Jiangong Li. 2024. Gpt-4o: Visual perception performance of multimodal large language models in piglet activity understanding. *ArXiv*, abs/2406.09781.

Rui Xiao, Yiwen Dong, Junbo Zhao, Runze Wu, Minmin Lin, Gang Chen, and Haobo Wang. 2023. Freeal: Towards human-free active learning in the era of large language models. *ArXiv*, abs/2311.15614.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1271–1279, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

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

Alexander Yates, Michele Banko, Matthew Broadhead, Michael Cafarella, Oren Etzioni, and Stephen Soderland. 2007. TextRunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, Rochester, New York, USA. Association for Computational Linguistics.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 571–581, Vancouver, Canada. Association for Computational Linguistics.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Jun Zhao, Tao Gui, Qi Zhang, and Yaqian Zhou. 2021. A relation-oriented clustering method for open relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9707–9718, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jun Zhao, Xin Zhao, WenYu Zhan, Qi Zhang, Tao Gui, Zhongyu Wei, Yun Wen Chen, Xiang Gao, and Xuanjing Huang. 2023. Open set relation extraction via unknown-aware training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9453–9467, Toronto, Canada. Association for Computational Linguistics.

Muzhi Zhu, Chengxiang Fan, Hao Chen, Yang Liu, Weian Mao, Xiaogang Xu, and Chunhua Shen. 2024. Generative active learning for long-tailed instance segmentation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 62349–62368. PMLR.

A Appendix

A.1 Ablation Study

To study the contribution of each component in the proposed method, we conduct ablation experiments on the two datasets, Table 3 illustrates the found results, UOREX w/o GC does not use a General classifier, thereby using the labeled data for the pretraining of the projector only, This leads to a drastic drop in performance, as the optimization cycle formed with the Encoder - Projector - Discoverer results in the encoder space collapsing. This issue can also be viewed as a mismanagement of known relation knowledge, which is not used to supervise the encoder space, but only the projector. Consequently, the model tends to converge toward the initial clustering results after pretraining, bypassing the refinement step. UOREX w/o Soft Clustering uses fuzzy c-means with $m = 1 + 10^{-4}$ which leads to hard assignments, ignoring the relational uncertainty. UOREX w/o Pretrain do not pretrain the projector before beginning to optimize the Discoverer and the General Classifier. This leads to poor performance, as the clustering distributions passed capture no relational knowledge, resulting in the collapse of the projector space.

Dataset	Model	B3_F1	V_F1	ARI
TACRED	UOREX	0.839	0.863	0.806
	w/o Soft Clustering	0.805	0.852	0.709
	w/o GC	0.697	0.739	0.593
	w/o Pretrain	0.318	0.257	0.108
FewRel	UOREX	0.858	0.888	0.842
	w/o Soft Clustering	0.782	0.853	0.730
	w/o GC	0.693	0.783	0.607
	w/o Pretrain	0.229	0.285	0.108

Table 3: Ablation results on FewRel and TACRED

A.2 Implementation Details

We ran the training on NVIDIA A100 GPUs, allocating 6GB of memory for TACRED and 8GB for FewRel. The reported training configuration takes approximately 8 hours for TACRED and 14 hours for FewRel. For our experiments, we utilized BERT-base uncased and unfroze the 8th layer, Table 4 summarize the rest of the hyperparameters used.

For the implementation of the compared approaches:

- **SelfORE:** We re-implemented the SelfORE approach from scratch, following to the reported configuration in the original publication.
- **RSN:** We utilized the existing RSN repository, extended its data configuration for TACRED compatibility.
- **RoCORE:** We re-implemented RoCORE from scratch, selecting a low transfer coefficient (0.001) as it gave the best results.
- **KNoRD:** For KNoRD, we used the open implementation and adapted it to our test and train set. For TACRED we used the original paper configuration changing the setup to classify unlabeled data only. For FewRel we changed the hyperparameter from the original paper, to find better results we used a high quality data threshold of 0.06 instead of 0.15 in the original paper.

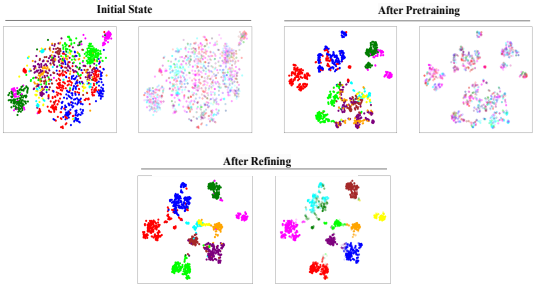

Figure 5: The Projection space evolution, for different stages, for each stage we have (Left) real labels (Right) the predictions of the discoverer with the confidence level represented by the opacity of the point.

A.3 Evolution of the space of representations

Figure 5 illustrates the evolution of the relational space with correct labels (left) and the predictions made by the discoverer (right), where the color intensity represents confidence levels. We utilize t-SNE (Hinton and Roweis, 2003) to visualize this space in 2D. Initially, the projector weights are randomly initialized, causing the data points to be dispersed randomly across the space without relational considerations. A slight separation can be observed, attributable to the contextual information encoded by BERT in its initial state. After the projector's pretraining, the space shows significantly improved separation, and some relation classes become more apparent. Although part of the relation classes are mixed, the application of soft clustering ensures the gradual refinement of these classes until they converge. Once the refining stage is completed, the separation between classes becomes more distinct, with uncertain data points lying in the areas between clusters.

Hyperparameter	**Value**
Optimizer	Adam
Learning Rate (β)	1×10^{-4}
Batch Size	100
L2 Regularization (λ)	10^{-5}
epochs	100
max length	160
Discovery Factor (μ_d)	0.01
Weighting Factor (γ)	0.001
Knowledge transfer factor (ρ)	0.01
cluster fuzziness factor (m)	1.3
Pre-training Epochs ($epoch_{pretrain}$)	10
Context Epochs ($epoch_{context}$)	11
Clustering trials number (d)	30

Table 4: Hyperparameters used

A.4 Errors among the lowest 10% confident predictions

Approach	Percentage
UOREX	0.746
KNoRD	0.620
RoCORE	0.538

Table 5: Error percentage among the lowest 10% confident predictions

Table 5 illustrates the difference in error percentages among the lowest 10% confident predictions. This explains why UOREX performs significantly better within a generative active learning framework, as the majority of labeled data from the oracle are mistakes. This implies that UOREX gains the most from retraining by effectively learning from errors.

A.5 Oracle Implementation Details

To implement the oracle, we used `gpt-4o-2024-08-06` from the OpenAI API. We sampled 5 answers per query, and selected 10 confident examples from each class for indices alignment. The process followed this pipeline:

1. Among the k (number of novel relations) most confident classes, retrieve the 10 most confident examples from each predicted class.
2. Construct a prompt following the *Alignment Template* (see Appendix A.6.1).
3. After deducing the relation names for each class ID, we use another prompt to review the homogeneity of these classes by incorporating examples from the labeled data to guide the LLM (see *Refinement Prompt* in Appendix A.6.2).
4. Once the relation list is refined, we prompt the LLM to provide potential extensions of these classes and exhaustive definitions (see *Extension Prompt* in Appendix A.6.3).
5. After obtaining a broad analysis of each relation class represented by `{extended_classes}`, we classify data instances using the *Classification Prompt* (see Appendix A.6.5).
6. Using the previous prompt, we sample 5 potential answers and employ majority voting. An example of a response is provided in Appendix A.6.4.

A.6 Prompt Templates

A.6.1 Alignment Template

Alignment Template

I have unlabeled data, and my model generated predictions in the form of prediction class indices. For each prediction index, I need you to deduce the relation name that corresponds to that class.
{
class_id_1: {
ex1: sentence, entities
},
class_id_2: {
ex2: sentence, entities
},
...
}
For each class, go through the sentences and entities carefully, explain your reasoning step-by-step, and provide the relation name between the entities. Once you've deduced all relation names, summarize by listing each relation name corresponding to the different classes.

A.6.2 Refinement Prompt

Refinement Prompt

Now, provide the list of relation names, ensuring that each class represents a distinct relation. Be careful with granularity, as some classes may appear similar but hold subtle differences that are important to capture. For example, *city of birth* and *state of birth* may seem alike but represent different levels of detail. Similarly, distinctions like *date of employment* vs. *year of employment* or *manager of* vs. *employer of* are crucial to reflect accurately.
Please list each relation name, paying attention to these nuances to ensure precise representation of each class.

A.6.3 Extension Prompt

Extension Prompt

Examine the provided relation classes critically and assess whether some categories may be too narrowly defined. In cases where the current class names are highly specific, it's important to reframe them into more inclusive categories that capture a wider range of relationships between entities. Think about how each relation could encompass more generalized associations, rather than being limited to precise, narrowly defined roles. The goal is to ensure that the relation classes remain versatile enough to cover various cases, thus reducing the risk of overfitting to overly specific situations.
When analyzing each relation, consider how it might be grouped into a broader category that captures related associations without sacrificing relevance. This approach will make the classification more flexible and applicable across diverse contexts, avoiding overly restrictive interpretations that may miss key relational nuances.

A.6.4 Sample Response

Sample Response

To determine the appropriate relation class for the entities "Ble Goude" and "Guiberoua" in the given sentence, we should analyze the relationship between the head and tail entities.
Analysis:

- **Sentence:** "Ble Goude was born in 1972 in Gbagbo's centre west home region, Guiberoua."
- **Head Entity:** Ble Goude
- **Tail Entity:** Guiberoua
- **Relation:** The sentence mentions Ble Goude's birthplace, indicating a relationship between a person and their birth location.

Matching to Existing Class:
Looking at the generalized relation categories and the existing classes:

- **Class 9:** *Person-BornIn-City* with the generalized relation *Person-BornIn-Location* fits this scenario as it describes a person's place of birth, which is precisely the relationship described in the sentence.
- **Other Classes:** None of the other classes primarily focus on a person's birthplace.

Conclusion:
Therefore, the relationship between "Ble Goude" and "Guiberoua" should be classified under the class that captures a person's birthplace.
Here is the JSON response with the index and the class ID:
[language=json] "index": 1, "class$_id$" : 9
This classification reflects Class 9's focus, which fits the relationship depicted here, taking into account the broader interpretation of birthplace as any location.

A.6.5 Classification Prompt

Classification Prompt

Assign a relation class to the relation between these two entities:
`{example}`
Return a JSON with the index and the class ID in the form `{"index": idx, "class_id": id}`. You need to find the closest class. The following is the extended description of each class:
`{extended_classes}`