

Chương 1

Phương pháp đề xuất

sdfsdf

1.1 AnyBURL

Anytime Bottom-Up Rule Learning for Knowledge Graph Completion
[3] sdfsdf

Dùng lệnh để trích dẫn một hoặc nhiều tài liệu Lưu ý khi trích dẫn tài liệu tham khảo, cần viết câu sao cho bỏ phần trong cặp ngoặc vuông đi thì câu vẫn đầy đủ ý nghĩa. Ví dụ, thay Một ví dụ khác, thay vì viết “... như trong công trình nghiên viết “... nh’

Để chèn mã nguồn, cần dùng package listings

1.2 CGAT

Ở phần này chúng tôi sẽ giới thiệu tóm lược về mạng đồ thị chú ý (GATs [9]) và cải tiến của chúng tôi trên lớp GAT mà chúng tôi gọi là *mạng cộng tác đồ thị chú ý* CGAT, và sau đó chúng tôi áp dụng vào để xây dựng đồ thị theo mô hình KGAT [6] để tối ưu quá trình dự đoán các mối quan hệ.

1.2.1 Mô hình GAT

Mạng đồ thị tích chập (Graph Convolutional Networks GCNs) giúp tổng hợp thông tin bằng cách tính trung bình thông tin từ các thực thể lân cận, tuy nhiên cách này sẽ làm cho các thực thể có trọng số ngang bằng nhau không biểu diễn đúng thông tin trong thế giới thực. Để giải quyết vấn đề đó, GATs [9] ra đời để đối xử với các thực thể lân cận bằng sự quan trọng của chúng.

Đầu vào của mô hình là vector biểu diễn đặc trưng của từng thực thể (entity) $E = \{\vec{e}_1, \vec{e}_2, \dots, \vec{e}_N\}$. Và mục tiêu của chúng ta là biến đổi thành một đặc trưng đầu ra mới $E'' = \{\vec{e}_1'', \vec{e}_2'', \dots, \vec{e}_N''\}$; với \vec{e}_i và $\vec{e}_i'' \in \mathbb{R}^T$ tương ứng là vector nhúng đầu vào và vector đầu ra của của thực thể e_i , N là số lượng của các thực thể, T là số miền nhúng đặc trưng đầu vào.

Mô hình sẽ đi qua hai quá trình biến đổi vector đặc trưng \vec{e}_i và có thể tóm lược như sau :

$$\vec{e}_i \longrightarrow \vec{e}_i' \longrightarrow \vec{e}_i'' \quad (1.1)$$

Ở quá trình biến đổi đầu tiên, mô hình sẽ tổng hợp thông tin từ các thực thể lân cận và ghép chồng lên nhau để tạo ra vector \vec{e}_i' sau đó mô hình sẽ dùng vector \vec{e}_i' để coi là vector nhúng của thực thể cho lớp mới và tiếp tục quá trình tổng hợp từ các thông tin lân cận và tạo ra vector \vec{e}_i'' cuối cùng.

Đầu tiên để tham số hóa quá trình biến đổi tuyến tính, ta cần một trọng số $W \in \mathbb{R}^{N_e \times T}$ ánh xạ vector đầu vào thành một vector mới với miền không gian lớn hơn và một hàm chú ý a chúng ta tùy chọn :

$$e_{ij} = a(W\vec{e}_i, W\vec{e}_j) \quad (1.2)$$

trong đó e_{ij} là giá trị chú ý của một cạnh (e_i, e_j) trong đồ thị \mathcal{G} hay e_{ij} thể hiện sự quan trọng của đặc trưng cạnh (e_i, e_j) so với thực thể e_i . Sau

đó, chúng ta áp dụng hàm *softmax* qua tất cả các giá trị nhúng của hàng xóm để tạo ra α_{ij} . Quá trình tổng hợp các sự chú ý được thể hiện ở biểu thức sau :

$$\vec{a}_{ij} = \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{e}_j \right) \quad (1.3)$$

Tiếp theo mô hình GAT sẽ đi qua *lớp chú ý đa đỉnh* (multi-head attention) để ổn định quá trình học bằng cách ghép A đỉnh chú ý với nhau :

$$\vec{x}_i = \parallel_{a=1}^A \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^a \mathbf{W}^a \vec{x}_j \right) \quad (1.4)$$

trong đó phép \parallel biểu diễn quá trình ghép chồng lên nhau và σ là bất kỳ hàm biến đổi phi tuyến tính nào, α_{ij}^a là hệ số chú ý được chuẩn hóa của cạnh (e_i, e_j) được tính từ lớp thứ a^{th} cơ chế chú ý. Cuối cùng \vec{x}_i được coi là vector thực thể nhúng mới và cho vào lớp chú ý đa đỉnh với đầu ra thay vì ghép chồng như trên thì được tính trung bình như công thức sau :

$$\vec{x}_i'' = \sigma \left(\frac{1}{A} \sum_{a=1}^A \sum_{j \in \mathcal{N}_i} \alpha_{ij}^a \mathbf{W}^a \vec{x}_j \right) \quad (1.5)$$

1.2.2 Knowledge Base Attention Network (KBAT)

Trong phần này sẽ trình bày về mô hình Knowledge Base Attention (KBAT [6]), là một mô hình cải tiến dựa trên mô hình GAT ở trên bằng cách lấy thêm thông tin của mối quan hệ trên một cạnh. Mục tiêu của mô hình là tạo ra hai ma trận mới biểu thị cho **ma trận nhúng thực thể**

và **ma trận nhúng quan hệ**. Mô hình sẽ biến đổi từ ma trận thực thể $\mathbf{E} \in \mathbb{R}^{N_e \times T}$ và ma trận quan hệ $\mathbf{R} \in \mathbb{R}^{N_r \times P}$ để tạo thành một ma trận thực thể mới $\mathbf{E}' \in \mathbb{R}^{N_e \times T'}$ và ma trận quan hệ mới $\mathbf{R}' \in \mathbb{R}^{N_r \times P'}$, trong đó với mỗi dòng thứ i^{th} trong ma trận với trọng số tương ứng biểu thị cho một thực thể e_i hoặc một mối quan hệ r_i ; N_e hoặc N_r tương ứng là số lượng của tập thực thể và tập quan hệ; trong khi T và P tương ứng là số chiều đầu vào của vector nhúng; thì T' và P' tương ứng là số chiều của vector nhúng đầu ra. Các ma trận trọng số trên được khởi tạo ngẫu nhiên theo phân phối chuẩn hoặc tái huấn luyện(pre-train) bằng mô hình TransE [4] .

Tuy nhiên một cạnh trong đồ thị (\vec{e}_i, \vec{e}_j) Tuy nhiên một vector \vec{e}_i hay \vec{r}_k không thể biểu thị một cách đầy đủ tri thức của một thực thể e_i hay quan hệ r_k trong một đồ thị, vì một tri thức sẽ có thể có mối liên hệ với những tri thức lân cận hay nói cách khác một **thực thể nhúng** (entity embedding) hay một **quan hệ nhúng** (relation embedding) sẽ cần thêm thông tin của các vector nhúng của thực thể lân cận và quan hệ lân cận khác để có thể là một đại diện đầy đủ . Chính vì vậy phương pháp *n-hop neighborhood* [0] sẽ giúp bổ sung thêm thông tin lân cận của thực thể e_i và quan hệ r_k bằng cách ghép chồng để tạo thành một vector mới theo công thức sau :

$$\vec{h}_i = [\vec{e}_i \parallel \vec{e}_{i_{n\text{-hop}}}] \quad ; \quad \vec{g}_k = [\vec{r}_k \parallel \vec{r}_{k_{n\text{-hop}}}] \quad (1.6)$$

axis=1 axis=1

Trong đó \vec{h}_i, \vec{g}_k tương ứng là vector nhúng mới của thực thể e_i và các thực thể lân cận ($e_{i_{n\text{-hop}}}$) hay quan hệ r_k và các quan hệ lân cận ($r_{k_{n\text{-hop}}}$); ký hiệu $\parallel_{\text{axis}=1}$ biểu thị cho phép xếp chồng lên nhau. Các vector $\vec{e}_{i_{n\text{-hop}}}$ và $\vec{r}_{k_{n\text{-hop}}}$ được tính bằng tổng vector nhúng của các thực thể hoặc các quan hệ lân cận đi qua $n - \text{hop}$ độ sâu bắt đầu từ e_i theo công thức sau :

$$\overrightarrow{e_{i_{n\text{-hop}}}} = \prod_{d=1}^{n\text{-hop}} \sum_{n \in \mathcal{N}_i} \overrightarrow{e_n^d} \quad ; \quad \overrightarrow{r_{k_{n\text{-hop}}}} = \prod_{d=1}^{n\text{-hop}} \sum_{m \in \mathcal{N}_k} \overrightarrow{r_m^d} \quad (1.7)$$

Với mỗi thực thể e_n hay quan hệ r_m có độ sâu d (depth) bắt đầu từ thực thể e_i , ta sẽ tính tổng các vector nhúng và ghép chồng với nhau.

Để biểu thị cho quá trình biến tuyến tính của vector nhúng, ta cho mỗi vector nhúng đi qua một ma trận trọng số : $\overrightarrow{h_i} = \mathbf{W}_{\text{entity}} \overrightarrow{h_i}$ và $\overrightarrow{g_i} = \mathbf{W}_{\text{relation}} \overrightarrow{g_i}$. Tuy nhiên để thu được một trọng số mới biểu diễn bộ ba $t_{ij}^k = (e_{\text{head}}, \text{relation}, e_{\text{tail}})$ tương ứng với một cạnh trong KB. Ta thực hiện quá trình biến đổi tuyến tính đó bằng cách ghép cả thực thể và mối quan hệ với nhau rồi nhân với một ma trận trọng số chung như công thức sau :

$$\overrightarrow{t_{ijk}} = \mathbf{W}_1 [\overrightarrow{h_i} || \overrightarrow{h_j} || \overrightarrow{g_k}] \quad (1.8)$$

trong đó $\overrightarrow{t_{ijk}}$ là một vector nhúng biểu diễn cho một bộ ba t_{ij}^k . Các vector $\overrightarrow{h_i}$, $\overrightarrow{h_j}$ và $\overrightarrow{g_j}$ tương ứng là vector nhúng của các thực thể e_i , e_j và quan hệ r_k . $\mathbf{W}_1 \in \mathbb{R}^{3T \times S^{\text{batch}}}$. Để áp dụng cơ chế chú ý (attention mechanisms [8]), ta cần học sự quan trọng của một cạnh t_{ij}^k bởi giá trị của b_{ij}^k . Để tham số hóa quá trình biến đổi tuyến tính ta nhân với ma trận trọng số \mathbf{W}_2 và lấy giá trị chú ý tuyệt đối bằng hàm *LeakyRelu* theo công thức sau :

$$b_{ijk} = \text{LeakyRelu}(\mathbf{W}_2 \overrightarrow{t_{ijk}}) \quad (1.9)$$

Sau đó, với mỗi độ lớn của b_{ij}^k được cho qua hàm *softmax* để tính giá trị thể hiện xác suất của từng giá trị chú ý α_{ijk} đối với từng bộ ba .

$$\alpha_{ijk} = \text{softmax}_{jk}(b_{ijk}) = \frac{\exp(b_{ijk})}{\sum_{n \in \mathcal{N}_i} \sum_{r \in \mathcal{R}_{in}} \exp(b_{inr})} \quad (1.10)$$

trong đó \mathcal{N}_i là tập hợp những thực thể kế cận của e_i ; \mathcal{R} tập hợp những quan hệ nối hai thực thể e_i và e_j . Để tăng cường đặc trưng cạnh tương ứng với từng xác xuất, ta sẽ nhân giá trị chú ý tương đối α_{ijk} với vector nhúng \vec{t}_{ijk} đã được biến đổi tuyến tính ban đầu. Mô hình sẽ tạo ra vector thực thể nhúng mới với đã được tổng hợp từ các thực thể lân cận.

$$\vec{h}_i = \sigma \left(\sum_{j \in \mathcal{N}_i} \sum_{k \in \mathcal{R}_{ij}} \alpha_{ijk} \vec{t}_{ijk} \right) \quad (1.11)$$

Tương tự như mạng đồ thị chú ý ở trên, mô hình sẽ đi qua lớp chú ý đa đỉnh để ổn định quá trình học bằng cách ghép A đỉnh chú ý với nhau theo công thức sau :

$$\vec{h}_i = \parallel_{m=1}^M \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ijk}^m \vec{t}_{ijk}^m \right) \quad (1.12)$$

Như quá trình biến đổi tuyến tính của ma trận mỗi quan hệ ở trên, ma trận R đầu vào sẽ được nhân với một vector nhúng W quan hệ để tạo ra một ma trận mới biểu thể tập hợp các quan hệ nhúng mới :

$$R' = R\mathbf{W}^{\mathbf{R}}; \quad \text{Trong đó : } \mathbf{W}^{\mathbf{R}} \in \mathbb{R}^{P \times P'} \quad (1.13)$$

Trong đó $\mathbf{W}^{\mathbf{R}}$ là ma trận trọng số thể hiện quá trình biến đổi tuyến tính, P và P' lần lượt là số chiều của vector quan hệ nhúng đầu vào và đầu ra. Và trong trường hợp này P' phải bằng số chiều của vector nhúng thực

thể.

Đến đây, ta đã có hai ma trận H' và R' tương ứng là ma trận quan hệ và ma trận thực thể. Mô hình sẽ đi qua lớp cuối chú ý cùng bằng cách tiếp tục đi qua lớp mạng đồ thị chú ý tương tự ở trên nhưng thay vì ghép chồng thì ta thực hiện tính trung bình theo công thức sau :

$$\vec{h}_i'' = \sigma \left(\frac{1}{M} \sum_{m=1}^M \sum_{j \in \mathcal{N}_i} \sum_{k \in \mathcal{R}_{ij}} \alpha_{ijk}'^m t_{ijk}'^m \right) \quad (1.14)$$

Tương tự như trên, nhưng $\alpha_{ijk}'^m$ và $t_{ijk}'^m$ tương ứng là các vector được nhúng trên các mối quan hệ và thực thể mới.

Sau quá trình nhúng trên, ta sẽ có một ma trận thực thể H'' và quan hệ R'' mới đã được tổng hợp thêm một lần nữa. Tuy nhiên ta muốn biểu diễn ma trận thực thể với số chiều mới, ta thực hiện phép biến đổi sau :

$$\mathbf{E}' = \mathbf{W}^E \mathbf{E}^t + \mathbf{H}'' \quad (1.15)$$

Trước và sau mỗi lớp mạng đồ thị chú ý chúng tôi đề thực hiện chuẩn hóa.

Sau quá trình trên chúng ta đã tạo ra được ma trận nhúng thực thể và quan hệ.

Our model borrows the idea of a translational scoring function from (Bordes et al., 2013), which learns embeddings such that for a given valid triple $t_{ij}^k = (e_i, r_k, e_j)$, the condition $\vec{h}_i + \vec{g}_k \approx \vec{h}_j$ holds, i.e., e_j is the nearest neighbor of e_i connected via relation r_k .

Specifically, we try to learn entity and relation embeddings to minimize the L1-norm dissimilarity measure given by $d_{t_{ij}} = \|\vec{h}_i + \vec{g}_k - \vec{h}_j\|_1$.

We train our model using hinge-loss which is given by the following expression

$$L(\Omega) = \sum_{t_{ij} \in S} \sum_{t'_{ij} \in S'} \max\{d_{t'_{ij}} - d_{t_{ij}} + \gamma, 0\} \quad (1.16)$$

where $\gamma > 0$ is a margin hyper-parameter, S is the set of valid triples, and S' denotes the set of invalid triples, given formally as

$$S' = \underbrace{\{t_{ij}^k | e'_i \in \mathcal{E} \setminus e_i\}}_{\text{thay thể thực thể đầu}} \cup \underbrace{\{t_{ij}^k | e'_j \in \mathcal{E} \setminus e_j\}}_{\text{thay thể thực thể đuôi}} \quad (1.17)$$

where ω^m represents the m -th convolutional filter, W is a hyper-parameter denoting number of filters used, $*$ is a convolution operator, and $\mathbf{W} \in \mathbb{R}^{\Omega k^1}$ represents a linear transformation matrix used to compute the final score of the triple. The model is trained using soft-margin loss as

$$\mathcal{L} = \sum_{t_{ij}^k \in \{S \cup S'\}} \log(1 + \exp(l_{t_{ij}^k} \cdot f(t_{ij}^k))) + \frac{\lambda}{2} \|\mathbf{W}\|_2^2 \quad (1.18)$$

$$\text{where } l_{t_{ij}^k} = \begin{cases} 1 & \text{for } t_{ij}^k \in S \\ -1 & \text{for } t_{ij}^k \in S' \end{cases}$$

Attention

CAttention [5]

ConvKB [7]

Chương 2

Kết quả thực nghiệm và phân tích

sdfsdf

2.1 Tập dữ liệu

Tập dữ liệu bao gồm :

Dataset	Entities	Relations	Edges			
			Training	Validation	Test	Total
WN18RR	40,943	11	86,835	3034	3134	93,003
FB15k-237	14,541	237	272,115	17,535	20,466	310,116

Bảng 2.1: Mô tả tập dữ liệu

2.2 Phương pháp đánh giá

sdfsdf

	WN18RR					FB15K-237				
	MR	MRR	Hits@N			MR	MRR	Hits@N		
			@1	@3	@10			@1	@3	@10
TransE	7000	0.444	41.2	47	50.4	7000	0.444	41.2	47	50.4
ConvKB	7000	0.444	41.2	47	50.4	7000	0.444	41.2	47	50.4
Mô hình của chúng tôi										
AnyBURL	7000	<u>0.444</u>	41.2	47	<u>50.4</u>	7000	0.444	41.2	47	50.4
CGAT	7000	0.444	41.2	47	50.4	7000	0.444	41.2	47	50.4

Bảng 2.2: Kết quả thực nghiệm trên tập WN18RR và FB15K-237. Kết quả tốt nhất được **bôi đen** và kết quả tốt thứ hai được gạch chân

2.3 Phương pháp huấn luyện

sdfsdf

2.4 Kết quả và phân tích

sdfsdf

Chương 3

Kết luận

EXPERIMENTS RESULTS AND ANALYSIS

Dùng lện

Danh mục công trình của tác giả

1. Tạp chí ABC
2. Tạp chí XYZ

Tài liệu tham khảo

Phương pháp AnyBURL

- [3] Meilicke, Christian et al. “Anytime Bottom-Up Rule Learning for Knowledge Graph Completion.” In: *IJCAI*. 2019, pp. 3137–3143.

Phương pháp CGAT

- [0] In:
- [4] Bordes, Antoine et al. “Translating embeddings for modeling multi-relational data”. In: *Advances in neural information processing systems*. 2013, pp. 2787–2795.
- [5] Cordonnier, Jean-Baptiste, Loukas, Andreas, and Jaggi, Martin. “Multi-Head Attention: Collaborate Instead of Concatenate”. In: *arXiv preprint arXiv:2006.16362* (2020).
- [6] Nathani, Deepak et al. “Learning attention-based embeddings for relation prediction in knowledge graphs”. In: *arXiv preprint arXiv:1906.01195* (2019).
- [7] Nguyen, Dai Quoc et al. “A novel embedding model for knowledge base completion based on convolutional neural network”. In: *arXiv preprint arXiv:1712.02121* (2017).
- [8] Vaswani, Ashish et al. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

- [9] Veličković, Petar et al. “Graph attention networks”. In: *arXiv preprint arXiv:1710.10903* (2017).

Phụ lục A

Kết quả training

Đây là phụ lục.

Phụ lục B

Tập dữ liệu

Đây là phụ lục 2.