

# Contrastive Learning with Hard Negative Entities for Entity Set Expansion

Yinghui Li\*

liyinghu20@mails.tsinghua.edu.cn  
Tsinghua Shenzhen International  
Graduate School, Tsinghua University  
Shenzhen, Guangdong, China

Yangning Li\*

liyn20@mails.tsinghua.edu.cn  
Tsinghua Shenzhen International  
Graduate School, Tsinghua University  
Shenzhen, Guangdong, China

Yuxin He\*

21S051047@stu.hit.edu.cn  
Harbin Institute of Technology  
Shenzhen, Guangdong, China

Tianyu Yu

Tsinghua Shenzhen International  
Graduate School, Tsinghua University  
Shenzhen, Guangdong, China

Ying Shen<sup>†</sup>

School of Intelligent Systems  
Engineering, Sun-Yat Sen University  
Shenzhen, Guangdong, China

Hai-Tao Zheng<sup>†</sup>

Tsinghua Shenzhen International  
Graduate School, Tsinghua University  
Shenzhen, Guangdong, China

## ABSTRACT

Entity Set Expansion (ESE) is a promising task which aims to expand entities of the target semantic class described by a small seed entity set. Various NLP and IR applications will benefit from ESE due to its ability to discover knowledge. Although previous ESE methods have achieved great progress, most of them still lack the ability to handle *hard negative entities* (i.e., entities that are difficult to distinguish from the target entities), since two entities may or may not belong to the same semantic class based on different granularity levels we analyze on. To address this challenge, we devise an entity-level masked language model with contrastive learning to refine the representation of entities. In addition, we propose the ProbExpan, a novel probabilistic ESE framework utilizing the entity representation obtained by the aforementioned language model to expand entities. Extensive experiments<sup>1</sup> and detailed analyses on three datasets show that our method outperforms previous state-of-the-art methods.

## CCS CONCEPTS

• Information systems → Retrieval models and ranking.

## KEYWORDS

Knowledge Discovery; Entity Set Expansion; Contrastive Learning

## ACM Reference Format:

Yinghui Li, Yangning Li, Yuxin He, Tianyu Yu, Ying Shen, and Hai-Tao Zheng. 2022. Contrastive Learning with Hard Negative Entities for Entity Set Expansion. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*, July 11–15, 2022, Madrid, Spain. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3477495.3531954>

\*These authors contributed equally to this research

<sup>†</sup>Corresponding author: sheny76@mail.sysu.edu.cn, zheng.haitao@sz.tsinghua.edu.cn.

<sup>1</sup>The source codes are available at <https://github.com/geekjuro/ProbExpan>.

## 1 INTRODUCTION

Entity Set Expansion (ESE) aims to expand from a set of seed entities (e.g., “China”, “America”, “Japan”) to more new target entities (e.g., “Russia”, “Germany”, ...) that belong to the same semantic class (i.e., Country) as the seed entities. The ESE task can benefit kinds of NLP or IR downstream applications, such as knowledge graph construction [26], Web search [2], and question answering [30].

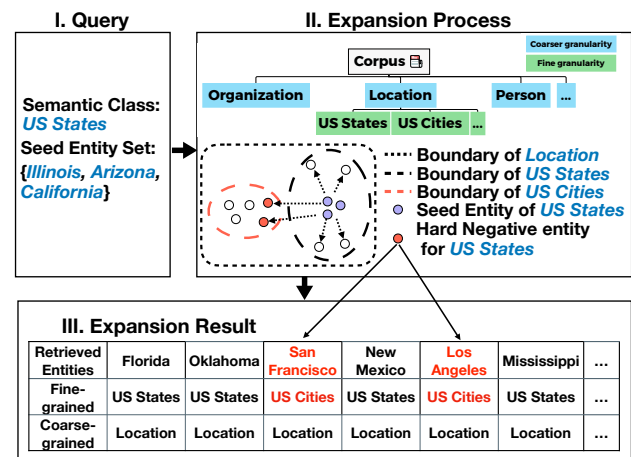


Figure 1: Examples of hard negative entities in ESE.

In recent years, kinds of iterative bootstrapping methods have gradually become the mainstream of ESE research. These methods [24, 32, 33] mainly select the most confident candidate entities of the model to the expanded set iteratively. A core challenge for these methods is to avoid selecting *hard negative entities* that are semantically ambiguous with the target entities [7, 12]. As shown in Figure 1, when we want to expand target entities belong to class US States, a competitive model is likely to wrongly expand hard negative entities, such as “San Francisco” and “Los Angeles”. If judged according to US States, it is clear that these hard negative entities do not belong to the target semantic class. But if we follow a coarser granularity (i.e., Location), hard negative entities can be regarded as the same class as target entities. Furthermore, as the



This work is licensed under a Creative Commons Attribution International 4.0 License.

SIGIR '22, July 11–15, 2022, Madrid, Spain

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8732-3/22/07.

<https://doi.org/10.1145/3477495.3531954>

characteristic of the iterative expansion process, a small number of negative entities that are incorrectly expanded in the early iterations will cause errors to accumulate in later iterations, resulting in a gradual decrease in expansion performance. This is the long-term “semantic drift” problem faced by ESE methods [3, 18, 25].

To address the above challenge and problem, we propose to use contrastive learning [1, 5, 21] to empower the ESE model to better deal with hard negative entities. Contrastive learning is a hot topic in self-supervised learning field, which is originally applied to learn visual representation and attracts more and more attention from NLP researchers. In general, contrastive learning aims to learn more effective representation by pulling samples belonging to similar classes and further pushing samples from different classes [8]. Intuitively, the idea that we want the ESE model to better distinguish hard negative entities coincides with the motivation of contrastive learning naturally. In our study, contrastive learning provides the ESE model with clearer semantic boundary and make entity representation closer to semantics.

Motivated by the above intuition, we propose a novel ESE method that consists of three parts: (1) *Entity representation model*, an entity-level masked language model pre-trained under the entity prediction task we specially design for ESE. Then we apply contrastive learning to refine the semantic representation learned by our model, which can be utilized in later expansion process. (2) *Model selection and ensemble*. Due to the randomness of training samples in the pre-training process, the model will be sensitive to the quality of the training context features. To alleviate this issue, we will pre-train multiple models mentioned in (1), then select and ensemble top models to avoid the randomness of single model. (3) *Probabilistic expansion framework*, a novel framework that can utilize the ensemble model obtained in (2) through a window search algorithm and an entity re-ranking algorithm, both based on probabilistic representation similarity of the candidate entity and entity set. Through this framework, we can finally get the ideal target entities that we want to expand.

In summary, our contributions are in three folds:

- We firstly apply contrastive learning in ESE to better handle hard negative entities and derive more effective entity representation in semantic space.
- We propose a novel ESE framework, ProbExpan, which can uniformly represent entity/entity set in the probability space and utilize the ESE model to expand target entities.
- We conduct extensive experiments and detailed analysis on three public datasets and get state-of-the-art performance. Solid results demonstrate the substantial improvement of our method over previous baseline methods.

## 2 RELATED WORK

**Entity Set Expansion.** Recently, many corpus-based ESE methods have gradually become the mainstream paradigm. These corpus-based ESE methods can be divided into two main categories:(1) one-time ranking methods [15, 17, 33] which introduce pairwise semantic similarity into set expansion tasks and suffer a lot from the *Entity Intrusion Error* problem, that is, they cannot clearly convey the semantic meaning of entities. (2) iterative pattern-based bootstrapping methods [11, 22, 24] which aim to bootstrap the seed

entities set by iteratively selecting context pattern and ranking expanded entities. But these methods usually are troubled by the *Semantic Drift* problem, that is, the target semantic class will change gradually when noise arises during the iterations.

**Language Representation.** Early representation methods focus on word-level embeddings, such as Word2Vec [6] and Glove [19] which output a single embedding for each word in vocabulary. After that, researchers design many context-aware representation methods to utilize context information better. The outstanding representative of context-aware representation methods is the masked language models represented by BERT [4]. It is noted that CGExpan [34] has utilized BERT’s representation to enhance ESE. But BERT can still only perform word-level representation and CGExpan only use the pre-trained BERT embeddings without any task-specific training. To obtain better representation in more complex tasks, ERNIE [35] is designed to learn entity/phrase-level representation. To the best of our knowledge, entity-level representation methods have not yet been applied to ESE.

**Contrastive Learning.** Contrastive learning has been widely applied in self-supervised field [14, 16, 20, 29]. The main motivation of contrastive learning is to attract the positive samples and repulse the negative samples [1, 5, 8, 13]. Recent work [21] shows that contrastive representation learning benefits from hard negative samples(those samples which are difficult to distinguish from positive samples). This idea coincides with the challenge we have observed in the existing ESE methods, that is, most expansion models cannot handle hard negative entities well. SynSetExpan [23] enhances the ESE task via another related task, Synonym Discovery. It is different from the idea of contrastive learning, hoping to suppress the effect of negative entities by obtaining more positive entities. NEG-FINDER [18] is concerned with semantic negative classes same as our work, but it proposes to perform offline negative discovery and then utilize the pre-selected negative categories to alleviate the semantic drift of the bootstrapping algorithms. Unlike NEG-FINDER that is just a heuristic and untrainable algorithm, our study aims to pre-train a task-specific model which has better entity representation and clearer semantic boundaries for ESE by contrastive learning.

## 3 METHODOLOGY

In this section, we firstly introduce the entity representation model and the entity prediction task we design for ESE. Specially, we will discuss how we apply contrastive learning to refine entity representation. Then we will illustrate the mechanism of model selection and ensemble. Finally, we will describe the expansion framework and algorithm to expand target entities. The overview of our proposed method is shown in Figure 2.

### 3.1 Entity Representation Model

The entity representation model mainly contains an entity-level masked language model, which takes a tokenized sentence with entity masked as input and outputs a probability distribution describing which entity the masked token can be. The entity representation is defined as the average of the predicted entity distributions of all its sentences. And the representation of an entity set is defined as the average of the representation of all its entities. Our

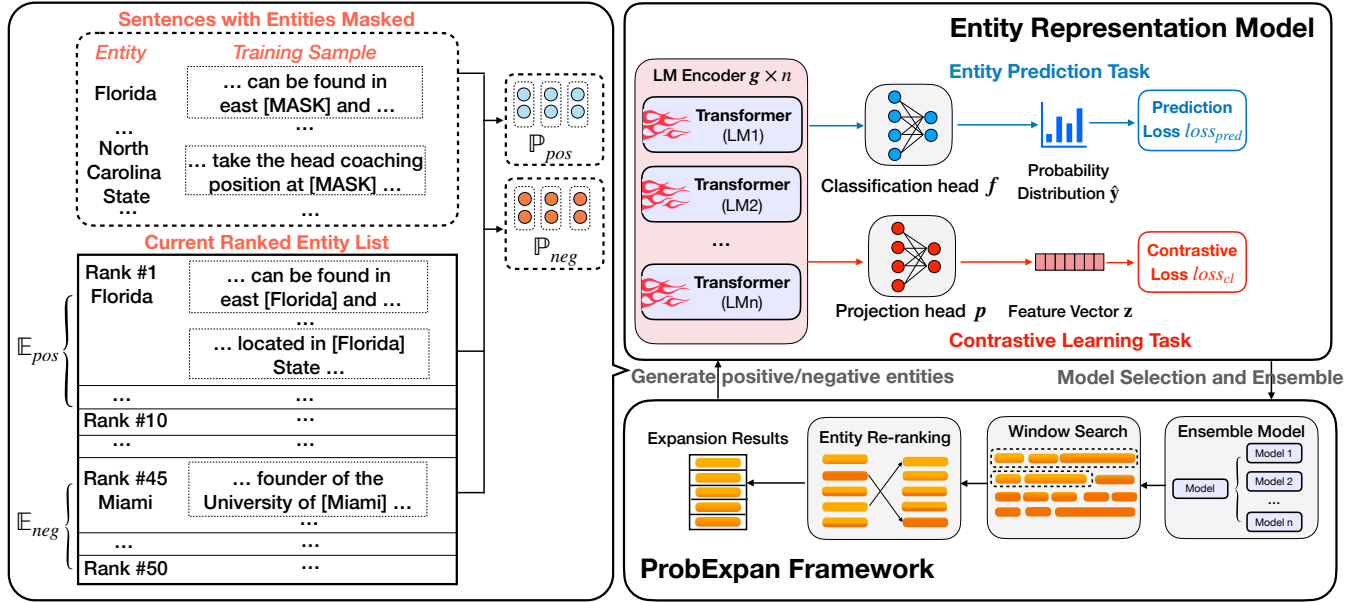


Figure 2: Overview of our proposed method. We jointly train the entity representation model to obtain clearer semantic boundaries through our designed entity prediction task and contrastive learning task. Based on multiple pre-trained entity representation models, we utilize the model selection and ensemble mechanism to avoid the randomness brought by a single model. Two simple yet effective algorithms, namely window-search and entity re-ranking algorithms, are used to search and sort entities to obtain ideal target entities, according to the similarity of probabilistic representations derived from the ensemble model.

entity-level masked language model contains an encoder  $g$  and a classification head  $f$ . To be specific, we initialize the encoder  $g$  with pre-trained parameters of BERT<sub>BASE</sub>, so that grammatical and semantic knowledge learned from large-scale corpus by BERT can be utilized. The classification head  $f$  consists of two linear layers with GeLU activation and a softmax layer. We set biases of the classification head  $f$  to be 0 and initialize weights from Kaiming uniform distribution [9].

Concerning the masked entity prediction pre-training task, for every entity in the vocabulary, we replace its span with [MASK] to get a training sample for all the sentences that it appears. During each training epoch, we restrict the number of samples from every entity to be the average number of samples of all entities, for concern of sample imbalance and the facility of following ensemble learning. We choose the Label Smoothing loss function [27] rather than traditional cross-entropy loss function, so that entities sharing similar semantic meaning with target entity will not be overly suppressed. The prediction loss is defined as:

$$loss_{pred} = -\frac{1}{N} \sum_i^N \sum_j^{V_e} (\mathbb{1}_{j=y_i} (1-\eta) \cdot \log \hat{y}_i[j] + \mathbb{1}_{j \neq y_i} \eta \cdot \log \hat{y}_i[j]), \quad (1)$$

where  $N$  is the size of mini-batch,  $V_e$  is the size of entity vocabulary,  $\eta$  is the smoothing factor and the larger the  $\eta$  is, the higher the smoothness of the label is,  $y_i$  is the index of the entity corresponding to the training sample  $i$ ,  $\hat{y}_i$  is the output of  $f$ .

### 3.2 Contrastive Representation Learning

We apply contrastive learning to refine the semantic space learned by our model so that the representation of the same semantic class entities are pulled closer while the representation of different semantic class entities are pushed outward.

To do this, we firstly generate positive/negative entities for each semantic class from seed sets and previous expansion results. Note that these previous expansion results are from the last iteration, since our expansion framework is an iterative process. Positive entities  $\mathbb{E}_{pos}$  are defined as seed entities or entities that rank higher than a threshold  $thr_{pos}$  in the expanded ranked lists. The entities that lie in a pre-defined interval  $(L_{neg}, U_{neg})$  of the expanded ranked lists are automatically selected as negative entities  $\mathbb{E}_{neg}$ .

$$\mathbb{E}_{pos} = \{e | e \in \mathbb{E}_{seed} \text{ or } \text{rank}(e) < thr_{pos}\}, \quad (2)$$

$$\mathbb{E}_{neg} = \{e | L_{neg} < \text{rank}(e) < U_{neg}\}, \quad (3)$$

where these thresholds are the hyper-parameters for positive/negative entities selection. Additionally, it is worth noting that we determine these thresholds based on a reasonable assuming, i.e., hard negative entities would be ranked close to positive entities during the expansion process. Therefore, we set the  $\mathbb{E}_{neg}$ 's lower bound  $L_{neg}$  a little larger than the size of all positive entities to select negative entities in practice.

Inspired by [21], we design the contrastive learning method which can concentrate on hard negative entities for ESE. Specifically, we initialize our models in the same way as we discuss above, while

attaching an auxiliary projection head  $\mathbf{p}$  on top of the encoder of our model. The projection head  $\mathbf{p}$  maps the final hidden embedding of the masked entity into a normalized feature vector  $\mathbf{z} \in \mathbb{R}^D$ , where  $D$  is the dimension of output. To calculate the contrastive loss, samples from positive/negative entities are paired up to form positive/negative sample pairs:

$$\mathbb{P}_{pos} = \{(\mathbf{x}, \mathbf{x}') | ent(\mathbf{x}) \in \mathbb{E}_{pos}, ent(\mathbf{x}') \in \mathbb{E}_{pos}\}, \quad (4)$$

$$\mathbb{P}_{neg} = \{(\mathbf{x}, \mathbf{x}') | ent(\mathbf{x}) = ent(\mathbf{x}') \in \mathbb{E}_{neg}\}, \quad (5)$$

where  $ent(\mathbf{x})$  indicates the entity corresponding to the training sample  $\mathbf{x}$ . The contrastive loss is then defined as follow:

$$loss_{cl} = - \sum_{i=1}^{2N} \log \frac{S_i^+}{S_i^+ + S_i^-}, \quad (6)$$

$$S_i^+ = e^{\mathbf{z}_i^\top \cdot \mathbf{z}_{j(i)}/t}, \quad (7)$$

$$S_i^- = \max\left(\frac{-(2N-2) \cdot \tau^+ \cdot S_i^+ + \widetilde{S}_i^-}{1 - \tau^+}, e^{\frac{-1}{\tau^+}}\right), \quad (8)$$

$$\widetilde{S}_i^- = \frac{(2N-2) \sum_{k:k \neq i \neq j(i)} e^{(1+\beta)\mathbf{z}_i^\top \cdot \mathbf{z}_k/t}}{\sum_{k:k \neq i \neq j(i)} e^{\beta \mathbf{z}_i^\top \cdot \mathbf{z}_k/t}}, \quad (9)$$

where  $S_i^+/S_i^-$  respectively reflects the similarity between two training samples from the same/different sample pair, and  $j(i)$  indicates that the training samples corresponding to indexes  $i$  and  $j$  can form a positive/negative sample pair, that is,  $(\mathbf{x}_i, \mathbf{x}_j) \in \mathbb{P}_{pos} \cup \mathbb{P}_{neg}$ ,  $N$  is the size of mini-batch,  $\tau^+$  is the class-prior probability which can be estimated from data or treated as a hyper-parameter,  $\beta$  is the hyper-parameter controlling the level of concentration on hard negative samples,  $t$  is the temperature scaling factor which we set as 0.5 in all our experiments. It is noted that the training process alternates between the prediction loss and contrastive loss.

### 3.3 Model Selection and Ensemble

It is reasonable to hypothesize that a model which has learned more common semantic meaning of a class will output more consistent representation of seed entities from that class. Under this hypothesis, we design a scoring function to estimate a model's expansion performance on a semantic class:

$$sco(\theta, cls) = - \frac{\sum_i^M \sum_{j:i \neq j}^M \text{KL\_Div}(r(e_i), r(e_j))}{M * (M - 1)}, \quad (10)$$

$$r(e) = \frac{1}{|\mathbb{S}_e|} \sum_{\mathbf{x} \in \mathbb{S}_e} f(\mathbf{g}(\mathbf{x}|\theta)|\theta), \quad (11)$$

$$M = |\mathbb{E}_{seed}^{cls}|, \quad (12)$$

where  $\theta$  is parameters of the model,  $\mathbb{E}_{seed}^{cls}$  is the set of seed entities of the class,  $\mathbb{S}_e$  is the set of all samples of entity  $e$ ,  $r(e)$  is the probabilistic representation of entity  $e$ ,  $\text{KL\_Div}$  is KL Divergence.

The overall score of a model on a dataset is then defined as the geometric mean of the model's scores on all the classes:

$$\widetilde{sco}(\theta) = - \left| \sqrt[N_{cls}]{\prod_i^{N_{cls}} sco(\theta, cls_i)} \right|. \quad (13)$$

---

#### Algorithm 1 Window Search

---

**Input:** candidate entity list  $L$ ; current set  $L_{cur}$ ; window size  $w$ ; anchor distribution  $\mathbf{d} \in \mathbb{R}^{V_e}$ ; entity representation  $\mathbf{r} \in \mathbb{R}^{V_e}$ ; scaling factor  $\alpha$ ; stage step  $\tau$ ; counter  $c$ .

**Output:** target entity  $e_t$ .

```

1:  $c \leftarrow 0$ ;
2:  $s_t \leftarrow -\infty$ ;
3:  $p \leftarrow \frac{1}{V_e}$ ;
4: for  $e$  in  $L$  do
5:   if  $c \geq w$  then
6:     break
7:    $\mathbf{r} \leftarrow \frac{1}{|\mathbb{S}_e|} \sum_{\mathbf{x} \in \mathbb{S}_e} f(\mathbf{g}(\mathbf{x}))$ ;
8:    $\mathbf{d} \leftarrow [p]^{V_e}$ ;
9:    $\mathbf{d}[\text{index}(e)] \leftarrow \mathbf{r}[\text{index}(e)]$ ;
10:  for  $i$  in  $|L_{cur}|$  do
11:     $\mathbf{d}[\text{index}(e_i)] \leftarrow p * \alpha * 2^{-\lfloor \frac{i}{\tau} \rfloor}$ ;
12:   $\mathbf{d} \leftarrow \text{Softmax}(\mathbf{d})$ ;
13:   $s(e) \leftarrow -\text{KL\_Div}(\mathbf{r}, \mathbf{d})$ ;
14:  if  $s(e) > s_t$  then
15:     $e_t \leftarrow e$ ;
16:     $s_t \leftarrow s(e)$ ;
17:   $c \leftarrow c + 1$ ;
18: return  $e_t$ .
```

---

With this scoring function, we are able to select top-k models  $\Theta_{top}$  from multiple models with only information of seed sets of each class. We ensemble these models as follow:

$$\widetilde{f}(\mathbf{g}(\mathbf{x})) = \frac{1}{|\Theta_{top}|} \sum_{\theta \in \Theta_{top}} f(\mathbf{g}(\mathbf{x}|\theta)). \quad (14)$$

The practical model training process and analysis on model efficiency are described in Appendix A.

### 3.4 Probabilistic Entity Set Expansion

Our proposed ProbExpan is an iterative framework based on the probabilistic representation of entities and entity sets. At the beginning of expansion, we initialize the current set  $L_{cur}$  as the given seed set. In every expansion step, we first calculate the probabilistic representation of current set  $r(L_{cur})$  with our pre-trained ensemble model:

$$r(L_{cur}) = \frac{1}{|L_{cur}|} \sum_{e \in L_{cur}} \frac{1}{|\mathbb{S}_e|} \sum_{\mathbf{x} \in \mathbb{S}_e} \widetilde{f}(\mathbf{g}(\mathbf{x})). \quad (15)$$

$r(L_{cur})$  is essentially the average of predicted entity distributions of all entities in current set, whose dimension is the size of the entity vocabulary. Sorting it and filtering out entities in current set give us a ranked candidate entity list  $L$ .

The window search Algorithm 1 on  $L$  is to expand the target entities of current set. The algorithm judges the quality of a candidate entity by the similarity between its representation  $\mathbf{r} \in \mathbb{R}^{V_e}$  and the anchor distribution  $\mathbf{d} \in \mathbb{R}^{V_e}$  of current set. Therefore, an entity that is not so prominent (i.e., long-tail entity) but shares more similar representation with current set will be expanded in current set. The anchor distribution  $\mathbf{d}$  reflects entity distribution of current set, where seed entities and entities expanded earlier weigh heavier.

Methods	Wiki			APR			SE2		
	MAP@10	MAP@20	MAP@50	MAP@ 10	MAP@20	MAP@50	MAP@10	MAP@20	MAP@50
EgoSet	0.904	0.877	0.745	0.758	0.710	0.570	0.583	0.533	0.433
SetExpan	0.944	0.921	0.720	0.789	0.763	0.639	0.473	0.418	0.341
SetExpander	0.499	0.439	0.321	0.287	0.208	0.120	0.520	0.475	0.397
CaSE	0.897	0.806	0.588	0.619	0.494	0.330	0.534	0.497	0.420
CGExpan	<u>0.995</u>	<u>0.978</u>	0.902	<u>0.992</u>	<u>0.990</u>	0.955	0.601	0.543	0.438
SynSetExpan	0.991	<u>0.978</u>	<u>0.904</u>	0.985	<u>0.990</u>	<b>0.960</b>	<u>0.628</u>	<u>0.584</u>	<u>0.502</u>
ProbExpan	<b>0.995</b>	0.982	0.926	0.993	0.990	0.934	<b>0.683</b>	<b>0.633</b>	<b>0.541</b>
ProbExpan-CN	<b>0.995</b>	<b>0.983</b>	<b>0.929</b>	<b>1.000</b>	<b>0.996</b>	0.955	-	-	-

**Table 1: MAP@K(10/20/50) of different methods. The choices of  $K$  value are exactly following the previous works [23, 34]. All baseline results are directly from other published paper. Note that the class name guidance step in CGExpan is proposed for relatively coarse-grained semantic classes, while the semantic classes of SE2 dataset are more fine-grained, so this method is not very operable on SE2 dataset. We underline the previous state-of-the-art performance on three datasets for convenient comparison.**

The base of it is set as  $\frac{1}{v_e}$ , the average entity prediction probability. To make the anchor distribution robust to candidate entities, the anchor probability of candidate entity is set to be the same as the predicted probability of candidate entity. And the anchor probability of each entity in current set scales over  $p$ , where entities with higher ranks will get larger scale. Note that the anchor distribution  $\mathbf{d}$  is transformed into a probability distribution by Softmax before calculating the KL\_Div.

We increase window size  $w$  according to the current set size, since the anchor distribution will be more concrete as the current set size grows larger:

$$w = w_0 + g * \lfloor \frac{|L_{cur}|}{s} \rfloor, \quad (16)$$

where  $w_0$  is the initial window size,  $g$  is window growing rate,  $s$  is window growing step.

Once expanded set reaches target size  $S_{tgt}$ , we stop the expansion and run the entity re-ranking algorithm. In particular, for every entity  $e$  in the expanded set, we first calculate its score  $s(e)$  in the same way as we do in the window search algorithm. A ranked list  $L_{rank}$  can be constructed according to these scores. The aggregation score of every expanded entity is then calculated as follow:

$$score(e_i) = \sqrt{\frac{1}{i} * \frac{1}{rank(e_i)}}, \quad i = 1...S_{tgt}, \quad (17)$$

where  $i$  is the expand order of entity  $e_i$  in expanded set,  $rank(e_i)$  is the rank of entity  $e_i$  in  $L_{rank}$ .

Sorting the expanded set according to these aggregation scores will get the final expansion results.

## 4 EXPERIMENTS

### 4.1 Experiment Setup

**1. Datasets.** To verify the correctness of our intuition and proposed method, we choose two public datasets widely used in previous work and an additional recently released larger and more challenging dataset [23]:

- (1) **Wiki** and **APR**, which contains 8 and 3 semantic classes respectively. Each semantic class has 5 seed sets and each seed set has 3 queries, following the previous work.
- (2) **SE2**, which contains 60 semantic classes and 1200 seed queries. The scale of dataset shows that SE2 is more challenging. The datasets used in the experiment are detailed in Appendix B.

**2. Compared methods.** We will compare the following ESE methods in our experiments, the implementation details and hyperparameter choices of our experiments are shown in Appendix C:

- (1) **EgoSet** [22]: A multifaceted set expansion system based on skip-gram features, word2vec embeddings and WikiList.
- (2) **SetExpan** [24]: A method iteratively selects context features from the corpus and proposes an ensemble mechanism to rank entities.
- (3) **SetExpander** [17]: A corpus-based model for expanding a seed entity set into a more complete entity set that belong to the same semantic class.
- (4) **CaSE** [33]: A framework that constructs candidate entities with lexical features and ranks candidates using the similarity of distributed representation.
- (5) **CGExpan** [34]: A method that generates the target semantic class name by querying a pre-trained language model and utilizes generated class names to expand new entities.
- (6) **SynSetExpan** [23]: Current state-of-the-art method that jointly conducts two related tasks and utilizes synonym information to improve performance of ESE.
- (7) **ProbExpan**: In our proposed framework, we first apply contrastive learning on entity representation model to obtain better entity semantic representation. Then we use model selection and ensemble to avoid the randomness of the pre-training process. Finally we run two novel algorithms to get expansion results.
- (8) **ProbExpan-CN**: Because our proposed entity representation model is end-to-end trainable, we can combine it with the class name guidance step in CGExpan.

**3. Evaluation Metrics.** The task objective of ESE is to expand a ranked list of entities belong to the same semantic class. Thus,

to evaluate the ranked result, we choose to use the **Mean Average Precision at different top K positions** as:  $\text{MAP@K} = \frac{1}{|Q|} \sum_{q \in Q} \text{AP}_K(L_q, S_q)$ , where  $Q$  is the set of all seed queries and for each query  $q$ , we use  $\text{AP}_K(L_q, S_q)$  to denote the traditional average precision at position  $K$  given a ranked list of entities  $L_q$  and a ground-truth set  $S_q$ . To ensure the fairness of experiment, we are completely consistent with the baseline methods' evaluation metric settings.

## 4.2 Experiment Results

We will first report the overall performance, then analyze and explain the experiment results comprehensively.

**1. Overall Performance.** Table 1 shows the overall performance of different ESE methods. We can see that ProbExpan along with its variant outperform all baselines including current state-of-the-art methods on three datasets, which demonstrates the effectiveness of our proposed method. It is also worth noting that the Wiki and APR are small and relatively easy, the baselines don't leave us much space for improvement. But even so, our methods still perform well compared to the baselines.

Semantic Class	MAP@100
China Provinces	0.824 - 0.728 = 0.096 ↑
Companies	0.969 - 0.950 = 0.019 ↑
Countries	0.930 - 0.941 = -0.011 ↓
Disease	0.959 - 0.948 = 0.011 ↑
Parties	0.948 - 0.913 = 0.035 ↑
Sports Leagues	1.000 - 0.909 = 0.091 ↑
TV Channels	0.888 - 0.875 = 0.013 ↑
US States	0.763 - 0.750 = 0.013 ↑
<b>Overall</b>	<b>0.033</b> ↑

Table 2: The improvement (MAP@100) of ProbExpan based on CGExpan under different classes.

**2. Performance Analysis.** (1) For different datasets, our methods stably perform at a competitive level while existing methods fluctuate fiercely. Especially on SE2, which has more entities and semantic classes, our model's advantage is more obvious. (2) For different semantic classes, Table 2 shows that ProbExpan outperforms previous work under most classes, even though we use more challenging evaluation metric such as MAP@100. (3) For flexibility and expandability, the performance improvement of ProbExpan-CN compared with ProbExpan suggests that our proposed method can be combined with other methods friendly.

## 4.3 Parameter Studies

In Section 3.2, we propose to automatically select negative entities using a pre-defined interval  $(L_{neg}, U_{neg})$ , according to the Equation 3. Furthermore, to select those really hard negative entities as accurately as possible, we will manually ensure that the value of  $L_{neg}$  is slightly larger than the size of positive entities when we determine the values of these two hyper-parameters. Therefore, it is reasonable to suspect that the values of  $(L_{neg}, U_{neg})$  will affect the hardness of the selected negative entities, thereby affecting the performance of the ProbExpan. But we can prove both theoretically

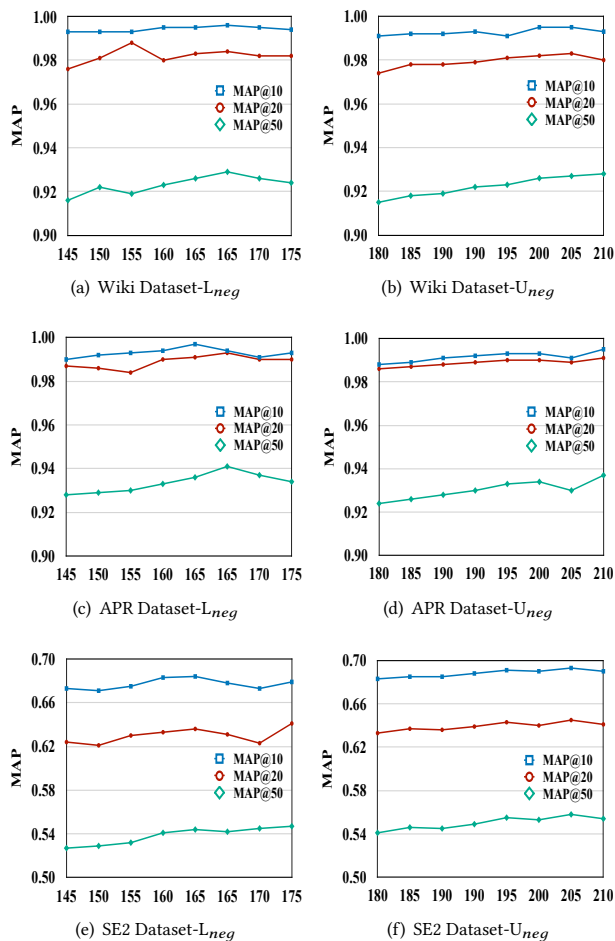


Figure 3: Sensitivity analysis of  $L_{neg} / U_{neg}$  in ProbExpan.

and empirically that such a phenomenon that parameters affect performance does not exist in our proposed framework.

**Theoretically**, even if we set an inappropriate and large  $L_{neg}$ , it will not cause a drop in the overall performance of the ProbExpan, because our proposed contrastive loss can adaptively focus on really hard entities in a training batch. The negative entities that are more similar to the positive entities will receive higher weight when calculating loss through Equation 9. **Empirically**, we carry out the parameter studies as shown in Figure 3 to verify the insensitivity of ProbExpan to these two hyper-parameters. Specifically, we fix one of  $(L_{neg}, U_{neg})$  and change the value of the other, and run the ProbExpan on different datasets to test its performance. From Figure 3, we can see that the performance of our proposed ProbExpan is not very sensitive to their specific values when these two parameters are within a reasonable range, because as  $L_{neg}$  or  $U_{neg}$  changes, the model performance (MAP@K) does not change very significantly. **To sum up**, the values of  $(L_{neg}, U_{neg})$  will indeed determine what entities we select as the hard negative entities, but due to the design of other structures and training strategy of our

Methods	Wiki			APR		
	MAP@10	MAP@20	MAP@50	MAP@10	MAP@20	MAP@50
CGExpan-NoCN	0.968	0.945	0.859	0.909	0.902	0.787
ProbExpan-NoCLEN	0.983	0.974	0.910	0.990	0.977	0.898
ProbExpan-NoEN	0.989	0.980	0.899	0.992	0.981	0.912
ProbExpan-NoCL	0.991	0.980	0.917	<b>0.993</b>	0.984	0.910
ProbExpan	<b>0.995</b>	<b>0.982</b>	<b>0.926</b>	<b>0.993</b>	<b>0.990</b>	<b>0.934</b>

Table 3: Ablation studies of ProbExpan and its variants on two datasets. We arrange the results from top to bottom in the order of increasing components of the model.

model, their values will not affect the overall performance of the model significantly.

#### 4.4 Ablation Studies

To provide a detailed analysis of how our proposed method works on ESE, we perform a series of ablation experiments to see how each component affects the model’s expansion performance. Besides, the ablation results will also provide empirical proofs for our intuitions.

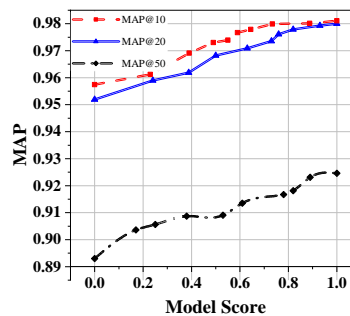
Because the full method of CGExpan leverages some fixed patterns well manually designed by researchers (i.e., Hearst patterns [10]), to ensure ablation studies’ fairness, we will compare ProbExpan’s variants with CGExpan-NoCN [34], which mainly consists of a traditional pre-trained language model such as BERT. The ProbExpan’s variants include:

- (1) ProbExpan-NoCLEN: The ablation of ProbExpan without contrastive learning and model selection and ensemble.
- (2) ProbExpan-NoEN: The ablation of ProbExpan without model selection and ensemble.
- (3) ProbExpan-NoCL: The ablation of ProbExpan without contrastive learning.

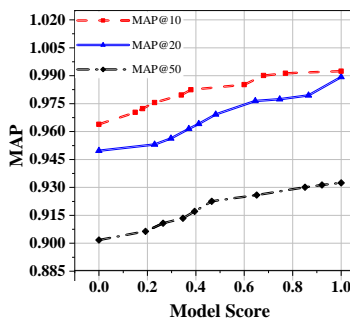
The results of these methods are shown in Table 3.

**1. Can Entity Representation Model Empower ESE?** From Table 3 we can see that ProbExpan-NoCLEN has a great improvement compared to CGExpan-NoCN, especially for the MAP@50. The significant improvement of ProbExpan-NoCLEN indicates the entity-level masked language model can represent entities better. Besides, it is worth noting that the ProbExpan-NoCLEN’s results on APR are better than results on Wiki, which is exactly the opposite of CGExpan-NoCN. Because CGExpan-NoCN incorporates the average BERT representation to select entities and the BERT is pre-trained on Wikipedia corpus which is similar to the corpus of Wiki dataset in ESE. Therefore, CGExpan-NoCN cannot handle other source corpus, which also reflects that the entity representation model we design is not sensitive to the source corpus and has good generalization performance.

**2. Can Contrastive Learning Divide A Clearer Semantic Boundary?** The comparison between ProbExpan-NoEN and ProbExpan-NoCLEN shows that contrastive learning effectively refines the entity representation. According to our observation, previous works such as CGExpan already have competitive performance, the most error-prone case is that they face entities that are semantically ambiguous. This is also the motivation we choose contrastive learning



(a) Wiki Dataset



(b) APR Dataset

Figure 4: Correlation analysis of model score and performance on Wiki and APR datasets.

to handle these hard negative entities. The performance results of Table 3 and the case study in Figure 5 together show that contrastive learning can indeed divide a clearer semantic boundary.

**3. Can Model Selection And Ensemble Strategy Work?** The results about ensemble method in Table 3 show that the model selection and ensemble step we design can bring remarkable improvement. Especially for the ProbExpan’s results, we are pleasantly surprised to find that on the basis of ProbExpan-NoEN, application of model selection and ensemble strategy can still improve further. In addition, to verify the validity of the Equation 13, we analyze the correlation between model score and performance. For the convenience of display, we normalize the model score. The positive

Seed Entity Set	ProbExpan-NoCLEN		ProbExpan-NoEN		ProbExpan	
China Provinces {“Anhui”, “Fujian”, “Hunan”}	1	“Zhejiang”	1	“Jiangsu”	1	“Jiangsu”
	2	“Shandong”	2	“Zhejiang”	2	“Guangdong”
	3	“Henan”	3	“Guangdong”	3	“Zhejiang”
		...		...		...
	30	“Wuhan”	30	“Chongqing”	30	“Chongqing”
	31	“Shanghai”	31	“Hubei province”	31	“Fujian province”
	32	“Liaoning province”	32	“Liaoning”	32	“Guangdong”
	33	“Hangzhou”	33	“Fujian province”	33	“Tianjin”
	34	“Guangzhou”	34	“Harbin”	34	“Liaoning province”
	...		...		...	
Sports Leagues {“Victorian Football Leagues”, “NBA”, “Australian Football League”}	1	“AFL”	1	“AFL”	1	“AFL”
	2	“VFL”	2	“VFL”	2	“VFL”
	3	“NFL”	3	“SANFL”	3	“SANFL”
		...		...		...
	15	“MLB”	15	“Southern Football League”	15	“WAFL”
	16	“major league baseball”	16	“Victorian Premier League”	16	“Victorian Premier League”
	17	“major leagues”	17	“A-league”	17	“League of Ireland”
	18	“national league”	18	“NSL”	18	“NSL”
	19	“American League”	19	“St Kilda Football Club”	19	“national basketball league”
	...		...		...	

Figure 5: Results of two seed entity sets with different semantic classes. We mark the wrong entities in red.

correlation results presented in Figure 4 show that the Equation 13 can effectively evaluate the model.

#### 4.5 Case Studies

We will present different models’ representative expansion cases as further verification of our methods’ advantages.

Figure 5 shows some expansion results of ProbExpan’s variants for several queries from different semantic classes. We see that even though ProbExpan-NoCLEN has achieved very good overall performance (as can be seen from Table 3), it still occasionally has difficulty distinguishing some hard negative samples. For example, municipal administrative regions such as “Wuhan”, “Hangzhou”, and “Guangzhou” are likely to have great similarities in context with provincial administrative regions such as “Shanghai” and “Zhejiang” when training a language model, because they all actually belong to `Location` entities. Therefore, ProbExpan-NoCLEN cannot represent these entities in a more fine-grained manner at the semantic level. As shown in the comparison between ProbExpan-NoCLEN and ProbExpan-NoEN columns of Figure 5, ProbExpan-NoEN can recall more entities belonging to the correct target semantic class. So we can know that contrastive learning can divide a tighter and clearer boundary for the target semantic class through by extending the distance between negative and positive samples and shortening the distance between positive samples in the semantic space.

From the ProbExpan-NoEN column of Figure 5, we can see contrastive learning still can not solve some extreme situations. For example, suppose a person does not have any external background knowledge, then when he/she sees “St Kilda Football Club”, he/she must be easy to literally classify it as `Sports Leagues`. Therefore, we design the model selection and ensemble mechanism to get better expanded entities on the basis of ProbExpan-NoEN and the mechanism’s effectiveness can be reflected from the ProbExpan

column of Figure 5. From the whole Figure 5 we can know that the effect of ProbExpan-NoEN is better than ProbExpan-NoCLEN, and ProbExpan can be further improved based on ProbExpan-NoEN. Such experimental results are in line with our design expectations.

## 5 CONCLUSIONS

In this paper, we introduce to pre-train an entity-level masked language model with the entity prediction task. Then we firstly empower the ESE model to better handle hard negative entities with contrastive learning task. To utilize our pre-trained entity representation model, we propose the ProbExpan, a novel probabilistic ESE framework that consists of two simple yet effective algorithms, namely window-search and entity re-ranking algorithms. In the future, we will further study how to apply our pre-trained ESE model in cross-domain scenarios to better exploit its generalization ability. Combining various domain adaptation methods with our model will be an interesting direction. Moreover, it is also a worthy and promising research direction to study how to automatically measure the hardness of negative entities, so that the really hard negative entities can be better directly selected.

## 6 ACKNOWLEDGEMENTS

This research is supported by the Shenzhen General Research Project (Grant No. JCYJ20190808182805919) and the 173 program (Grant No. 2021-JCJQ-JJ-0029), National Natural Science Foundation of China (Grant No. 6201101015), Beijing Academy of Artificial Intelligence(BAAI), Natural Science Foundation of Guangdong Province (Grant No. 2021A1515012640), the Basic Research Fund of Shenzhen City (Grant No. JCYJ20210324120012033), National Key R&D Program of China (No. 2021ZD0112905) and Overseas Cooperation Research Fund of Tsinghua Shenzhen International Graduate School (Grant No. HW2021008).



## A THE TRAINING PROCESS OF MODEL IN PRACTICE

In practice, the overall training process of entity representation model consists of four phases:

- (1) In the first phase, we train multiple models in parallel with a masked entity prediction task.
- (2) In the second phase, we select top-k models using an algorithm based on the consistency of probabilistic representation of seed set entities, and ensemble them.
- (3) In the third phase, we run the expansion procedure and generate positive/negative entities from the expansion results and seed sets, with which we train multiple models in parallel with the combination of contrastive loss and masked entity prediction loss.
- (4) In the fourth phase, we select and ensemble models in the same way as we do in the second phase.

For the overall model efficiency, since the bottom layers of encoder  $g$  are frozen and all models are trained in parallel, the time cost is close to fine-tuning single BERT. Once training is done, we parallelly use all models to calculate the predicted entity distributions for all samples on the corpus and saved the output into disk, so that models' output can be easily retrieved during model selection and ensemble. Overall, the framework is efficient enough for downstream applications.

## B DATASETS USED IN EXPERIMENTS

We select Wiki and APR datasets which are used in previous work and an recently proposed more challenging dataset, SE2. We use these datasets following previous works to process the corpus. The download links of these datasets are all available in their original papers [23, 34]. The details of these datasets are described as follow:

- (1) **Wiki**, which is from English Wikipedia articles. It mainly contains 8 semantic classes, namely China Provinces, Companies, Countries, Diseases, Parties, Sports Leagues, TV Channels and US States. Each semantic class has 5 queries and each queries has 3 entities as the initial seed entity set.
- (2) **APR**, which is from news articles published by Associated Press and Reuters in 2015. It mainly contains 3 semantic classes, namely Countries, Parties and US States. Each semantic class has 5 queries and each query has 3 entities as the initial seed entities.
- (3) **SE2**, which is from Wikipedia 20171201 dump. It contains 60 major semantic classes and 1200 seed set queries. It is the latest and largest benchmark for ESE.

## C IMPLEMENTATION DETAILS AND HYPER-PARAMETER CHOICES

### C.1 Implementation Details of Baselines

If a baseline method has make its code public available, we directly use the open-source code of this method in our experiments. But if there is no open-source code for a baseline method, we directly report its performance obtained in other published paper [23], because the datasets and evaluation metric used in the original paper are completely consistent with these in our experiments.

### C.2 Implementation Details of ProbExpan

Our proposed entity representation model adopts the Encoder-Decoder architecture, the encoder is a multi-layer bidirectional Transformer [28] that follows the BERT<sub>BASE</sub> setting, and the decoder is a classification head. Specifically, we firstly tokenize the sentences by the WordPiece tokenizer [31]. The input embedding of each token is the sum of its token embedding and position embedding. The sequence of input embedding then passes through 12 stacked bidirectional Transformer blocks with  $H = 768$  hidden dimensions and 12 self-attention heads. It is noted that we freeze part of the layers in our model when pre-train the entity representation model to utilize the pre-training knowledge of BERT.

### C.3 Implementation Details of ProbExpan-CN

The entity probabilistic representation of our model can be easily used in other ESE frameworks. Here we discuss how to combine our entity representation model with the class name guidance step of CGExpan. In the Class-Guided Entity Selection module of CGExpan, score of a candidate entity  $e_i$  is formulated as:

$$score_i = \sqrt{score_i^{loc} * score_i^{glb}}, \quad (18)$$

where  $i$  is the index of the candidate entity in the vocabulary,  $score_i^{loc}$  is related to the guidance class name and we leave it alone,  $score_i^{glb}$  measures the similarity between the candidate entity and entities of current set in the embedding space and is calculated as follow:

$$score_i^{glb} = \frac{1}{|\mathbb{E}_s|} \sum_{e \in \mathbb{E}_s} \cos(\mathbf{V}_{e_i}, \mathbf{V}_e), \quad (19)$$

where  $\mathbb{E}_s$  is a set of entities sampled from current set,  $\mathbf{V}_e$  is the averaged word embedding of entity  $e$  by BERT,  $\cos(\cdot)$  is cosine similarity function. Finally, we replace  $score_i^{glb}$  with  $\overline{score}_i^{glb}$ :

$$\overline{score}_i^{glb} = \frac{1}{|\mathbb{E}_s|} \sum_{e \in \mathbb{E}_s} \frac{1}{|\mathbb{S}_e|} \sum_{x \in \mathbb{S}_e} \overline{f(g(x))}[i]. \quad (20)$$

### C.4 Hyper-parameter Choices

In our experiments, our model is trained by an AdamW optimizer with the betas of (0.9, 0.999), epsilon of  $1e - 6$ , and weight decay of  $1e - 2$ . For different datasets, the choices of other hyper-parameters are shown in Table 4. It is worth noting that we use the standard and general hyper-parameter selection strategy, i.e., grid search the hyper-parameters by model performance on a validation set that does not overlap with the test set.

Hyper-parameter	Wiki	APR	SE2
Frozen layers	11	11	10
Initial $lr_{pred}$	$1e - 5$	$1e - 5$	$2.5e - 6$
Smoothing factor $\eta$	0.075	0.1	0.15
Initial $lr_{cl}$	$1.5e - 5$	$1.5e - 5$	$3.5e - 6$
$thr_{pos}$	12	10	5
$L_{neg}$	170	175	160
$U_{neg}$	200	200	180
$\tau^+$	0.05	0.1	0.01
$\beta$	1	1	2

Table 4: The hyper-parameter settings in experiments.

## REFERENCES

- [1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv preprint arXiv:2002.05709* (2020).
- [2] Zhe Chen, Michael Cafarella, and HV Jagadish. 2016. Long-tail vocabulary dictionary extraction from the web. In *Proceedings of the Ninth ACM international conference on Web search and data mining*. 625–634.
- [3] James R Curran, Tara Murphy, and Bernhard Scholz. 2007. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, Vol. 6. Citeseer, 172–180.
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [5] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *arXiv preprint arXiv:2104.08821* (2021).
- [6] Yoav Goldberg and Omer Levy. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014).
- [7] Sonal Gupta and Christopher Manning. 2014. Improved Pattern Learning for Bootstrapped Entity Extraction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Ann Arbor, Michigan, 98–108. <https://doi.org/10.3115/v1/W14-1611>
- [8] R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, Vol. 2. 1735–1742. <https://doi.org/10.1109/CVPR.2006.100>
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
- [10] Marti A. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*. <https://www.aclweb.org/anthology/C92-2082>
- [11] Jiaxin Huang, Yiqing Xie, Yu Meng, Jiaming Shen, Yunyi Zhang, and Jiawei Han. 2020. Guiding Corpus-Based Set Expansion by Auxiliary Sets Generation and Co-Expansion. In *Proceedings of The Web Conference 2020 (Taipei, Taiwan) (WWW '20)*. Association for Computing Machinery, New York, NY, USA, 2188–2198. <https://doi.org/10.1145/3366423.3380284>
- [12] Prateek Jindal and Dan Roth. 2011. Learning from negative examples in set-expansion. In *2011 IEEE 11th International Conference on Data Mining*. IEEE, 1110–1115.
- [13] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362* (2020).
- [14] Taek Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-Guided Contrastive Learning for BERT Sentence Representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 2528–2540. <https://doi.org/10.18653/v1/2021.acl-long.197>
- [15] Guy Kushilevitz, Shaul Markovitch, and Yoav Goldberg. 2020. A Two-Stage Masked LM Method for Term Set Expansion. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 6829–6835. <https://doi.org/10.18653/v1/2020.acl-main.610>
- [16] Yinghui Li, Qingyu Zhou, Yangning Li, Zhongli Li, Ruiyang Liu, Rongyi Sun, Zizhen Wang, Chao Li, Yunbo Cao, and Hai-Tao Zheng. 2022. The Past Mistake is the Future Wisdom: Error-driven Contrastive Probability Optimization for Chinese Spell Checking. *arXiv preprint arXiv:2203.00991* (2022).
- [17] Jonathan Mamou, Oren Perer, Moshe Wasserblat, Alon Eirew, Yael Green, Shira Guskin, Peter Izsak, and Daniel Korat. 2018. Term Set Expansion based NLP Architect by Intel AI Lab. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Brussels, Belgium, 19–24. <https://doi.org/10.18653/v1/D18-2004>
- [18] Tara McIntosh. 2010. Unsupervised Discovery of Negative Categories in Lexicon Bootstrapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Cambridge, MA, 356–365. <https://aclanthology.org/D10-1035>
- [19] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.
- [20] Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. ERICA: Improving Entity and Relation Understanding for Pre-trained Language Models via Contrastive Learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 3350–3363. <https://doi.org/10.18653/v1/2021.acl-long.260>
- [21] Joshua Robinson, . Chuang, Ching-Yao, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive Learning with Hard Negative Samples. *International Conference on Learning Representations* (2021).
- [22] Xin Rong, Zhe Chen, Qiaozhu Mei, and Eytan Adar. 2016. EgoSet: Exploiting Word Ego-Networks and User-Generated Ontology for Multifaceted Set Expansion. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (San Francisco, California, USA) (WSDM '16)*. Association for Computing Machinery, New York, NY, USA, 645–654. <https://doi.org/10.1145/2835776.2835808>
- [23] Jiaming Shen, Wenda Qiu, Jingbo Shang, Michelle Vanni, Xiang Ren, and Jiawei Han. 2020. SynSetExpan: An Iterative Framework for Joint Entity Set Expansion and Synonym Discovery. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 8292–8307. <https://doi.org/10.18653/v1/2020.emnlp-main.666>
- [24] Jiaming Shen, Zeqiu Wu, Dongming Lei, Jingbo Shang, Xiang Ren, and Jiawei Han. 2017. Setexpan: Corpus-based set expansion via context feature selection and rank ensemble. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 288–304.
- [25] Bei Shi, Zhenzhong Zhang, Le Sun, and Xianpei Han. 2014. A Probabilistic Co-Bootstrapping Method for Entity Set Expansion. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 2280–2290. <https://www.aclweb.org/anthology/C14-1215>
- [26] Chuan Shi, Jiayu Ding, Xiaohuan Cao, Linmei Hu, Bin Wu, and Xiaoli Li. 2021. Entity set expansion in knowledge graph: a heterogeneous information network perspective. *Frontiers of Computer Science* 15, 1 (2021), 1–12.
- [27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.
- [28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).
- [29] Dong Wang, Ning Ding, Piji Li, and Haitao Zheng. 2021. CLINE: Contrastive Learning with Semantic Negative Examples for Natural Language Understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 2332–2342. <https://doi.org/10.18653/v1/2021.acl-long.181>
- [30] Richard C Wang, Nico Schlaefer, William Cohen, and Eric Nyberg. 2008. Automatic set expansion for list question answering. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. 947–954.
- [31] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).
- [32] Lingyong Yan, Xianpei Han, Le Sun, and Ben He. 2019. Learning to bootstrap for entity set expansion. 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)*. 292–301.
- [33] Puxuan Yu, Zhiqi Huang, Raziheh Rahimi, and James Allan. 2019. Corpus-Based Set Expansion with Lexical Features and Distributed Representations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (Paris, France) (SIGIR '19)*. Association for Computing Machinery, New York, NY, USA, 1153–1156. <https://doi.org/10.1145/3331184.3331359>
- [34] Yunyi Zhang, Jiaming Shen, Jingbo Shang, and Jiawei Han. 2020. Empower Entity Set Expansion via Language Model Probing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 8151–8160. <https://doi.org/10.18653/v1/2020.acl-main.725>
- [35] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1441–1451. <https://doi.org/10.18653/v1/P19-1139>