# Chương 1

# Cơ sở lý thuyết

## 1.1 Các khái niệm cơ bản

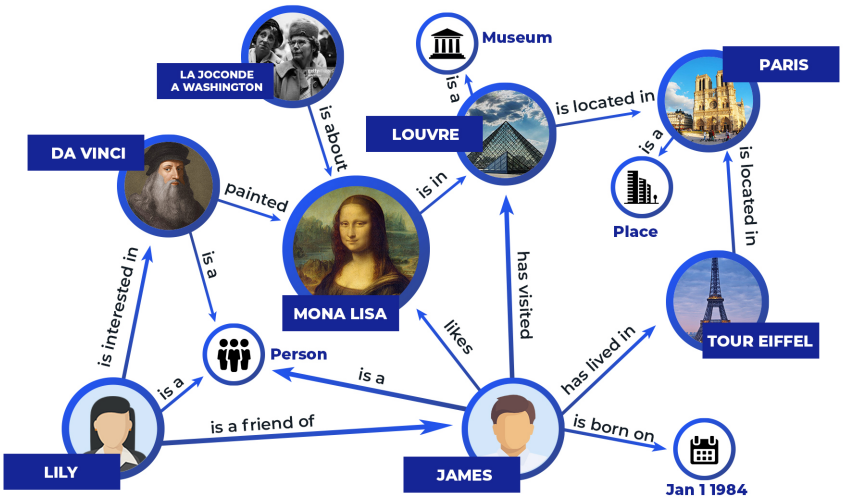
Đồ thị tri thức lưu trữ các thông tin có cấu trúc trong thế giới thực, được sử dụng rộng rãi trong lĩnh vực trí tuệ nhân tạo như truy xuất thông tin, xử lý ngôn ngữ tự nhiên, hệ thống khuyến nghị,...

**Định nghĩa 1.1.1.** *(Đồ thị vô hướng)*
*Một đồ thị vô hướng $G$ là một cặp có thứ tự $G = (V, E)$, với $V$ là một tập, còn $E$ là tập với các phần tử là các đa tập lực lượng hai trên $V$. Các phần tử của $V$ được gọi là đỉnh, còn các phần tử của $E$ được gọi là các cạnh của đồ thị vô hướng $G$. Nếu $e = \{a, b\}$ là một cạnh của $G$ thì $a, b$ được gọi là các đỉnh liên thuộc với $e$. Cạnh có dạng $\{a, a\}$ với $a \in V$ được gọi là khuyên.*

**Định nghĩa 1.1.2.** *(Đồ thị có hướng)*
*Một đồ thị có hướng $G$ là một cặp có thứ tự $G = (V, E)$, với $V$ là một tập, còn $E$ là một tập con của tích Dề các $V \times V$. Các phần tử của $V$ được gọi là các đỉnh của đồ thị có hướng $G$, còn các phần tử của $E$ được gọi là các cung của đồ thị có hướng $G$. Cụ thể, nếu $(a, b) \in E$ thì $(a, b)$ được gọi là cung của $G$ với đỉnh đầu là $a$, đỉnh cuối là $b$ và có hướng đi từ $a$ đến $b$.*

**Định nghĩa 1.1.3.** *(Đa đồ thị có hướng)*
*Một đa đồ thị có hướng $G = (V, E)$, nếu $G$ là đồ thị có hướng và có thể tồn tại nhiều hơn một cung kết nối hai đỉnh $a, b \in V$.*

**Định nghĩa 1.1.4.** *(Đồ thị tri thức)*
*Đồ thị tri thức là một đa đồ thị (có hướng) $G = (E, R)$ bao gồm tập đỉnh $E$ và tập cạnh $R$, trong đó tập đỉnh tương ứng với tập các thực thể và tập cạnh tương ứng với tập các quan hệ. Mỗi dữ kiện trong đồ thị thường được biểu diễn dưới dạng bộ 3 $(h, r, t)$, trong đó $h$ được gọi là thực thể đầu, $t$ được gọi là thực thể cuối và $r$ là quan hệ giữa $h$ và $t$.*




Hình 1.1: Đồ thị tri thức [1].

Hình 1.1 mô tả đồ thị tri thức với số lượng thực thể là 12 và số lượng quan hệ là 14, nhưng trong thực thể theo thống kê đồ thị tri thức của google xây dựng lên tới 5 tỷ thực thể và 500 tỷ sự kiện. Số lượng các thực thể và các quan hệ ngày càng tăng cùng với việc khai thác thông tin trong đồ thị tri thức ngày càng trở nên trọng, bài toán hoàn thiện đồ thị tri thức là điều cần thiết.

[1]Nguồn ảnh: `https://www.researchgate.net`

**Định nghĩa 1.1.5.** *(Bài toán dự đoán liên kết)*
*Dự đoán liên kết là một nhiệm vụ cơ bản trong việc hoàn thành biểu đồ tri thức sử dụng các mối quan hệ hiện có để suy ra các quan hệ mới để từ đó xây dựng biểu đồ tri thức đầy đủ hơn. Về cơ bản, nó bao gồm hai nhiệm vụ:*

- *Nhiệm vụ thứ nhất: Dự đoán thực thể, dự đoán $h$ khi cho trước $(?, r, t)$ hoặc dự đoán $t$ khi cho trước $(h, r, ?)$*
- *Nhiệm vụ thứ hai: Dự đoán quan hệ, dự toán $r$ khi cho trước $(r, ?, t)$.*

Để giải quyết bài toán dự đoán liên kết, có nhiều kỹ thuật khác nhau được đề xuất bao gồm phương pháp dựa trên phân tách, phương pháp dựa trên đường dẫn và phương pháp dựa trên nhúng. Bài báo cáo này tập trung vào các mô hình nhúng để giải quyết bài toán dự đoán liên kết.

**Định nghĩa 1.1.6.** *(Nhúng đồ thị)*
*Nhúng đồ thị là quá trình biến đổi các đặc trưng của đồ thị sang một không gian khác có số chiều thấp. Về cơ bản, nhúng đồ thị là phương thức ánh xạ nội dung của các thực thể và các quan hệ là các vec-tơ chiều cao trong đồ thị tri thức thành các vec-tơ chiều thấp, sao cho các thuộc tính của đồ thị được lữu trữ càng nhiều càng tốt.*

## 1.2 Mô hình DistMult

Bài báo Yang et al., 29 September 2017. "Embedding Entities and Relations for Learning and Inference in Knowledge Bases" mô tả một cách học biểu diễn thực thể và quan hệ (gọi tắt mô hình DistMult) được tóm lược như sau:

- Học biểu diễn thực thể: Mỗi đầu vào của thực thể và quan hệ được biểu diễn bởi một vector có số chiều lớn hoặc là vector "one-hot" hoặc là vector "n-hot". Kí hiệu $x_{e_1}$ và $x_{e_2}$ lần lượt là các vector đầu vào của thực thể $e_1$ và thực thể $e_2$, $x_r$ là vector đầu vào của quan hệ $r$. Kí hiệu $W_E$ là ma trận nhúng thực thể, $W_R$ là ma trận nhúng quan hệ. Các biểu diễn thực thể đã được học, $y_{e_1}, y_{e_2}, y_r$ có thể viết như sau:

  $$y_{e_1} = f(W_E x_{e_1}), y_{e_2} = f(W_E x_{e_2}), y_r = W_R x_r$$

  trong đó $f$ là hàm tuyến tính hoặc phi tuyến và $W$ là ma trận tham số. Trong trường hợp đơn giản, các vector biểu diễn có thể được tính toán như sau: $y_{e_1} = W_E x_{e_1}, y_{e_2} = W_E x_{e_2}, y_r = W_R x_r$ và $y_{e_1}, y_{e_2}, y_r \in \mathbb{R}^N$, với $N$ là số chiều của vector nhúng.

- Học biểu diễn quan hệ: Mô hình DistMult lựa chọn hàm tính điểm là hàm tuyến tính. Hàm tính điểm với mục tiêu cho các bộ 3 hợp lệ nhận điểm số cao và các bộ 3 không hợp lệ nhận điểm số thấp.

  $$S(e_1, r, e_2) = y_{e_1}^T diag(y_r) y_{e_2} = \sum_{i=1}^{N} y_{e_1}[i] y_r[i] y_{e_2}[i]$$

  trong đó $diag(y_r) \in \mathbb{R}^{N \times N}$ là ma trận đường chéo.

- Hàm mất mát của mô hình: Các tham số của mô hình $W_E, W_R$ được học bằng cách tối ưu hàm mất mát. Cho bộ 3 dương $T$, chúng ta xây dựng các bộ 3 âm $T'$ bằng cách thay thế các thực thể sao cho $T' = \{(e_1', r, e_2) | e_1' \in E, (e_1', r, e_2) \notin T\} \cup \{(e_1, r, e_2') | e_1' \in E, (e_1, r, e_2') \notin T\}$. Kí hiệu hàm tính điểm của bộ 3 $(e_1, r, e_2)$ là $S_(e_1, r, e_2)$. Mục tiêu huấn luyện là giảm thiểu hàm margin-based ranking:

  $$L(\Omega) = \sum_{(e_1,r,e_2) \in T} \sum_{(e_1',r,e_2') \in T'} \max\{S_(e_1', r, e_2') - S_(e_1, r, e_2) + 1, 0\}$$

Trong bài báo Kadlec et el., 30 May 2017. "Knowledge Base Completion: Baselines Strike Back", mô hình DistMult được đánh giá là mô hình đơn giản, có hiệu suất cao trên các bộ dữ liệu tiêu chuẩn FB15k và WN18 so với các mô hình khác vào thời điểm đó.

## 1.3 Mô hình TuckER

Bài báo Ivana Balazevic et al., 24 Aug 2019. "TuckER: Tensor Factorization for Knowledge Graph Completion" đã đưa mô hình nhúng TuckER dựa trên ý tưởng phân tách ma trận và mô hình này được chứng minh là sự tổng quát của mô hình DistMult. Ý tưởng của mô hình: Tucker decomposition (Tucker, 1964) phân rã một tensor thành một tập các ma trận và một tensor lõi.

$$X \approx Z \times A \times B \times C,$$

trong đó $X \in \mathbb{R}^{I \times J \times K}, Z \in \mathbb{R}^{P \times Q \times R}, A \in \mathbb{R}^{I \times P}, B \in \mathbb{R}^{J \times Q}, C \in \mathbb{R}^{K \times R}$, thông thường $P, Q, R$ nhỏ hơn $I, J, K$ nên có thể coi $Z$ là phiên bản nén của $X$. Từ ý tưởng phân tách ma trận, mô hình TuckER mô tả cách học biểu diễn thực thể và quan hệ tóm lược như sau:

- Xét $\mathbf{E}$ là tập hợp các thực thể và $\mathbf{R}$ là tập các quan hệ của đồ thị tri thức. Trong đó một bộ ba được kí hiệu là $(e_1, r, e_2)$ với $e_1, e_2 \in \mathbf{E}$ và $r \in \mathbf{R}$. Ma trận nhúng thực thể $W_E = A = C \in \mathbb{R}^{n_e \times d_e}$ và ma trận nhúng quan hệ $W_R = B \in \mathbb{R}^{n_r \times d_r}$ với $n_e$ và $n_r$ lần lượt là số thực thể và số quan hệ, $d_e$ và $d_r$ lần lượt là độ dài vector biểu diễn thực thể và quan hệ. Các vector nhúng được tính toán như sau: $y_{e_1} = W_E x_{e_1}, y_{e_2} = W_E x_{e_2}, y_r = W_R x_r$ và $y_{e_1}, y_{e_2} \in \mathbb{R}^{d_e}, y_r \in \mathbb{R}^{d_r}$. Định nghĩa hàm tính điểm TuckER như sau:

$$S(e_1, r, e_2) = W \times e_1 \times w_r \times e_2,$$

- Hàm sigmoid $p = \sigma(S(.))$ được áp dụng vào hàm tính điểm cho mỗi bộ 3 $(e_1, r, e_2)$ mục tiêu đưa $p \in (0, 1)$ để dự đoán một bộ 3 hợp lệ hay không.

- Hàm mất mát của mô hình: Các tham số của mô hình $W_E, W_R$ và $W$ được học bằng cách tối ưu hàm mất mát. Với mỗi bộ 3 $(e_1, r, e_2)$ trong dữ liệu, chúng ta xây dựng thêm bộ 3 tương ứng đảo ngược $(e_1, r^{-1}, e_2)$. Mục tiêu huấn luyện là cực tiểu hoá hàm mất mát Bernoulli neagtive log-likelihood được định nghĩa như sau:

$$L = -\frac{1}{n_e} \sum_{i=1}^{n_e} (y^{(i)} \log(p^{(i)}) + (1 - y^{(i)})(1 - \log p^{(i)}))$$

  với $p \in \mathbb{R}^{n_e}$ là vector dự đoán xác suất và $y \in \mathbb{R}^{n_e}$ là vector nhị phân.

Chương trình của mô hình TuckER đã được công bố tại: `https://github.com/ibalazevic/TuckER`, chương trình được cài đặt bằng pytorch, được kiểm thử trên Google Colab với 2 bộ dữ liệu tiêu chuẩn FB15k và WN18 có hiệu suất cao so với mô hình Distmult và các mô hình cùng thời điểm.

## 1.4 Các chỉ số đánh giá mô hình

Trong bài toán dự đoán liên kết, các chỉ số sau đây được đưa ra để đánh giá hiệu suất của một mô hình nhúng:

- Xếp hạng đối ứng trung bình (MRR): MRR là thước đo đánh giá hiệu quả của một mô hình được định nghĩa bởi công thức

$$\text{MRR} = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{rank_i}$$

trong đó $Q$ là số lượng truy vấn, $\frac{1}{rank_i}$ là xếp hạng của truy vấn thứ $i$.

**Ví dụ 1.4.1.** *Sau khi mô hình được huấn luyện, với tập kiểm tra bao gồm* 3 *câu hỏi* $(e_1, r, e_2)$*, mỗi câu hỏi đưa ra tập danh sách các câu trả lời* $e_2$*. Dựa vào bảng kết quả ta có đánh giá của mô hình:*

| **Tập câu trả lời** | **Câu trả lời đúng** | **Xếp hạng** |
|---|---|---|
| Hà Nội, Hà Nam, Nam Định | Hà Nội | 1 |
| Xây Dựng, Bách Khoa, Công Nghiệp | Bách Khoa | 2 |
| CNTT, ĐTVT, TT | TT | 3 |

$$MRR = \frac{1}{3}(\frac{1}{1} + \frac{1}{2} + \frac{1}{3}) = 0.611$$

- Chỉ số đánh giá Hits@k được định nghĩa bởi công thức

$$Hits@k = \frac{|q \in Q : rank(q) \leq k|}{|Q|}$$

với $rank(q)$ là xếp hạng câu trả lời cho $q$ và $k \in \{1, 3, 10\}$.
Chỉ số này cho biết tỉ lệ số lượng câu trả lời đúng có xếp hạng trong tập $k$ câu trả lời trên tổng số lượng câu hỏi.

**Ví dụ 1.4.2.** *Trong ví dụ 1.4.1, Hits@1 của mô hình là* 1/3 *bởi chỉ có duy nhất một câu trả lời đầu tiên xếp hạng 1 trong tập các câu trả lời, Hits@3 của mô hình là* 1 *bởi cả* 3 *câu trả lời đều xếp hạng nhỏ hơn* 3 *trong tập các câu trả lời.*

## 1.5 Mô hình CILK

Bài báo Sahisnu Mazumder et al., 21 Dec 2019. "Lifelong and Interactive Learning of Factual Knowledge in Dialogues" mô tả mô hình đối thoại, trong đó hệ thống tương tác người dùng để người dùng cung cấp các dữ kiện hỗ trợ sau đó hệ thống suy luận đưa ra câu trả lời cho truy vấn.

1. Kiến trúc tổng quan

- Đầu vào: Người dùng đưa ra truy vấn có dạng bộ 3 $(h, r, ?)$ hoặc $(?, r, t)$.
- Thuật toán: Bao gồm 2 phần

  Module tương tác: Tương tác với người dùng thu được các sự kiện hỗ trợ SFs.

  Mô hình suy luận: Suy luận câu trả lời cho truy vấn.
- Đầu ra: Đưa ra câu trả lời e cho truy vấn hoặc từ chối truy vấn.

2. Vấn đề tương tác

- Cơ sở tri thức KB $\mathbf{K}$ là tập hợp các bộ 3 $\{(h, r, t)\} \subset E \times R \times E$ trong đó $E$ là tập hợp các thực thể và $R$ là tập hợp các quan hệ.
- Hai loại truy vấn người dùng $q = (e, r, ?)[q = (?, r, e)]$. Nếu $e \in E$ và $r \in R$ thì $q$ là truy vấn thế giới đóng. Nếu $e \notin E$ và $r \notin R$ thì $q$ là truy vấn thế giới mở.
- Chuyển đổi truy vấn thế giới mở sang thế giới đóng: CILK tương tác với người dùng để thu được một tập hợp các sự kiện hỗ trợ SFs. Sự kiện hỗ trợ bao gồm tập hợp bộ manh mối $C_r$ là các bộ 3 liên quan đến $r$ và tập hợp $F_e$ là các bộ 3 liên quan đến $e$.
- Mô hình sử dụng $(K \cup C_r \cup F_e)$ để suy luận $q$.

3. Mô hình suy luận $M$

Đầu vào truy vấn $q = (h, r, ?)$, mô hình suy luận $M$ suy luận $q$ bằng cách tìm kiếm thực thể trả lời $e_i$ từ tập $E$. Sử dụng phương pháp nhúng cơ trở tri thức (KBE) để thiết kế M:

- KBE mã hoá các thông tin quan hệ, nhúng và biểu diễn các vector chiều thấp các thực thể và các quan hệ. Mục tiêu các bộ 3 hợp lệ nhận điểm cao và các bộ 3 không hợp lệ nhận điểm thấp, được xác định bởi hàm $S(.)$.
- Trong mô hình KBE tuyến tính, với bộ 3 $(h, r, t)$ tương ứng là các vector one-hot hoặc n-hot $(x_h, x_r, x_t)$ chiếu lên các vector chiều thấp $(v_h, v_r, v_t)$ thông qua ma trận nhúng thực thể $W_E$ và ma trận nhúng quan hệ $W_R$

$$v_h = W_E x_h, v_r = W_R x_r, v_t = W_E x_t$$

- Bài báo cáo sử dụng mô hình Distmult. Với hàm tính điểm của mô hình được định nghĩa như sau:

$$S(h,r,t) = v_h^T diag(v_r) v_t = \sum_{i=1}^{N} v_h[i] v_r[i] v_t[i]$$

- Các tham số của $M$, $W_E, W_R$ được học bằng các cực tiểu hoá hàm margin-based $L$:

$$L = \sum_{d \in D^+} \sum_{d' \in D^-} \max\{S(d') - S(d) + 1, 0\}$$

với $D^+$ là bộ 3 có trong $K$, positive triples. $D^-$ là tập hợp các negative triples bằng cách thay thế thực thể đầu hoặc thử thế cuối của $(h,r,t)$ bởi ngẫu nhiên $h'$ và $t'$ sao cho $(h',r,t), (h,r,t') \notin K$.

Từ chối suy luận: Với mỗi truy vấn không tồn tại thực thể trả lời trong $K$, CILK từ chối các truy vấn không trả lời được. Để quyết định từ chối truy vấn hay không, CILK duy trì bộ ngưỡng đệm T.

- Bên cạnh việc tạo dữ liệu huấn luyện (training dataset), CILK tạo tập dữ liệu đánh giá (validtion dataset) $D_{vd}$ bao gồm bộ truy vấn có dạng $(q, E^+, E^-)$. Ở đây $q$ là một truy vấn đầu (đuôi) liên quan đến thực thể $e$ và quan hệ $r$.

- $E^+ = \{e_1^+, ..., e_p^+\}$ là tập hợp $p$ positive entities trong $K$ và $E^+ = \{e_1^+, ..., e_p^+$ là tập hợp $n$ negative entites lấy ngẫu nhiên từ $K$ sao cho $E^+ \cap E^- = \emptyset$. Xét $D_{vd}^e = \{(q, E^+, E^-) | (q, E^+, E^-) \in D_{vd}, e \in q\}$ là truy vấn đánh giá gồm thực thể $e$ và $D_{vd}^e = \{(q, E^+, E^-) | (q, E^+, E^-) \in D_{vd}, r \in q\}$ là các truy vấn đánh giá chứa quan hệ $r$.

- Tính toán ngưỡng dự đoán $T[z]$ với $z$ có thể là $e$ hoặc $r$:

$$T[z] = \frac{1}{2|D_{vd}^z|} \sum_{(q,E^+,E^-) \in D_{vd}^z} \mu_E^+ + \mu_E^-$$

trong đó $\mu_E^+ = \frac{1}{|E^+|} \sum_{e_i^+ \in E^+} S(q, e_i^+)$ và $\mu_E^- = \frac{1}{|E^-|} \sum_{e_i^- \in E^-} S(q, e_i^-)$. Ở đây $S(q, e_i^+) = S(e, r, e_i^+)$ với $q$ là truy vấn đuôi và $S(e_i^+, r, e)$ là truy vấn đầu. Tương tự với $S(q, e_i^-)$.

- Cho một truy vấn đầu hoặc truy vấn đuôi liên quan đến thực thể $e$ và quan hệ $r$, chúng ta tính toán ngưỡng $\mu_q$ cho $q$ là $\mu_q = \max\{T[e], T[r], 0\}$.

Nếu $\tilde{e} \in E$ là thực thể được dự đoán bởi mô hình suy luận $M$ cho truy vấn $q$ và $S(q, \tilde{e}) > \mu_q$. CILK sẽ đưa ra câu trả lời $\tilde{e}$. Ngược lại $q$ bị từ chối. 4. Hoạt động của CILK

- Khi người dùng đưa vào truy vấn $q$ bao gồm thực thể chưa xác định $e$ và/hoặc quan hệ $r$, CILK yêu cầu người dùng cung cấp các sự kiện hỗ trợ SFs, tuy nhiên trong một phiên, người dùng chỉ có thể cung cấp được một số ít các sự kiện.

- Để giảm thiểu tỉ lệ tương tác người dùng trong quá trình thu thập kiến thức, CILK sử dụng bộ đệm hiệu suất $P$ và xếp hạng đối ứng trung bình $MRR$ để đo hiệu suất mô hình suy luận M. Vào mỗi phiên mô hình phát hiện các truy vấn liên quan đến $e$ và $r$ có $MRR < p\%$, hệ thống yêu cầu người dùng cung cấp các sự kiện liên quan.

- Thuật toán CILK học và suy luận.

**Algorithm 1** CILK Knowledge Learning and Inference

**Input:** query $q_j = (e, r, ?)$ *or* $(?, r, e)$ issued by user at session-$j$; $\mathcal{K}_j$: CILK's KB at session-$j$; $\mathcal{P}_j$: Performance Buffer at session-$j$; $\mathcal{T}_j$: Threshold Buffer at session-$j$; $\mathcal{M}_j$: trained Inference Model at session-$j$; $\alpha$: probability of treating an acquired supporting fact as training triple; $\rho$: % of entities or relations in $\mathcal{K}_j$ that belong to the diffident set.
**Output:** $\widetilde{e}$ : predicted entity as answer of query $q_j$ in session-$j$.

**if** $r \notin \mathcal{K}_j$ or **IsDiffident**$(r, \mathcal{P}_j, \rho)$ **then**
  $C_r \leftarrow$ **AskUserforCLUE**$(r)$ {acquire supporting facts to learn $r$'s embedding}
**end if**
**if** $e \notin \mathcal{K}_j$ or **IsDiffident**$(e, \mathcal{P}_j, \rho)$ **then**
  $F_e \leftarrow$ **AskUserforEntityFacts**$(e)$ {Acquire supporting facts to learn $e$'s embedding}
**end if**
**if** $C_r \neq \emptyset$ **then**
  $\mathcal{K}_{(j+\frac{1}{2})} \leftarrow$ Add clue triples from $C_r$ into $\mathcal{K}_j$ and randomly mark $\alpha$% of $C_r$ as training triples and (1-$\alpha$)% as validation triples respectively in $\mathcal{K}_j$.
**end if**
**if** $F_e \neq \emptyset$ **then**
  $\mathcal{K}_{j+1} \leftarrow$ Add fact triples from $F_e$ into $\mathcal{K}_{(j+\frac{1}{2})}$ and randomly mark $\alpha$ % of these triples as training triples and (1-$\alpha$) % as validation triples.
**end if**
$D^r_{tr}, D^r_{vd} \leftarrow$ **SampleTripleSet**$(\mathcal{K}_{j+1}, r)$
$D^e_{tr}, D^e_{vd} \leftarrow$ **SampleTripleSet**$(\mathcal{K}_{j+1}, e)$
$\mathcal{M}_{j+1} \leftarrow$ **TrainInfModel**$(\mathcal{M}_j, D^r_{tr} \cup D^e_{tr})$
$\mathcal{P}_{j+1}, \mathcal{T}_{j+1} \leftarrow$ **UpdatePerfandThreshBuffer**
$(\mathcal{M}_{j+1}, (D^r_{vd} \cup D^e_{vd}), \mathcal{P}_j, \mathcal{T}_j)$
$\widetilde{e} \leftarrow$ **PredictAnswerEntity**$(\mathcal{M}_{j+1}, q_j, \mathcal{T}_{j+1})$

5. Cài đặt mô hình

Mô hình này cài đặt tập trung vào bộ dữ liệu **WordNet 18** bao gồm 3 file chính:

- $Wordnet\_edgelist\_pra0.tsv$ lưu trữ cơ sở tri thức $KB$ với định dạng h, t và r. Có 66338 bộ 3, số lượng thực thể 13150, số quan hệ là 12 24-nếu tính các bộ 3 đảo ngược (Theo như bài báo-chưa thống kê chính xác).
- $user\_wordnet.txt$ lưu trữ cơ sở tri thức $K_u$ của người dùng (tập các sự kiện hỗ trợ) có định dạng $\_r \rightarrow s1-; -t1\#\#s2-; -t2\#\#...\#\#END$.
- $test\_wordnet.txt$ tập câu truy vấn của người dùng định dạng giống $K_u$ (18 dòng/mỗi dòng ứng với một quan hệ r).

Cài đặt các tham số cho mô hình CILK (Lưu trữ trong file $controller\_module\_5$):

- $\alpha = 0.9$ tỉ lệ 9:1 giữa tập huấn luyện và tập đào tạo.
- Số vòng lặp $init\_train\_epoch = 100$, learning rate $lr = 0.001$, batch size 128.
- Kích thước tối đa của tập thực thể $max\_ent\_vocab\_size = 17000$, Kích thước tối đa của tập quan hệ $max\_rel\_vocab\_size = 60$, Số chiều nhúng thực thể và quan hệ $embd\_dim = 250$.