Condition	Value of $w_{i,j}$
$(x_i, y_i) \in D, (x_j, y_j) \in D, r_i = r_j$	1
$(x_i, y_i) \in D, (x_j, y_j) \in D, r_i \neq r_j$	0
$(x_i, y_i) \in D, (x_j, y_j) \in U, r_i = r_m$	$\frac{1}{2}p_{j,m}(\cos(M_m\vec{x}_i - \vec{x}_i, M_m\vec{x}_j - \vec{x}_j) + 1)$
$(x_i, y_i) \in U, (x_j, y_j) \in D, r_j = r_m$	$\frac{1}{2}p_{i,m}(\cos(M_m\vec{x}_i - \vec{x}_i, M_m\vec{x}_j - \vec{x}_j) + 1)$
$(x_i, y_i) \in U, (x_j, y_j) \in U$	$\frac{1}{2}\sum_{r_m \in R} p_{i,m}p_{j,m} \cdot (\cos(M_m\vec{x}_i - \vec{x}_i, M_m\vec{x}_j - \vec{x}_j) + 1)$

Table 1: The choice of $w_{i,j}$ according to different conditions.

place J_g based on the negative log likelihood:

$$J_g^{'} = - \sum_{(x_i,y_i) \in D \cup U} \sum_{(x_j,y_j) \in Nb(x_i,y_i)} \log \Pr((x_j, y_j)|\vec{r}_i) \tag{2}$$

A remaining problem is to define the neighborhood $Nb(x_i, y_i)$ properly, to preserve the hyperspherical similarity property of the distance function $g(f_i(\vec{x}_i) - \vec{x}_i, f_j(\vec{x}_j) - \vec{x}_j)$. In this work, we introduce a weight factor $w_{i,j} \in [0, 1]$ w.r.t. two pairs (x_i, y_i) and (x_j, y_j) in $D \cup U$ that quantifies the similarity between the two pairs in the SphereRE space. If $(x_i, y_i) \in D$ and $(x_j, y_j) \in D$, because the true lexical relation types are known, we simply have: $w_{i,j} = I(r_i = r_j)$.

We continue to discuss other conditions. If i) $(x_i, y_i) \in D$ has the lexical relation type r_m, and ii) the lexical relation type of $(x_j, y_j) \in U$ is unknown but is predicted to be r_m with probability $p_{j,m}$, the similarity between (x_i, y_i) and (x_j, y_j) in terms of angles is defined using the weighted cosine similarity function in the range of $(0, 1)$:

$$w_{i,j} = \frac{1}{2}p_{j,m}(\cos(M_m\vec{x}_i - \vec{x}_i, M_m\vec{x}_j - \vec{x}_j) + 1)$$

A similar case holds for $(x_i, y_i) \in U$ and $(x_j, y_j) \in D$. If $(x_i, y_i) \in U$ and $(x_j, y_j) \in U$, because the lexical relation types of both pairs are unknown, we compute the weight $w_{i,j}$ by summing up all the weighted cosine similarities over all possible lexical relation types in R:

$$\begin{aligned} w_{i,j} = &\frac{1}{2} \sum_{r_m \in R} p_{i,m}p_{j,m} \cdot \\ &(\cos(M_m\vec{x}_i - \vec{x}_i, M_m\vec{x}_j - \vec{x}_j) + 1) \end{aligned}$$

Readers can also refer to Table 1 for a summarization of the choices of $w_{i,j}$.

To reduce computational complexity, we propose a Monte-Carlo based sampling and learning method to learn SphereRE vectors based on the values of $w_{i,j}$. The algorithm is illustrated in Algorithm 1. It starts with the random initialization of SphereRE vector $\vec{r}_i$ for each $(x_i, y_i) \in D \cup U$. An iterative process randomly selects one pair (x_i, y_i) as the starting point. The next pair (x_j, y_j) is selected with probability as follows:

$$\Pr((x_j, y_j)|(x_i, y_i)) = \frac{w_{i,j}}{\sum_{(x_j^{'},y_j^{'}) \in D_{mini}} w_{i,j^{'}}} \tag{3}$$

where D_{mini} is a mini-batch of term pairs randomly selected from $D \cup U$. In this way, the algorithm only needs to traverse $|D_{mini}|$ pairs instead of $|D| + |U|$ pairs. This process continues, resulting in a sequence of pairs, denoted as $\mathcal{S}$: $\mathcal{S} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_{|\mathcal{S}|}, y_{|\mathcal{S}|})\}$. Denote l as the window size. We approximate $J_g^{'}$ in Eq. (2) by $-\sum_{(x_i,y_i) \in \mathcal{S}} \sum_{j=i-l(j \neq i)}^{i+l} \log \Pr((x_j, y_j)|\vec{r}_i)$ using the negative sampling training technique of the Skip-gram model (Mikolov et al., 2013a,b).

The values of SphereRE vectors $\vec{r}_i$ are continuously updated until all the iterations stop. We can see that $\vec{r}_i$s are the low-dimensional representations of lexical relation triples, encoded in the hyperspherical space. The process is shown in Algorithm 1.

Algorithm 1 SphereRE Learning

```
1: for each (x_i, y_i) ∈ D ∪ U do
2:     Randomly initialize SphereRE vector r_i;
3: end for
4: for i = 1 to max iteration do
5:     Sample a sequence based on Eq. (3):
       S = {(x_1, y_1), (x_2, y_2), ···, (x_|S|, y_|S|)};
6:     Update all SphereRE vectors r_i by minimizing
       −Σ_{(x_i,y_i)∈S} Σ_{j=i−l(j≠i)}^{i+l} log Pr((x_j, y_j)|r_i);
7: end for
```

In practice, we find that there is a drawback of the sampling process. Because the predictions for all $(x_i, y_i) \in U$ are probabilistic, it leads to the situation where the algorithm prefers to choose term pairs in D to form the sequence $\mathcal{S}$. The low sampling rate of U results in the poor representation learning quality of these pairs. Here, we employ a boosting approach to increase chances