



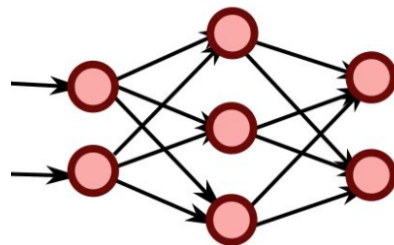
# Neural Snowball for Few-Shot Relation Learning

---

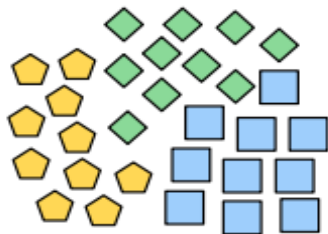
报告人：王翔

# 场景

|                                              |                                  |
|----------------------------------------------|----------------------------------|
| S1: Chicago is located in the United States. | Chicago United States<br>country |
| {<Chicago, country, United States>}          |                                  |



Existing Relations



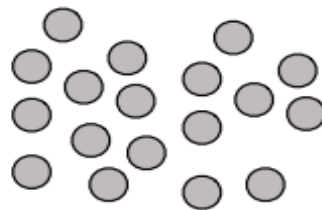
Large-Scale Data

The New Relation



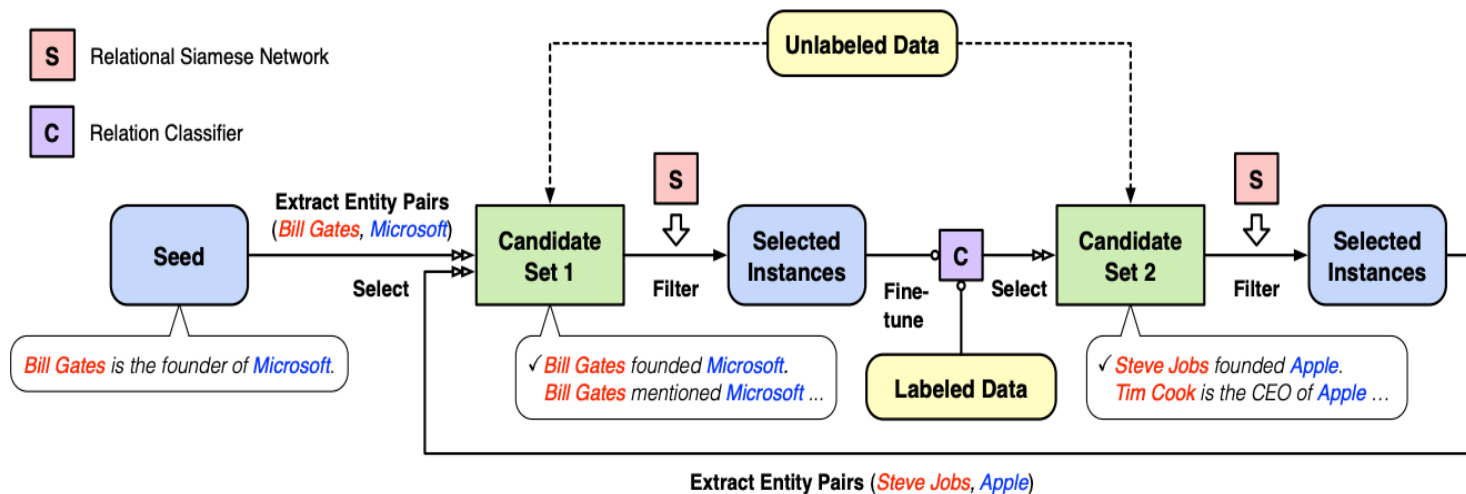
Few-Shot Instances

Unlabeled Corpora



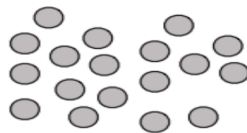
Large-Scale Data

# Neural Snowball Process



- 问题：扩充少样本关系的样本数量

Unlabeled Corpora

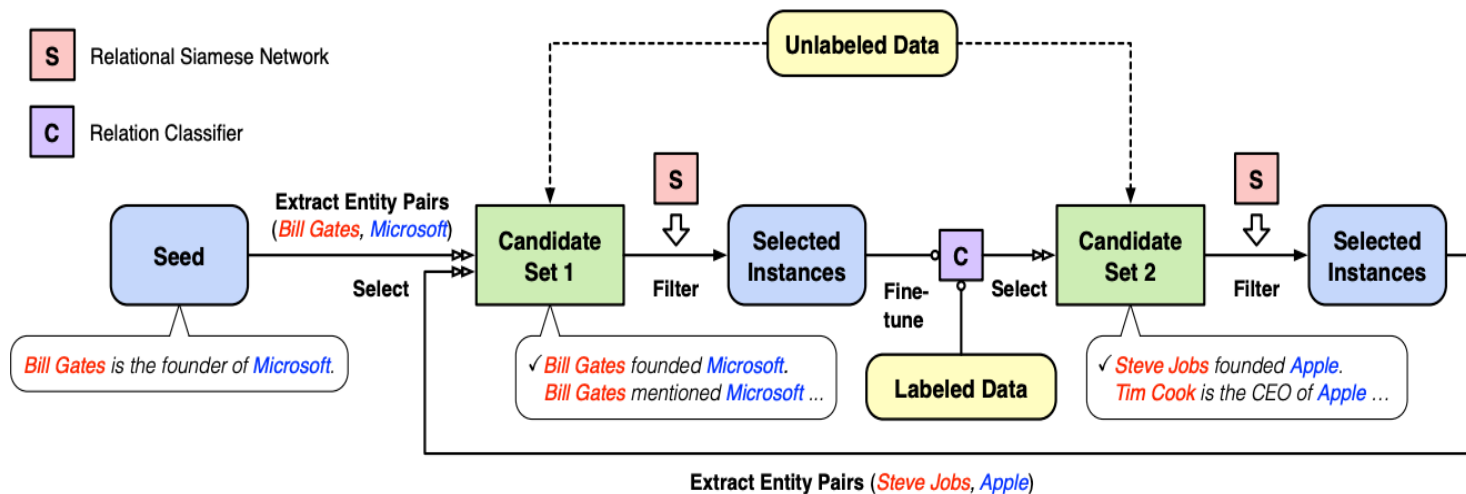


Large-Scale Data



Labeled

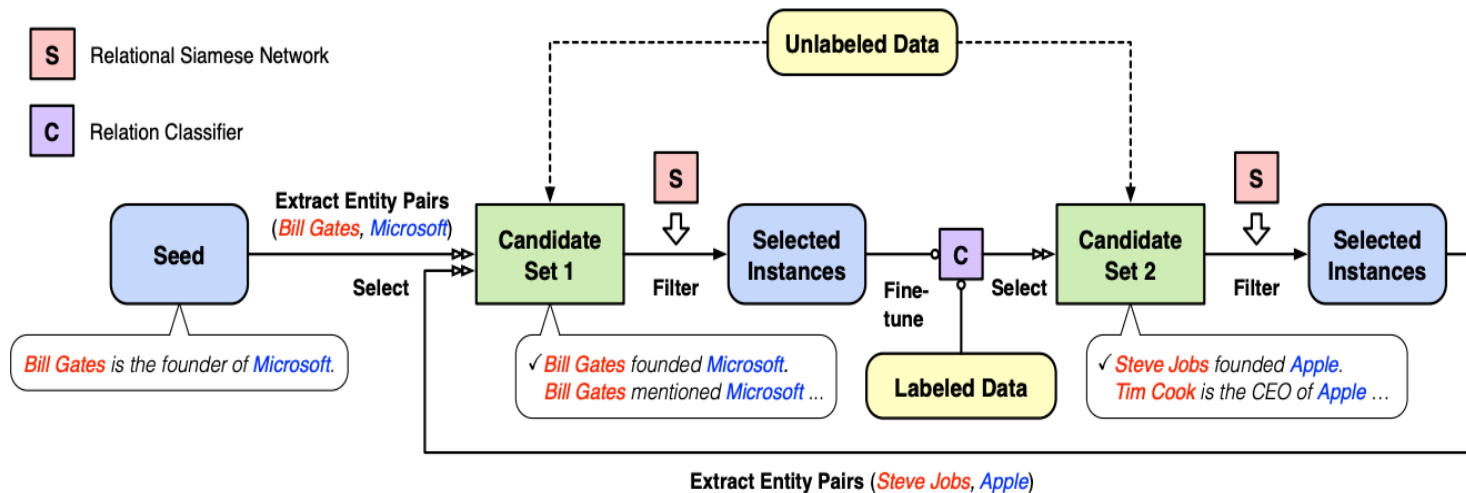
# Neural Snowball Process

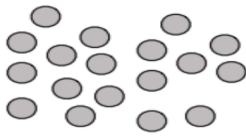


- 获取与Seed同实体的置信样本

- 获取与Seed同关系的置信样本

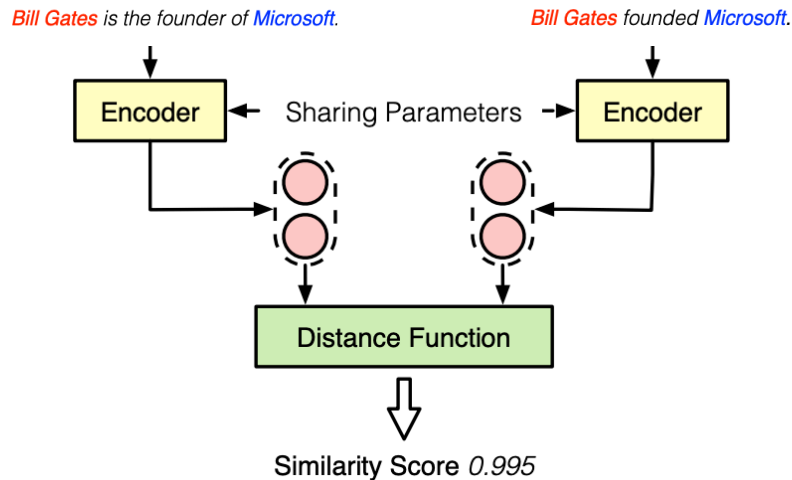
# 获取与Seed同实体的置信样本



*Bill Gates is the founder of Microsoft.* → (*Bill Gates*, *Microsoft*) →  → *Bill Gates founded Microsoft. Bill Gates mentioned Microsoft.*

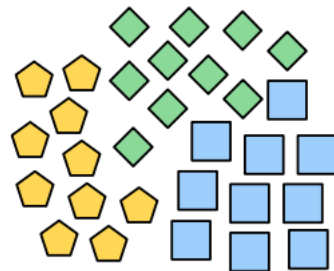
Large-Scale Data

# 获取与Seed同实体的置信样本



✓ *Bill Gates* founded *Microsoft*.  
*Bill Gates* mentioned *Microsoft*

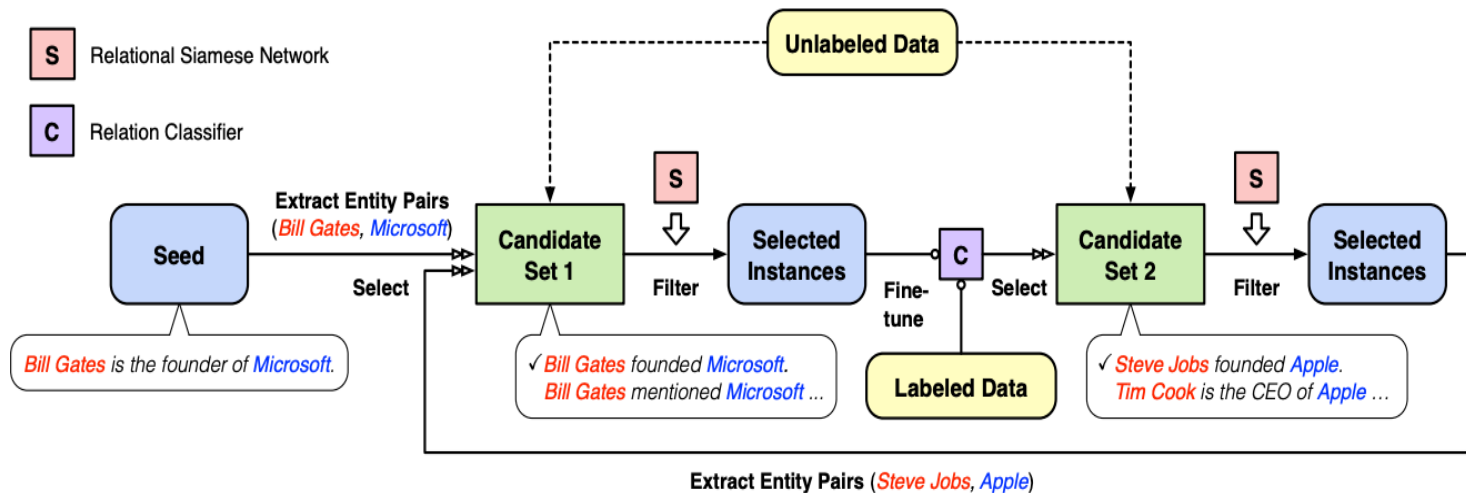
## Existing Relations



*Large-Scale Data*

$$s(x, y) = \sigma(\mathbf{w}_s^T (f_s(x) - f_s(y))^2 + b_s)$$

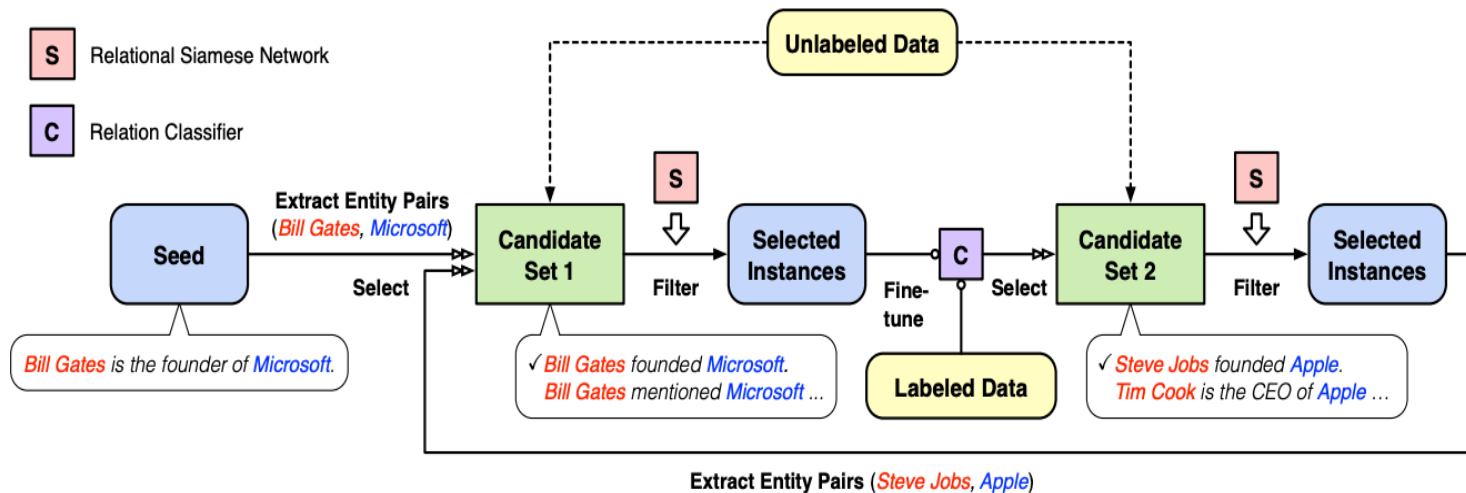
# 获取与Seed同实体的置信样本



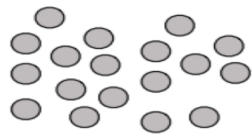
$$g(x) = \sigma(\mathbf{w}^T f(x) + b)$$

$$\begin{aligned} \mathcal{L}_{S_b, \mathcal{T}_b}(g_{\mathbf{w}, b}) &= \sum_{x \in S_b} \log g_{\mathbf{w}, b}(x) \\ &\quad + \mu \sum_{x \in \mathcal{T}_b} \log(1 - g_{\mathbf{w}, b}(x)) \end{aligned}$$

# 获取与Seed同关系的置信样本



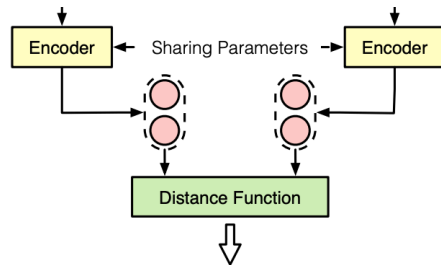
Unlabeled Corpora



Large-Scale Data



*Steve Jobs* founded *Apple*.





# 效果

| Model                             | 5 Seed Instances |              |              | 10 Seed Instances |              |              | 15 Seed Instances |              |              |
|-----------------------------------|------------------|--------------|--------------|-------------------|--------------|--------------|-------------------|--------------|--------------|
|                                   | P                | R            | F1           | P                 | R            | F1           | P                 | R            | F1           |
| BREDS                             | 33.71            | 11.89        | 17.58        | 28.29             | 17.02        | 21.25        | 25.24             | 17.96        | 20.99        |
| Fine-tuning (CNN)                 | 46.90            | 9.08         | 15.22        | 47.58             | 38.36        | 42.48        | 74.70             | 48.03        | 58.46        |
| Relational Siamese Network (CNN)  | 45.00            | 31.37        | 36.96        | 46.42             | 30.68        | 36.94        | 49.32             | 30.46        | 37.66        |
| Distant Supervision (CNN)         | 44.99            | 31.06        | 36.75        | 42.48             | 48.64        | 45.35        | 43.70             | 54.76        | 48.60        |
| <b>Neural Snowball (CNN)</b>      | 48.07            | 36.21        | 41.30        | 47.28             | 51.49        | 49.30        | 68.25             | 58.90        | 63.23        |
| Fine-tuning (BERT)                | 50.85            | 16.66        | 25.10        | 59.87             | 55.19        | 57.43        | <b>81.60</b>      | 58.92        | 68.43        |
| Relational Siamese Network (BERT) | 39.07            | <b>51.39</b> | 44.47        | 42.42             | 54.93        | 47.87        | 44.10             | 52.73        | 48.03        |
| Distant Supervision (BERT)        | 38.06            | 51.18        | 43.66        | 38.45             | <b>76.12</b> | 51.09        | 35.48             | <b>80.33</b> | 49.22        |
| <b>Neural Snowball (BERT)</b>     | <b>56.87</b>     | 40.43        | <b>47.26</b> | <b>60.50</b>      | 62.20        | <b>61.34</b> | 78.13             | 66.87        | <b>72.06</b> |

---

## Algorithm 1: Fine-tuning the Classifier

---

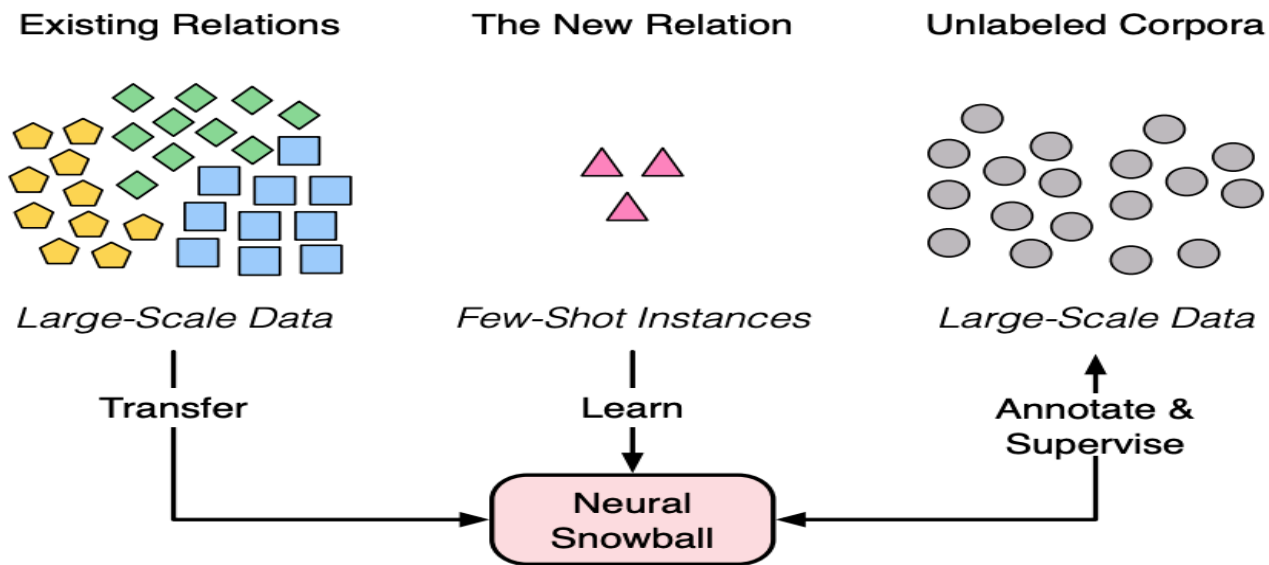
**Input:** New instance set  $\mathcal{S}_r$ , historical relation dataset  $\mathcal{S}_N$   
**Result:** Optimized  $\mathbf{w}$  and  $b$

```

1 Randomly initialize  $\mathbf{w}$  and  $b$ 
2 for  $i \leftarrow 1$  to  $e$  do
3   // Get a sequence of minibatches from  $\mathcal{S}_r$ 
4    $\mathcal{S}_{batch.seq} \leftarrow \text{batch\_seq}(\mathcal{S}_r, bs)$ 
5   for  $\mathcal{S}_b \in \mathcal{S}_{batch.seq}$  do
6     // Sample the negative batch
7      $\mathcal{T}_b \leftarrow \text{sample}(\mathcal{S}_N, bs)$ 
8     Update  $\mathbf{w}$  and  $b$  w.r.t.  $\mathcal{L}_{\mathcal{S}_b, \mathcal{T}_b}(g_{\mathbf{w}, b})$ 
9     with learning rate  $\lambda$ 
10  end
11 end

```

---



- 召回

- 打分 (排序 | 匹配)



# 感谢您的聆听！

---

报告人：王翔