Exploratory Neural Relation Classification for Domain Knowledge Acquisition

Yan Fan, Chengyu Wang, Xiaofeng He*
School of Computer Science and Software Engineering,
East China Normal University, Shanghai, China
{eileen940531,chywang2013}@gmail.com, xfhe@sei.ecnu.edu.cn

Abstract

The state-of-the-art methods for relation classification are primarily based on deep neural networks. This supervised learning method suffers from not only limited training data, but also the large number of low-frequency relations in specific domains. In this paper, we propose an exploratory relation classification method for domain knowledge harvesting. The goal is to learn a classifier on pre-defined relations while discovering new relations expressed in texts. A dynamically structured neural network is introduced to classify entity pairs to a continuously expanded relation set. We further propose the similarity sensitive Chinese restaurant process to discover new relations. Experiments conducted on a large corpus show that new relations are discovered with high precision and recall, illustrating the effectiveness of our method.

Title and Abstract in Chinese

基于领域知识获取的探索式神经网络关系分类

近几年来，关系分类任务主要利用神经网络模型来自动学习复杂的特征。然而这类监督式学习方法的局限性在于，首先它需要大量训练数据，其次特定领域存在长尾的低频关系无法被有效地预定义。在本论文中，我们提出了基于领域知识获取的探索式关系分类任务。它的目标在于学习预定义关系的分类器，同时从文本中发现新的语义关系。我们提出了一个动态结构的神经网络，它可以对持续扩充的关系集进行分类。我们进一步提出了相似度敏感的中餐馆过程算法，用于发现新关系。基于大语料库上的实验证明了该神经网络的分类效果，同时新发现的关系也有较高的准确率和召回率。

1 Introduction

Relation classification assigns semantic relation labels to entity pairs in texts. Besides traditional feature-based (Kambhatla, 2004) and kernel-based (Bunescu and Mooney, 2005) approaches, neural networks (NNs) are introduced to harvest relational facts in recent years. Classical architectures include convolutional neural networks (CNNs) (Zeng et al., 2014), recurrent neural networks (RNNs) (Xu et al., 2015), etc. However, above methods are still insufficient for domain knowledge acquisition due to two challenges: (i) most domain entities rarely occur in the corpus, hence pattern-based methods easily suffer from the feature sparsity problem; (ii) a domain knowledge graph tends to be incomplete w.r.t. relation labels and facts (Fan et al., 2017). As a result, unlabeled entity pairs are likely to be wrongly forced into existing relation labels by distant supervision approaches, instead of their true, unknown labels.

In this paper we propose a method, Exploratory Relation Classification (ERC), to populate domain knowledge graphs automatically. The goal of ERC is to not only classify entity pairs into a finite set of pre-defined relations, but also simultaneously discover previously unseen relation types and their respective instances from plain texts with high confidence. The resulting numbers of relation types and facts in the domain knowledge graph grow continuously.

*Corresponding author.

Proceedings of the 27th International Conference on Computational Linguistics, pages 2265–2276
Santa Fe, New Mexico, USA, August 20-26, 2018.

To solve ERC problem, we propose a Dynamic Structured Neural Network (DSNN). In the network, the convolutional and recurrent network units are employed to model local information such as syntactic and lexical features from the sentence level. For the context sparsity problem (Challenge (i)) an embedding layer is introduced to encode corpus-level semantic features of domain entities. The output classes of DSNN can be automatically expanded during the iterative training process, in order to classify entity pairs to both known and newly discovered relations. To explore new relations and corresponding instances, we introduce the similarity sensitive Chinese restaurant process (ssCRP) to generate clusters of entity pairs from unlabeled data that can not be classified into any known relations (Challenge (ii)). We can see that the DSNN only takes the training data of known relations directly derived from a domain knowledge graph without requiring any training data for new relations. Therefore, our approach is a very weakly supervised one with minimal human intervention.

The rest of the paper is organized as follows. Section 2 summarizes the related work and discusses relations between existing methods and ours. The detailed approach is introduced in Section 3, Section 4 and Section 5, with experiments presented in Section 6. Finally, we conclude this paper in Section 7.

2 Related Work and Discussion

In NLP research, feature-based (Kambhatla, 2004) and kernel-based (Bunescu and Mooney, 2005) methods have been proposed to utilize lexical, syntactic and semantic features for relation classification. In recent years, neural network based approaches are introduced to improve the performance. Word embeddings of entities are frequently used as inputs, instead of one-hot representations (Mikolov et al., 2013; Pennington et al., 2014). As shown in (Wang and He, 2016; Wang et al., 2017), word embeddings can be used for relation prediction. For neural network architectures, CNNs are capable of learning consecutive contexts of entities as local lexical features (Zeng et al., 2014; Vu et al., 2016), while RNNs exploit syntactic features by modeling long-term dependencies of sentences via memory units. The model of Long Short Term Memories (LSTMs) (Hochreiter and Schmidhuber, 1997) is an effective variant of RNNs to encode the shortest dependency path (SDP) between entity pairs (Xu et al., 2015; Shwartz et al., 2016). More recent work focuses on the design of integrated architecture, combining both CNNs and RNNs. Cai et al. (2016) propose a bidirectional recurrent CNN to model the directional information along the SDP forwards and backwards. The integrated neural network proposed by Raj et al. (2017) also experiments with attention-based pooling strategy for biomedical texts. Our model provides a more intuitive representation by feature concatenation instead of network layering, enabling the adaption of dynamic changes as new relations are discovered in iterations.

For relation discovery, one way is to formulate the task as open relation extraction (OpenRE) (Banko et al., 2007). Typical OpenRE systems include TextRunner (Banko et al., 2007), WOE (Wu and Weld, 2007), ReVerb (Etzioni et al., 2011), etc. Unlike approaches confined to a fixed set of pre-defined relations, these systems explore relations in a more general way. Contrary to the fact that the contexts for domain knowledge are sparse, the underlying idea of OpenRE is to identify phrases from sentences that are able to indicate unknown relations based on data redundancy. Therefore, OpenRE methods are not suitable for domain-specific knowledge harvesting within a limited text corpus. Similarly, Riedel et al. (2013) learn universal schemas by matrix factorization without pre-defined relations. An alternative way is to use clustering algorithms. Compared to standard clustering algorithms (e.g., KMeans), non-parametric Bayesian models (Rasmussen, 1999) are more suitable in our scenario, because they can automatically learn the number of clusters. Studies afterwards exploit data features for spatial and temporal data, such as distance dependent CRP (ddCRP) (Blei and Frazier, 2010), etc.

Our task is also related to a typical case of multi-class semi-supervised learning when unlabeled data contains unknown classes. We suggest that existing paradigms for this problem are not much useful for domain knowledge acquisition. Generally, distant supervision (Mintz et al., 2009) is employed to generate training data by aligning knowledge bases with free texts. But its strong alignment assumption may lead to noisy annotation results (Zhang and Wang, 2017). Traditional approaches assign new class labels to part of the unlabeled data on condition that none of the existing classes can fit the data well (Nigam et al., 2000; Dalvi et al., 2013). Therefore, considering the sparsity of domain knowledge, these methods applied on such corpus will introduce plenty of small classes, which are in fact noises and thus meaningless. In contrast, our ssCRP-based approach works more effectively since we only discover one large class with confident instances in each iteration. Hence such noises in the data will not be forced to generate small classes and are automatically discarded when the last iteration stops.

3 Dynamic Structured Neural Network for ERC

In this section, we present the definition of the task ERC and a high level introduction of the DSNN approach.

3.1 Task Definition

Before introducing the training process of DSNN, we first provide some preliminaries. Denote entity pairs as $X^l = \{(e_1, e_2)\}$ with corresponding labels Y^l, and unlabeled entity pairs as $X^u = \{(e_1, e_2)\}$. Exploratory Relation Classification (ERC) task is defined as follows:

Definition 1. *Given labeled data (X^l, Y^l) and unlabeled data X^u, the goal of ERC is to train a model to predict the relations for entity pairs in X^u with $K + n$ output labels, where K denotes the number of pre-defined relations in Y^l, and n is the number of newly discovered relations which is unknown.*

3.2 An Overview of DSNN

Algorithm 1 shows the iterative training process of DSNN. In each iteration, the algorithm tries to detect a new relation from unlabeled data based on ssCRP, and expands the neural network structure to perform relation classification over existing and new relations. Briefly, it consists of three modules: base neural network training[1], relation discovery and relation prediction.

Algorithm 1 DSNN Training Process

Input: Entity pairs X^l and their labels Y^l, unlabeled entity pairs X^u, pre-defined relation set $R_{old} = \{r_1, \ldots, r_K\}$
Output: New relation set $R_{new} = \{r_{K+1}, \ldots, r_{K+n}\}$, populated labeled entity pairs X^l and their labels Y^l

```
Initialize R_new = R_old, t = 1
while no new relations can be discovered do
    // Base neural network training
    Train base neural network NN_t with X^l and Y^l
    // Relation discovery
    Generate candidate clusters {C_1, ..., C_m} = ssCRP(X^u, X^l)
    C* = PickTheBestCluster(C_1, ..., C_m)
    r* = MapToRelation(C*)
    if r* ∉ R_old then
        R_new = R_new ∪ {r*}
    end if
    Label all entity pairs in C* with relation r* to be Y*
    X^l = X^l ∪ C*, Y^l = Y^l ∪ Y*, X^u = X^u \ C*
    // Relation prediction
    for each (e_1, e_2) in X^u do
        Pr(r|e_1, e_2, NN_t) = PredictRelDistribution(e_1, e_2, R_old, NN_t)
        if NotNearUniform(Pr(r|e_1, e_2, NN_t)) then
            Label (e_1, e_2) with relation argmax_{r*} Pr(r|e_1, e_2, NN_t) to be Y*
            X^l = X^l ∪ {(e_1, e_2)}, Y^l = Y^l ∪ Y*, X^u = X^u \ {(e_1, e_2)}
        end if
    end for
    R_old = R_new, t = t + 1
end while
return R_new, X^l, Y^l
```

In each iteration t, the first step is to train base neural network NN_t with training data X^l and Y^l. Note that in the first iteration, model NN_t is trained fully supervised. When $t > 1$, NN_t is trained semi-supervisedly, utilizing both labeled training data and new relations discovered in previous $t - 1$ iterations.

[1] In this paper, we refer to the structure of DSNN without the output layer expansion step as "base neural network". In each iteration, the training process of base neural network is the same despite the number of output units.

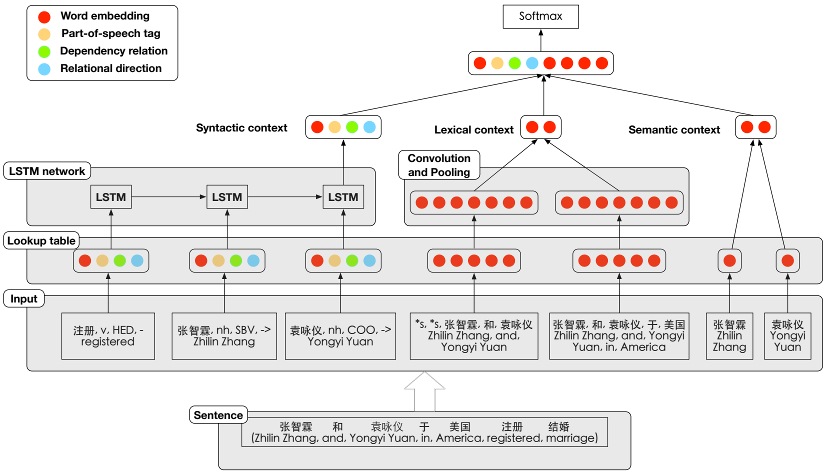

Figure 1: The architecture of the base neural network with k=2.

After NN_t is trained, ssCRP is employed to generate m clusters from X^u (i.e., $C_1, \ldots, C_m$), where each cluster represents the collection of entity pairs that are most likely to share the same underlying relation. We select the "best" cluster C^* from $C_1, \ldots, C_m$ based on the size and intra-cluster similarity, and map it to the relation r^*. r^* could either be one of the pre-defined relations, or a newly discovered relation. Either way, we update X^l and Y^l with C^* labeled as r^*, and remove them from X^u.

For unlabeled entity pairs $(e_1, e_2) \in X^u$, we use model NN_t to predict the probability distribution $\Pr(r|e_1, e_2, NN_t)$ over all possible relations. If the distribution is not "near uniform", the model is confident to predict the relation $r^* = \operatorname{argmax} \Pr(r|e_1, e_2, NN_t)$. Hence, (e_1, e_2) will be labeled as r^* and added to labeled data (i.e., X^l and Y^l). Here, we only add unlabeled data with confident predictions to the training set, in order to avoid error propagation in a semi-supervised learning environment.

After the execution of the above three modules in one iteration, if a new relation is found with its seed relation instances (i.e., entity pairs), the structure of the neural network retrained in the next iteration will be adjusted dynamically with parameters updated.

4 The Architecture of the Base Neural Network

Base neural network takes texts with annotated entity pairs as input and classify them to a fixed relation set. It learns representation of texts by exploiting the following three contexts: (i) syntactic contexts; (ii) lexical contexts; and (iii) global semantic contexts. The overall architecture is illustrated in Fig. 1.

Syntactic contexts. Previous work exploits the shortest dependency path (SDP) (Bunescu and Mooney, 2005) to capture predicate-argument sequences (Cai et al., 2016; Xu et al., 2015; Shwartz et al., 2016). However, SDP can not handle the dependency relation properly between e_1 and e_2 due to the lack of the associated predicate. Consider the example in Fig. 2. Entities "张智霖 (Zhilin Zhang)" and "袁咏仪 (Yongyi Yuan)" are in the coordinate dependency relation via a direct arc. Thus, the SDP between two entities is "张智霖 (Zhilin Zhang) → 袁咏仪 (Yongyi Yuan)", which contains not much useful information. In this paper, we design a structure to capture the root verb along the dependency path, namely root augmented dependency path (RADP):

Definition 2. *An RADP is the combination of two subpaths derived from the dependency graph, one from e_1 to the root verb, and the other one from the root to e_2.*

Here, the RADP is "注册 (registered) → 张智霖 (Zhilin Zhang) → 袁咏仪 (Yongyi Yuan) ". Compared to SDP, it includes the extra root verb "注册 (registered)", a crucial complement indicating the relation "配偶 (spouse)" holds between two entities.

In the base neural network, each node along the RADP is considered as a syntactic unit (Shwartz et al., 2016) with four parts: 1) word embedding; 2) POS tag; 3) dependency relation, which is the label on the arc from the governing node; and 4) relational direction, one of the three labels "→", "←" and "—". Thus we present a syntactic unit as the concatenation of four parts: $\vec{v}_{syn} = [\vec{v}_{word}, \vec{v}_{pos}, \vec{v}_{rel}, \vec{v}_{dir}]$

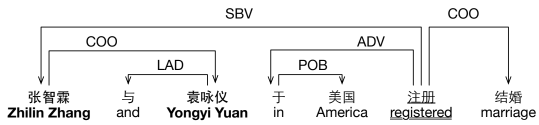

Figure 2: An example of the dependency graph.

where $\vec{v}_{word}$, $\vec{v}_{pos}$, $\vec{v}_{rel}$, $\vec{v}_{dir}$ are the embeddings of the word, POS tag, dependency relation and relational direction, respectively. $[\cdot]$ is the concatenation of embeddings. These embeddings are initialized as one-hot vectors except for $\vec{v}_{word}$, pre-trained on a large corpus.

We employ an LSTM network to handle the long-term dependencies. For an RADP with sequence $w_1, \ldots, w_j$, the corresponding embeddings $\vec{v}_{syn_1}, \ldots, \vec{v}_{syn_j}$ are fed into an LSTM network. The output vector $\vec{v}_{lstm}$ is the context vector modeling the syntactic features of the whole sequence.

Lexical contexts. Specific patterns around target entities can imply semantic relations between entity pairs (Hearst, 1992). Similar to previous research (Zeng et al., 2014; Vu et al., 2016), given a window size k, we define the lexical context of an annotated entity as k words ahead and k words following. Formally, the lexical context vector $\vec{v}_{con}$ is represented as follows: $\vec{v}_{con} = [\vec{v}_{w_{i-k}}, \ldots, \vec{v}_{w_{i-1}}, \vec{v}_{e_i}, \vec{v}_{w_{i+1}}, \ldots, \vec{v}_{w_{i+k}}]$ where $\vec{v}_{e_i}$ and $\vec{v}_{w_j}$ are embeddings of entity and context word at index i and j.

For entities e_1 and e_2 in the pair, $\vec{v}_{con_{e_1}}$ and $\vec{v}_{con_{e_2}}$ are fed into CNN units, respectively. After a convolutional layer and a max pooling layer, the resulting vectors are concatenated as $\vec{v}_{cnn}$.

Semantic contexts. The context-sparsity problem of domain knowledge motivates us to explore semantic contexts in the global corpus, instead of being limited to local features. Based on the distributional hypothesis, we train a Skip-gram model (Mikolov et al., 2013) to learn the distributional representations of words in a large corpus. For entity pair e_1 and e_2, the embeddings $\vec{v}_{e_1}$ and $\vec{v}_{e_2}$ are further concatenated to build the final representation of semantic contexts: $\vec{v}_{emb} = [\vec{v}_{e_1}, \vec{v}_{e_2}]$.

Prediction. The final context representation for prediction is the concatenation of three resulting vectors $\vec{v}_f = [\vec{v}_{lstm}, \vec{v}_{cnn}, \vec{v}_{emb}]$. Then we apply a *softmax* layer on $\vec{v}_f$ to predict the class distribution y: $\vec{y} = \text{softmax}(W \cdot \vec{v}_f + b)$ where W is the transformation matrix and b is the bias vector. The prediction of base neural network is the relation whose probability is maximal in $\vec{y}$.

5 Relation Discovery

To expand the output layer of base neural network, a clustering algorithm ssCRP is proposed to explore new relations from unlabeled data. In this part, we first introduce how the network expands, and briefly cover the basics of CRP, followed by the algorithm ssCRP in detail.

5.1 Network Expansion

After training base neural network, we take its final hidden layer as the representation for each entity pair, since it contains all high-level context features. The representations used as the input of ssCRP should be close to each other in the embedding space, if the same relation holds for these pairs.

In iteration t, ssCRP and the table selection process (to be discussed later) generate a cluster C^* from unlabeled data. Let r^* be the mapping relation that entity pairs in C^* hold, $\{r_1, \ldots, r_{K+l}\}$ be the K pre-defined relations and l new relations discovered in $t-1$ iterations. If $r^* \notin \{r_1, \ldots, r_{K+l}\}$, a new relation is found and base neural network trained in the next iteration $t+1$ expands its output layer with an extra class; otherwise, it remains the same with $K+l$ output classes. Fig 3 shows the base neural network and its expansion network.

5.2 Similarity Sensitive Chinese Restaurant Process

Preliminaries. The Chinese restaurant process (CRP) is a stochastic process, which groups customers into random tables where they sit (Aldous, 1985). Assume N_p denotes the number of customers sitting at table p, z_i denotes the index of the table where the i-th customer sits, and vector $\vec{z}_{-i}$ represents the

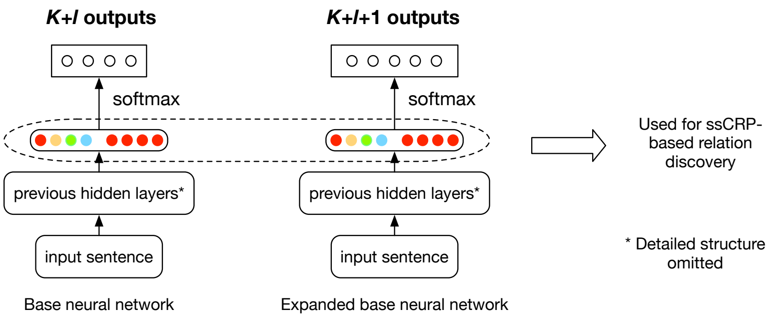

Figure 3: Base neural network and the expanded network.

table assignment for other customers. The distribution over table assignments is as follows:

$$\Pr(z_i = p \mid \vec{z}_{-i}, \alpha) \propto \begin{cases} N_p & \text{if } p \leq K \\ \alpha & \text{if } p = K+1 \end{cases} \quad (1)$$

where α is a scaling parameter and K is the number of occupied tables.

ssCRP. Although CRP can be applied for clustering in a non-parametric Bayesian fashion, it does not consider the similarity of relation instances. We observe that some extensions of CRP (e.g., ddCRP (Blei and Frazier, 2010)) leverage the distances between data points but they do not consider the iterative cluster generation process with pre-defined clusters. In this paper, we propose a similarity-based clustering algorithm especially designed for ERC, called similarity sensitive Chinese restaurant process (ssCRP).

In ssCRP, we turn the problem of table assignment into customer assignment. Unlike the CRP and ddCRP, we accommodate labeled data by initializing tables with K pre-defined classes. Customer i has the choice of sitting next to any customer j based on similarity, which leads to three cases: (i) i joins the table where j sits; (ii) i and an upcoming j generates a new table[2]; and (iii) i sits at an empty table alone. Case (i) is specifically introduced for task ERC due to the initialization of existing tables. Each table in case (i) can be represented as a virtual customer averaged from all its seated customers.

Denote $\eta = \{S, N, \alpha, \beta\}$ as the set of hyperparameters, where S is a similarity matrix between all customers, $N = (N_1, \cdots, N_K)$ is the size vector of existing K tables, α is the scaling parameter and β is a parameter balancing the weight of table size. The distribution of customer assignment c_i is:

$$\Pr(c_i = j \mid \eta) \propto \begin{cases} \alpha & \text{if } j \text{ is customer } i \text{ itself} \\ g(s_{ij}) & \text{if } j \text{ is an upcoming customer} \\ g(s_{ij})(1 + \beta \lg N_p) & \text{if } j \text{ is averaged from table } p \end{cases} \quad (2)$$

where g is a function modeling the cosine similarity s_{ij} between customer i and j. We design a magnifying function $g(s_{ij}) = -1/\ln s_{ij}$ where $s_{ij} > 0$, to increase the dissimilarity of inputs. The first two cases in Eq. (2) are derived from CRP. The third case gives existing tables additional weights to avoid generating too many small clusters. An alternative measuring system is to use the distance function $d_{ij} = 1 - s_{ij}$ and to apply different non-increasing functions to mediate the distances between customers. Given a parameter a, typical decay functions like window decay $f(d) = 1[d < a]$, exponential decay $f(d) = \exp(d/a)$ and logistic decay $f(d) = \exp(-d + a)/(1 + \exp(-d + a))$ are provided as alternatives (Blei and Frazier, 2010).

ssCRP-based relation discovery. The overall sampling and table selection process is illustrated in Fig. 4. In Step 1, we initialize fixed tables with classification results of labeled data. In Step 2, ssCRP draws customer assignments based on Gibbs sampling. Denote C_p as the collection of entity pairs w.r.t table p. Given the hyperparameter set η, the likelihood function of table p denoted by C_p is calculated as follows: $\Pr(C_p \mid \eta) = \prod_{i=1}^{|C_p|} \Pr(c_i^p \mid \eta)$ where c_i^p represents the customer assignment that customer i sits at table p.

[2]If the assignment of j is generated following i, then j is an upcoming customer w.r.t. i.

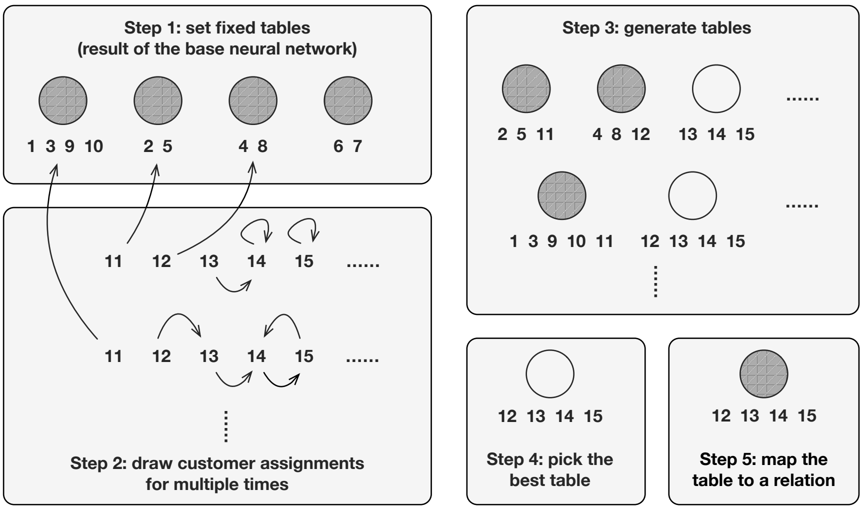

Figure 4: Relation discovery based on ssCRP. Circles with shadow are existing relations, while plain circles are new tables surrounded with customers representing entity pairs.

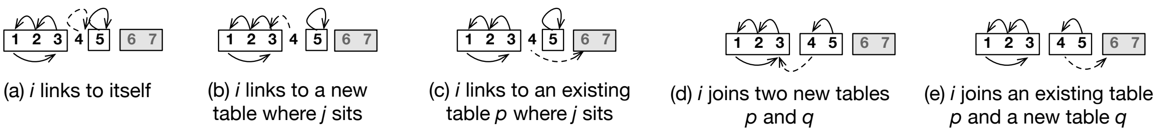

Figure 5: Five cases of customer assignment with $i = 4$ based on Gibbs sampling. Grey boxes are existing tables while white ones are generated new tables.

Given the rest of customer assignments $\mathbf{c}_{-i}$, customer assignment c_i can be divided into five cases (Blei and Frazier, 2010), generated by Gibbs sampling as Fig. 5 shows:

$$\Pr(c_i = j \mid \mathbf{c}_{-i}, \eta) \propto \begin{cases} \alpha & \text{(a)} \\ g(s_{ij}) & \text{(b)} \\ g(s_{ij})(1 + \beta \lg N_p) & \text{(c)} \\ g(s_{ij}) \frac{\Pr(C_p \cup C_q|\eta)}{\Pr(C_p|\eta)\Pr(C_q|\eta)} & \text{(d)} \\ g(s_{ij})(1 + \beta \lg N_p) \frac{\Pr(C_p \cup C_q|\eta)}{\Pr(C_p|\eta)\Pr(C_q|\eta)} & \text{(e)} \end{cases} \tag{3}$$

After this process, all customers are connected together with direct links from one to another or self-loops. New tables can be automatically generated by connected customers while existing tables are extended with new customers in Step 3. In our implementation, we run Gibbs sampling for multiple times to select the best configuration (i.e., largest likelihood), to enhance the fault tolerance.

Step 4 calculates a score for each table p considering pairwise similarities and table size, defined as:

$$score(p) = \frac{\lg N_p}{2\binom{N_p}{2}} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} s_{ij}(i \neq j) \tag{4}$$

where $\frac{1}{2\binom{N_p}{2}} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} s_{ij}(i \neq j)$ is the average pairwise similarity of instances in table p. $\lg N_p$ gives additional weight to large tables. It means the best table has similar relation instances and large size. The table with the highest $score(p)$ will be the selected cluster C^*. The last step (i.e., Step 5) maps the "best" cluster C^* to a semantic relation r^* with a proper relation predicate. The relation can be either a new one with confident seeds (i.e., entity pairs) or an existing relation extended with new entity pairs.

Predefined relations	执导 (directing)	演唱 (singing)	主演 (starring)	配偶 (spouse)
# Instances	633	648	1609	590

Table 1: The statistics of pre-defined relations extracted from (Fan et al., 2017).

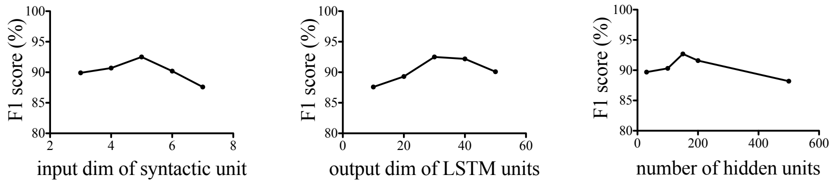

Figure 6: The effects of hyperparameters of the base neural network.

5.3 Relation Prediction

New relations generated via ssCRP only contain a small number of seed instances. To populate relations, part of unlabeled entity pairs will be assigned with labels based on model prediction, and added to the training set. To avoid error propagation, the algorithm requires that only if the prediction of DSNN is highly confident for an entity pair, can we add it to the training set.

Consider the distribution $[\Pr(r_1|e_1,e_2),\ldots,\Pr(r_{K+l}|e_1,e_2)]$ for entity pair (e_1,e_2). If it is "near uniform", none of the existing relation labels are confident enough to be the perfect fit. Goodness-of-fit tests are able to determine if the prediction follows uniform distribution (e.g. Kolmogorov-Smirnov test), but experiments show they do not work in this scenario (refer to Section 6.3). In this paper, we define a heuristic "Max-SecondMax" value to estimate the confidence score of a prediction:

$$conf(e_1,e_2) = \frac{\max([\Pr(r_1|e_1,e_2),\ldots,\Pr(r_{K+l}|e_1,e_2)])}{\text{secondMax}([\Pr(r_1|e_1,e_2),\ldots,\Pr(r_{K+l}|e_1,e_2)])} \tag{5}$$

where secondMax$(\cdot)$ is the second largest value of probabilities. If $conf(e_1,e_2)$ is no less than a given threshold τ, the label with the highest probability is indeed a convincing prediction. Such entity pairs will be added to labeled data together with their labels.

6 Experiments

In this section, we conduct extensive experiments for ERC task to evaluate our method and compare it with state-of-the-art approaches.

6.1 Datasets

We use distant supervision (Mintz et al., 2009) to create datasets. We choose four relations from an entertainment knowledge graph (Fan et al., 2017), and extract texts from Chinese Wikipedia where two entities from the same pair have joint occurrence. We manually check the correctness of relation labeling and remove annotation errors. In total, we have 3480 sentences with annotated entity pairs and relations, and spilt them into training data, testing data and validation data, with the proportion of 70%, 20% and 10%. The statistics of labeled data are summarized in Table 1. Similarly, two entities which do not appear in pairs of pre-defined relations are used for harvesting unlabeled data, which contains 3161 sentences.

6.2 Evaluation of Relation Classification

We implement base neural network with Keras[3] and use dependency parsing results generated by pyltp[4]. The word embeddings are initialized as 50 dimensions, trained on Chinese Wikipedia dump[5] via the Skip-gram model (Mikolov et al., 2013).

[3]https://github.com/fchollet/keras/tree/master/keras
[4]http://www.ltp-cloud.com/
[5]https://dumps.wikimedia.org/zhwiki/20170222

Classifier	Feature set	F1 (%)
logistic regression/ SVM	entity pairs (add)	77.3/ 77.4
	entity pairs (sub)	75.9/ 80.8
	entity pairs (concat)	89.0/ 87.5
	syntactic units, entity pairs (concat)	84.9/ 82.5
	context words, entity pairs (concat)	87.6/ 86.6
	syntactic units, context words	89.2/ 87.8
	syntactic units, context words, entity pairs (concat)	89.9/ 88.0
Shwartz et al. (Shwartz et al., 2016)	shortest dependency path, entity pairs	65.3
Zeng et al. (Zeng et al., 2014)	context words, entity pairs	81.5
RNN+E	syntactic units, entity pairs (concat)	66.8
CNN+E	context words, entity pairs (concat)	91.4
Full implementation	syntactic units, context words, entity pairs (concat)	**92.2**

Table 2: Classifiers with their feature sets and F1 score in relation classification.

Relation name	# Instances	Relation name	# Instances
团队成员 (group members)	1328	所属国家 (belong to the country)	956
家庭成员 (family members)	355	系列作品 (series works)	247
签约公司 (employed by)	144	制作公司 (produced by)	18

Table 3: Semantic relations discovered via ssCRP.

We first study the effects of hyperparameters, i.e., the input dimension of syntactic unit d_{syn}, the output dimension of LSTM units d_{lstm} and the number of convolutional hidden units h. We tune these hyperparameters on the validation set and illustrate the F1 scores with different settings in Fig. 6. As we can see, the best performance is achieved when $d_{syn} = 5$ and $d_{lstm} = 30$. The base neural network shows signs of overfitting with h larger than 150. We heuristically set the window size $k = 5$ and train the network with 0.5 weighted L_2 regularization.

The second part is to evaluate the effectiveness of the proposed features. We implement several state-of-the-art methods of representing the embeddings of entity pairs: concatenation $\vec{v}_{e_1} \oplus \vec{v}_{e_2}$, difference $\vec{v}_{e_1} - \vec{v}_{e_2}$ and sum $\vec{v}_{e_1} + \vec{v}_{e_2}$ model (Baroni et al., 2012; Roller et al., 2014; Mirza and Tonelli, 2016). We train logistic regression and SVM with the combination of the above features. As Table 2 shows, both classifiers achieve the highest F1 scores when trained with all three features. Similar to observations (Mirza and Tonelli, 2016), the concatenated embeddings of entity pairs are the most effective.

The third part is the comparison of our model and other approaches. We implement the CNN-based model (Zeng et al., 2014) and keep the way they use features. Another competitor is the RNN-based model for hypernymy detection (Shwartz et al., 2016), which is modified slightly to classify more generalized semantic relations. We also evaluate the variations of base neural network during iterations, e.g. removing LSTM units (CNN+E) or the convolution layer (RNN+E). The results shown in Table 2 prove that our model has the best performance. The CNN-based models such as CNN+E and model (Zeng et al., 2014) are both significantly effective than RNN-based models, e.g. RNN+E and model (Shwartz et al., 2016). It suggests that lexical features are more effective than syntactic features. The embedding layer improves the performance in either situation due to the use of semantic features.

6.3 Evaluation of Relation Discovery

We first introduce the semantic relations found during the relation discovery process. Table 3 summarizes six relations discovered via ssCRP, containing 3048 instances. The sizes of relations are unbalanced due to the random selection of entity pairs in order to construct unlabeled data automatically.

As studied in previous research (Qiu and Zhang, 2014), OpenRE systems have low performance for Chinese due to flexible language expressions and low data redundancy. Hence, OpenRE methods are not treated as strong baselines for ERC task. To measure the effectiveness of ssCRP, we propose two baseline models following Balvi et al. (Dalvi et al., 2013) for multi-class semi-supervised learning task. Seeded KMeans proposed by Basu et al. (2002) is a clustering method using labeled data to guide the clustering process. We implement an exploratory version where a new centroid is initialized as the most centered data in each iteration. Another intuitive approach is semi-supervised EM-based Naive Bayes with empty

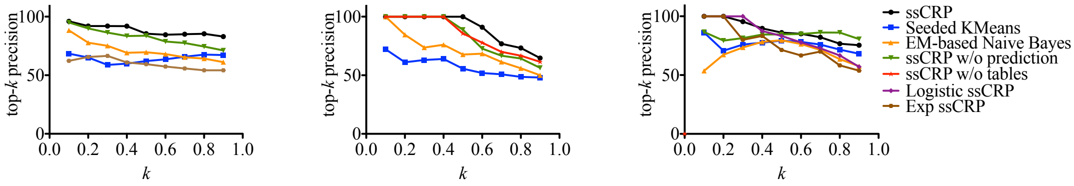

Figure 7: The top-k precisions (%) of three relations (series works, produced by, employed by) generated by various methods with different k.

Algorithm	# Instances	Precision (%)	Recall (%)	F1 (%)
Fit ssCRP	3161	31.0	35.7	33.2
Exploratory EM-based Naive Bayes	3161	70.7	40.2	52.8
Exploratory seeded KMeans	3161	80.5	53.0	63.9
ssCRP w/o tables	593	66.6	60.4	63.3
ssCRP w/o prediction	903	83.7	61.0	70.6
Exp ssCRP	3161	77.9	66.7	71.9
Logistic ssCRP	3161	81.4	66.9	73.0
Full implementation of ssCRP	3048	**83.1**	**68.4**	**75.0**

Table 4: Performance of different algorithms for relation discovery.

classes. The E-step starts with a random initialization of class assignment, and M-step retrains the model until convergence. We also try several variations of our model. For example, we replace the magnifying function with logistic decay (Logistic ssCRP) or exponential decay (Exp ssCRP) proposed in (Blei and Frazier, 2010). In relation prediction process, we make a comparison between the "Max-SecondMax" criterion and goodness-of-fit models such as Kolmogorov-Smirnov test with significance level of 0.05 (Fit ssCRP). Populating new clusters is essential for our model, so we conduct experiments of two related strategies, e.g. not allowing to join existing tables (ssCRP w/o tables) or removing the relation prediction process (ssCRP w/o prediction). We set $\tau = 2$ for all variations, fine tuned over the validation set.

The first experiment presents the top-k precision of newly found relations. Relation instances are sorted according to the cosine similarity between its embedding and averaged relation embedding. We ask human annotator to label whether the extracted top-k relations are correct or not, and evaluate the top-k precision with k ranging from 0.1 to 0.9, and show the results in Fig. 7. The illustrated relations achieve the best performance when they are generated by ssCRP, compared with other baseline models[6]. We heuristically choose $k = 0.4$ because the precision drops relatively faster when k is larger.

Next, we design a pairwise experiment to evaluate this non-standard clustering task. We manually construct a standard testing dataset by sampling pairs of instances from unlabeled data. For two entity pairs x_i and x_j with their respective sentences, we use the domain knowledge graph (Fan et al., 2017) as the ground truth to determine whether x_i and x_j have the same relation. We use Precision, Recall and F1 score as the evaluation metrics. We present the performance of ssCRP and other baseline models in Table 4. For two baseline models, exploratory seeded KMeans performs better than exploratory EM-based Naive Bayes. Experiments of ssCRP variations prove the effectiveness of our specially designed magnifying function and "Max-SecondMax" criterion. The reason that goodness-of-fit models fail is that they are sensitive to check whether the data follows uniform distribution or not, while our purpose is to select those with prominent peaks. Thus non-confident relation labels are assigned to unlabeled entity pairs roughly, increasing the number of false positive instances. For the strategies of table assignment and relation prediction process, the experimental results show that they not only populate new relations, but also improve the overall performance.

6.3.1 Error Analysis

To further obtain additional insights into our method, we study the errors in newly found relations and present three types of errors. A frequent type of confusion happens when entity pairs are closely related

[6]Some baselines do not generate certain relations, therefore these competitors are not included in corresponding figures.

under the same topic but not by the same relation, which accounts for 63.4% errors. For example, two entities involved with cooperation relation are be misclassified to the relation "团队成员 (group members)". Another 23.7% bad cases are due to the false positive results of NER where common nouns are mistaken as named entities, especially for specific names such as "黎明 (Ming Li, which also has the meaning of dawn)". The rest of errors result from the mixture of different grains of relation types. We observe that the relation "所属国家 (belong to the country)" contains a few instances where e_2 is not a country but a finer-grained city or a district.

7 Conclusion

In this paper, we propose the task of ERC to address the problem of domain-specific knowledge acquisition. We propose a DSNN model to address the task, consisting of three modules, an integrated base neural network for relation classification, a similarity-based clustering algorithm ssCRP to generate new relations and constrained relation prediction process with the purpose of populating new relations. Extensive experiments are conducted to evaluate the effectiveness of our approach.

Acknowledgements

This work is partially supported by the National Key Research and Development Program of China under Grant No. 2016YFB1000904.

References

David J Aldous. 1985. *Exchangeability and related topics*. Springer Berlin Heidelberg.

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

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *EACL*, pages 23–32.

Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. 2002. Semi-supervised clustering by seeding. In *ICML*, pages 27–34.

David M. Blei and Peter I. Frazier. 2010. Distance dependent chinese restaurant processes. In *ICML*, pages 87–94.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *HLT/EMNLP*, pages 724–731.

Rui Cai, Xiaodong Zhang, and Houfeng Wang. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *ACL*, pages 756–765.

Bhavana Bharat Dalvi, William W. Cohen, and Jamie Callan. 2013. Exploratory learning. In *ECML/PKDD*, pages 128–143.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011. Open information extraction: The second generation. In *IJCAI*, pages 3–10.

Yan Fan, Chengyu Wang, Guomin Zhou, and Xiaofeng He. 2017. Dkgbuilder: An architecture for building a domain knowledge graph from scratch. In *DASFAA*, pages 663–667.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING*, pages 539–545.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *ACL*, page 22.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL*, pages 1003–1011.

Paramita Mirza and Sara Tonelli. 2016. On the contribution of word embeddings to temporal relation classification. In *COLING*, pages 2818–2828.

Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom M. Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Likun Qiu and Yue Zhang. 2014. ZORE: A syntax-based system for chinese open relation extraction. In *EMNLP*, pages 1870–1880.

Desh Raj, Sunil Kumar Sahu, and Ashish Anand. 2017. Learning local and global contexts using a convolutional recurrent network model for relation classification in biomedical text. In *CoNLL 2017*, pages 311–321.

Carl Edward Rasmussen. 1999. The infinite gaussian mixture model. In *NIPS*, pages 554–560.

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

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *COLING*, pages 1025–1036.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *ACL*, pages 2389–2398.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. Combining recurrent and convolutional neural networks for relation classification. In *HLT-NAACL*, pages 534–539.

Chengyu Wang and Xiaofeng He. 2016. Chinese hypernym-hyponym extraction from user generated categories. In *COLING*, pages 1350–1361.

Chengyu Wang, Junchi Yan, Aoying Zhou, and Xiaofeng He. 2017. Transductive non-linear learning for chinese hypernym prediction. In *ACL*, pages 1394–1404.

Fei Wu and Daniel S. Weld. 2007. Autonomously semantifying wikipedia. In *CIKM*, pages 41–50.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *EMNLP*, pages 1785–1794.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344.

Qing Zhang and Houfeng Wang. 2017. Noise-clustered distant supervision for relation extraction: A nonparametric bayesian perspective. In *EMNLP*, pages 1809–1814.