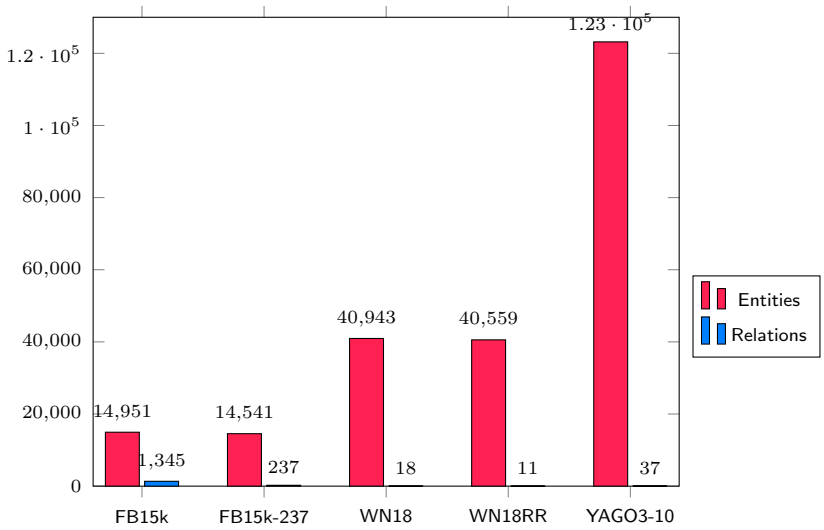
Chương 1. EXPERIMENTS

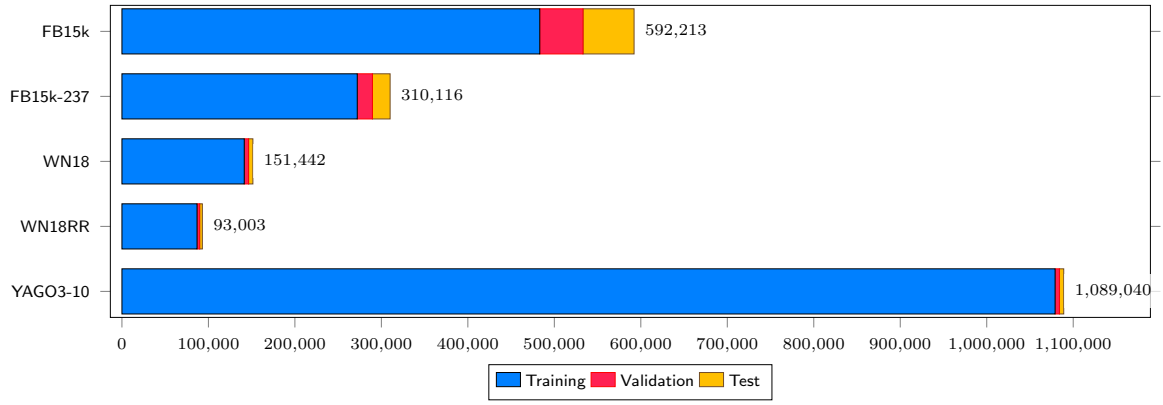


In this section, we describe the datasets used for our empirical evaluation, along with a comparison against notable existing methods as reported in **??**. Additionally, we evaluate our two proposed approaches for injecting new knowledge into the knowledge graph. Specifically, we treat the test set as a batch of new knowledge to be added, and use the validation set to re-evaluate the effectiveness of our method. Detailed results are presented in Table 1.2 and Table 1.3.

			# Edges		
Dataset	Entities	Relations	Training	Validation	Test
FB15k	14,951	1,345	483,142	50,000	59,071
FB15k-237	14,541	237	272,115	17,535	20,466
WN18	40,943	18	141,442	5,000	5,000
WN18RR	40,559	11	86,835	3,034	3,134
YAGO3-10	123,182	37	1,079,040	5,000	5,000

Bảng 1.1: Dataset Information

1.1 Training Datasets



In our experiments, we evaluate our approach on four widely used benchmark datasets: FB15k, FB15k-237 ([6]), WN18, and WN18RR ([3]). Each dataset is divided into three subsets: training, validation, and test sets. Detailed statistics for these datasets are presented in Table 1.1.

Each dataset consists of a collection of triples in the form $\langle head, relation, tail \rangle$. FB15k and WN18 are derived from the larger knowledge bases FreeBase and WordNet, respectively. However, they contain a large number of inverse relations, which allow most triples to be easily inferred. To address this issue and to better reflect real-world link prediction scenarios, FB15k-237 and WN18RR were constructed by removing such inverse relations.

1.1.1 Bộ dữ liệu FB15k

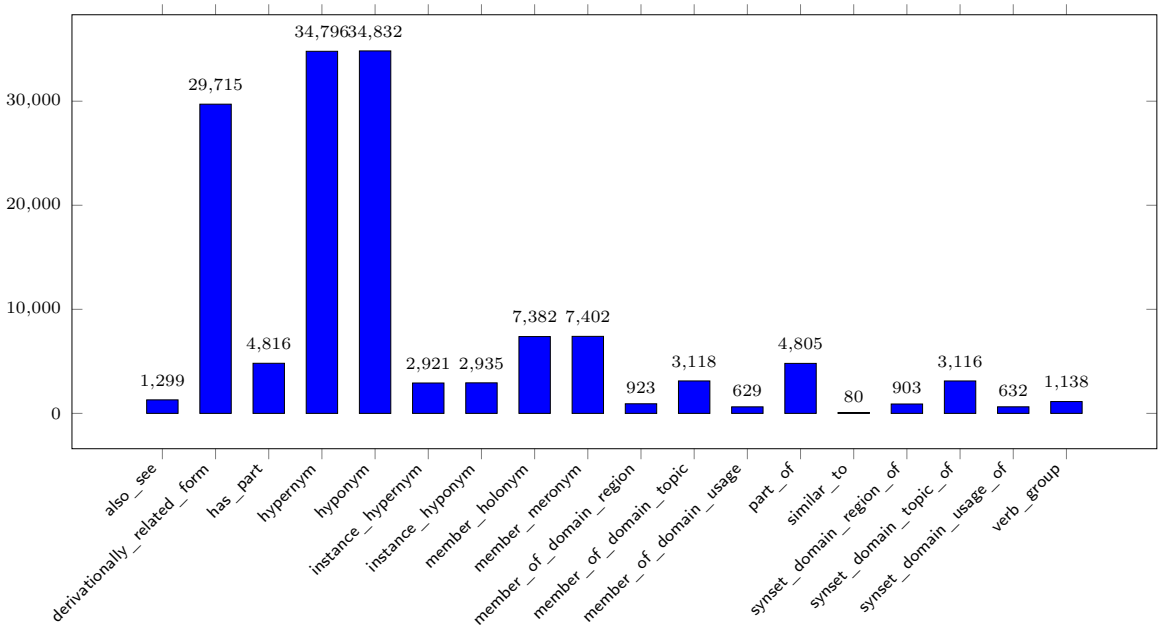
Bộ dữ liệu này được tạo bởi nhóm nghiên cứu A. Bordes, N. Usunier [1] bằng cách trích xuất từ bộ dữ liệu Wikilinks database [1].Wikilinks database thu thập các siêu liên kết (hyperlinks) đến Wikipedia gồm 40 triệu lượt đề cập trên 3 triệu thực thể, họ trích xuất tất cả các dữ kiện liên quan đến một thực thể nhất định có hơn 100 lần được đề cập đến bởi các tài liệu khác cùng với tất cả các dữ kiện liên quan đến thực thể đó (bao gồm cả những thực thể con được nhắc đến trong tài liệu Wikipedia đó), ngoại trừ những thông tin như: ngày tháng, danh từ riêng, v.v ... Họ cũng chuyển đổi các đỉnh có bậc n được biểu diễn thành các nhóm các cạnh nhị phân tức là liệt kê các cạnh và quan hệ của mọi đỉnh.

[1]https://code.google.com/archive/p/wiki-links/

1.1.2 Bộ dữ liệu FB15k-237

Bộ dữ liệu này là một tập hợp con của FB15k được xây dựng bởi Toutanova và Chen [6] lấy cảm hứng từ quan sát rằng FB15k bao gồm dữ liệu thử nghiệm được các mô hình nhìn thấy tại thời điểm đào tạo (test leekage). Trong FB15k, vấn đề này là do sự hiện diện của các quan hệ gần giống nhau hoặc nghịch đảo của nhau.FB15k-237 được xây dựng để trở thành một tập dữ liệu thách thức hơn: các tác giả đã chọn các dữ kiện liên quan đến 401 quan hệ xuất hiện nhiều nhất và loại bỏ tất cả các quan hệ tương đương hoặc nghịch đảo. Họ cũng đảm bảo rằng không có thực thể nào được kết nối trong tập huấn luyện cũng được liên kết trực tiếp trong tập test và validation.

1.1.3 Bộ dữ liệu WN18



Bộ dữ liệu này được giới thiệu bởi các tác giả của TransE [1], được trích xuất từ WordNet[2], một bản thể học ngôn ngữ KG có nghĩa là cung cấp một từ điển/từ đồng nghĩa để hỗ trợ NLP và phân tích văn bản tự động. Trong WordNet, các thực thể tương ứng với các tập hợp (*word senses*) và các quan hệ đại diện cho các kết nối từ vựng của chúng (ví dụ: "hypernym"). Để xây dựng WN18, các tác giả đã sử dụng WordNet làm điểm bắt đầu và sau đó lặp đi lặp

[2]https://wordnet.princeton.edu/

lại lọc ra các thực thể và mối quan hệ với quá ít lần được đề cập.

1.1.4 Bộ dữ liệu WN18RR

Bộ dữ liệu này là một tập hợp con của WN18 được xây dựng bởi DeŠmers et al.[2], cũng là người giải quyết vấn đề rò rỉ thử nghiệm (test leakage)trong WN18. Để giải quyết vấn đề đó, họ xây dựng tập dữ liệu WN18RR thách thức hơn nhiều bằng cách áp dụng một phương pháp tương tự được sử dụng cho FB15k-237 [6].

1.2 Các độ đo

Trong phần này chúng tôi mô tả lại các phương pháp đánh giá (độ do), môi trường thực hiện cũng như các tập dữ liệu mà chúng tôi sử dụng để dánh giá phương pháp của mình. Các phương pháp đánh giá (độ do) này cũng phổ biến nó được đánh giá cho hầu hết các mô hình dự đoán liên kết trên đồ thị. Chúng tôi tiến hành so sánh với bốn phương pháp nổi bật khác được báo cáo trong [5].

1.2.0.1 Độ đo Hit@K (H@K)

Đó là tỷ lệ các dự đoán đúng mà rank nhỏ hơn hoặc bằng ngưỡng K:

$$H@K = \frac{\mid q \in Q : rank(q) \leq K \mid}{\mid Q \mid}$$

1.2.0.2 Mean Rank (MR)

Đây là giá trị trung bình của rank thu được cho một dự đoán chính xác. Càng nhỏ thì mô hình càng chính xác:

$$MR = \frac{1}{\mid Q \mid} \sum_{q \in Q} rank(q)$$

Trong đó $\mid Q \mid$ là độ lớn của tập hợp các câu hỏi bằng độ lớn của tập test hoặc vadidation. Khi dự đoán chúng tôi dự đoán cả head và tail cho một dòng tương ứng trong tập dữ liệu thử nghiệm. Ví dụ chún tôi sẽ dự đoán $\langle ?, \ relation, \ tail \rangle$ và $\langle head, \ relation, \ ? \rangle$ cho 1 dòng tương ứng, q thể hiện cho câu hỏi chúng tôi dự đoán và $rank(q)$ thể hiện cho kết quả đúng của câu hỏi đứng ở vị trí thứ mấy trong xếp hạng của chúng tôi sau đó lấy trung bình rank của các dự đoán head

và tail. Rõ ràng độ đo này nằm giữa [1,| số lượng các entity |] do có tối da n cạnh nối 1 đỉnh tới $n-1$ đỉnh còn lại và thêm cạnh nối tới chính đỉnh nó(cạnh khuyên). Và độ đo này dễ bị ảnh hưởng bởi nhiễu vì có những quan hệ có những thực thể được xếp hạng gần cuối. Để giải quyết vấn đề này nhóm chúng tôi và các nhóm nghiên cứu khác sử dụng thêm độ đo Mean Reciprocal Rank (MMR)

1.2.0.3 Mean Reciprocal Rank (MMR)

Đây là xếp hạng đối ứng trung bình, là nghịch đảo của giá trị trung bình của rank thu được cho một dự đoán chính xác ở trên. Và càng lớn thì mô hình càng chính xác. Do độ đo này lấy nghịch đảo của các rank nên tránh dược vấn đề nhiễu của độ đo MR ở trên.

$$MRR = \frac{1}{\mid Q \mid} \sum_{q \in Q} \frac{1}{rank(q)}$$

1.3 Phương pháp huấn luyện

1.3.1 Huấn luyện trên mô hình KBGAT

Đầu tiên chúng tôi khởi tạo các vector nhúng bằng mô hình TransE ([1]). Để tạo ra các bộ ba không hợp lệ, chúng tôi thay thế các thực thể đầu và thực thể đuôi bằng một thực thể khác được lấy ngẫu nhiên trong tập thực thể .

Sau đó chúng tôi chia ra làm hai phần huấn luyện, phần đầu tiên được xem như mã hóa (encoder) giúp biến đổi các vector nhúng khởi tạo ban đầu thành các vector nhúng mới tổng hợp thông tin các nút lân cận bằng mô hình KBGAT để tạo ra các vector nhúng của thực thể và quan hệ. Phần thứ hai được xem như quá trình giải mã (decoder) để thực hiện nhiệm vụ dự đoán, bằng cách lấy thêm thông tin của n-hop giúp chúng tôi tổng hợp thêm thông tin từ các thực thể lân cận, ngoài ra chúng tôi còn sử dụng quan hệ phụ trợ để tổng hợp thêm thông tin hàng xóm trong đồ thị thưa. Chúng tôi sử dụng hàm tối ưu Adam với tốc độ học $\mu = 0.001$. Số chiều cuối cùng của cả thực thể và quan hệ đều bằng 200. Cụ thể các siêu tham số ưu được tìm kiếm bằng thuật toán tìm kiếm lưới (grid search) được trình bày ở **??**.

1.4 Kết quả thực nghiệm

Như đã nói trước đây với mô hình dựa trên luật của chúng tôi hoàn toàn có thể thực hiện trên một laptop với cấu hình thông thường. Trong thí nghiệm của chúng tôi cấu hình máy để thực thi như sau: T480, core i5 8th Gen, ram 16GB, 4 core 8 thread. Mã nguồn thực thi được viết bằng ngôn ngữ Python phiên bản 3.6 và dùng các hàm hỗ trợ có sẵn trong Python với không một thư viện bên thứ ba nào. Thí nghiệm được thực hiện với bốn tập dữ liệu phổ biến là FB15k, FB15-237, WN18 và WN18RR. Thông tin chi tiết các bộ dữ liệu này được mô tả ở Section 1.1 các tập dữ liệu huấn luyện.

Như mô tả ở phần **??**, thuật toán AnyBURL này sẽ học các luật được sinh ra trong một khoảng thời gian nhất định do người dùng cấu hình. Ở đây chúng tôi chọn cấu hình thời gian là 1000 giây tương đương khoảng 17 phút đào tạo, với độ bão hòa (SAT) 0.85, độ tin cậy Q 0.05, kích thước mẫu S ($\frac{1}{10}$ tập huấn luyện). Với cấu hình như vậy mô hình phiên bản Python của chúng tối cho kết quả tương đương với phiên bản Java nhóm tác giả Meilicke, Christian et al. [4] với cấu hình tương tự nhưng thời gian training là 100 giây. Sự khác biệt về thời gian học tập ở đây chủ yếu là do hiệu năng của hai ngôn ngữ Python và Java. Ở đây chúng tôi chọn ngôn ngữ Python vì nó được dùng làm ngôn ngữ chính cho nhiều mô hình trí tuệ nhân tạo gần đây, và cũng thuận tiện cho chúng tôi khi so sánh hiệu năng cũng như đánh giá với các phương pháp học sâu khác đa số được viết bằng Python.

	FB15k				**FB15k-237**			
	H@1	**H@10**	**MR**	**MRR**	**H@1**	**H@10**	**MR**	**MRR**
ComplEx	81.56	90.53	34	0.848	25.72	52.97	202	0.349
TuckER	72.89	88.88	39	0.788	25.90	53.61	162	0.352
TransE	49.36	84.73	45	0.628	21.72	49.65	209	0.31
RoteE	73.93	88.10	42	0.791	23.83	53.06	178	0.336
ConvKB	59.46	84.94	51	0.688	21.90	47.62	281	0.305
KBGAT	70.08	91.64	38	0.784	36.06	58.32	211	0.4353
AnyBURL	79.13	82.30	285	<u>0.824</u>	20.85	42.40	490	0.311

Bảng 1.2: Kết quả thực nghiệm trên tập FB15k, FB15k-237

	WN18				WN18RR			
	H@1	**H@10**	**MR**	**MRR**	**H@1**	**H@10**	**MR**	**MRR**
ComplEx	94.53	95.50	3623	0.349	42.55	52.12	4909	0.458
TuckER	94.64	95.80	510	0.951	42.95	51.40	6239	0.459
TransE	40.56	94.87	279	0.646	2.79	94.87	279	0.646
RoteE	94.30	96.02	274	0.949	42.60	57.35	3318	0.475
ConvKB	93.89	95.68	413	0.945	38.99	50.75	4944	0.427
KBGAT					35.12	57.01	1974	0.4301
AnyBURL	93.96	95.07	**230**	**0.955**	**44.22**	54.40	2533	0.497

Bảng 1.3: Kết quả thực nghiệm trên tập WN18, WN18RR

Table 1.2, và Table 1.3 mô tả các kết quả thực nghiệm của chúng tôi với các độ đo $H@K$ cùng với các kết quả thực nghiệm của các phương pháp khác được đề cập trong khảo sát [5]

		AnyBURL	**Batch edge AnyBURL**	**Edge AnyBURL**
FB-15k	hit@10	82.22	82.48	83.08
	MR	285	250	220
	MRR	0.824	0.853	0.866
FB15k-237	hit@10	42.40	43.40	43.51
	MR	490	472	441
	MRR	0.311	0.353	0.377
WN18	hit@10	95.07	95.09	95.19
	MR	230	229	228
	MRR	0.955	0.955	0.956
WN18RR	hit@10	54.40	54.63	54.70
	MR	2533	2346	2215
	MRR	0.497	0.553	0.581

Bảng 1.4: Kết quả độ chính xác hai chiến lược thêm tri thức mới

Table 1.4 mô tả các kết quả thực nghiệm của chúng tôi với hai chiến lược thêm tri thức mới vào đồ thị. Chúng tôi đánh giá trên tổng số luật được sinh ra, và số luật có độ tin cậy $>= 50\%$ và $>= 80\%$.

		Batch edge AnyBURL	**Edge AnyBURL**
FB15k	num rule	1011	1367
	confidence 50%	416 (41,14%)	1185 (86,69%)
	confidence 80%	284 (28, 09%)	481 (35,18%)
FB15k-237	num rule	1120	756
	confidence 50%	244 (21,79%)	660 (87,30%)
	confidence 80%	95 (8,48%)	162 (21,43%)
WN18	num rule	533	260
	confidence 50%	270 (38, 46 %)	252 (96,92%)
	confidence 80%	240 (34,19%)	225 (86,54%)
WN18RR	num rule	439	106
	confidence 50%	110 (25,05%)	102 (96,22%)
	confidence 80%	83 (18,91%)	85 (81,19%)

Bảng 1.5: Kết quả đánh giá về số luật hai chiến lược thêm tri thức mới

TÀI LIỆU TRÍCH DẪN

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[2] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. *arXiv preprint arXiv:1707.01476*, 2017.

[3] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[4] Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. Anytime Bottom-Up Rule Learning for Knowledge Graph Completion, 2019.

[5] Andrea Rossi, Donatella Firmani, Antonio Matinata, Paolo Merialdo, and Denilson Barbosa. Knowledge graph embedding for link prediction: A comparative analysis. *arXiv preprint arXiv:2002.00819*, 2020.

[6] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, 2015.