



KHOÁ LUẬN TỐT NGHIỆP

MẠNG TÍCH CHẬP ĐỒ THỊ NÂNG CAO NGỮ NGHĨA VÀ CÚ PHÁP CHO PHÂN TÍCH QUAN ĐIỂM MỨC KHÍA CẠNH VÀ ÁP DỤNG VÀO TIẾNG VIỆT

Sinh viên thực hiện: **Nguyễn Thị Thuý Hiền**

Cán bộ hướng dẫn: **PGS.TS Hà Quang Thụy**

Cán bộ đồng hướng dẫn: **ThS Nguyễn Thị Cẩm Vân**

Nội dung chính

- 01** Đặt vấn đề và bài toán khoá luận
- 02** Phương pháp nghiên cứu
- 03** Mô hình khóa luận
- 04** Thực nghiệm và kết quả
- 05** Kết luận

1.


ĐẶT VẤN ĐỀ

Định nghĩa

- Quan điểm là một bộ năm $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ [10].
- Bài toán khai phá quan điểm mức khía cạnh đối với tài liệu d quy về tìm ra tất cả các bộ năm quan điểm xuất hiện trong d .
- Trong thực tế, bài toán quy về tìm bộ
(đích quan điểm, quan điểm)

Điền  Đã mua tại TGDD



 Sẽ giới thiệu cho bạn bè, người thân

Máy dùng ok , cảm giác pin tụt hơi nhanh Nv tv nhiệt tình



Hữu ích



Thảo luận

Đã dùng khoảng 1 ngày ?



Máy tốt giá



Hữu ích

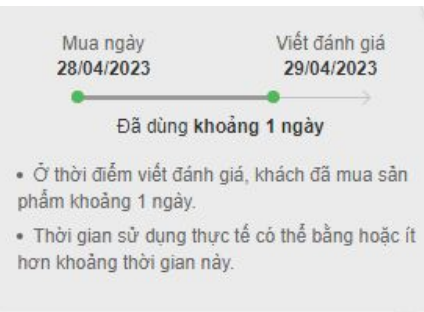
Điền 



Máy dùng o



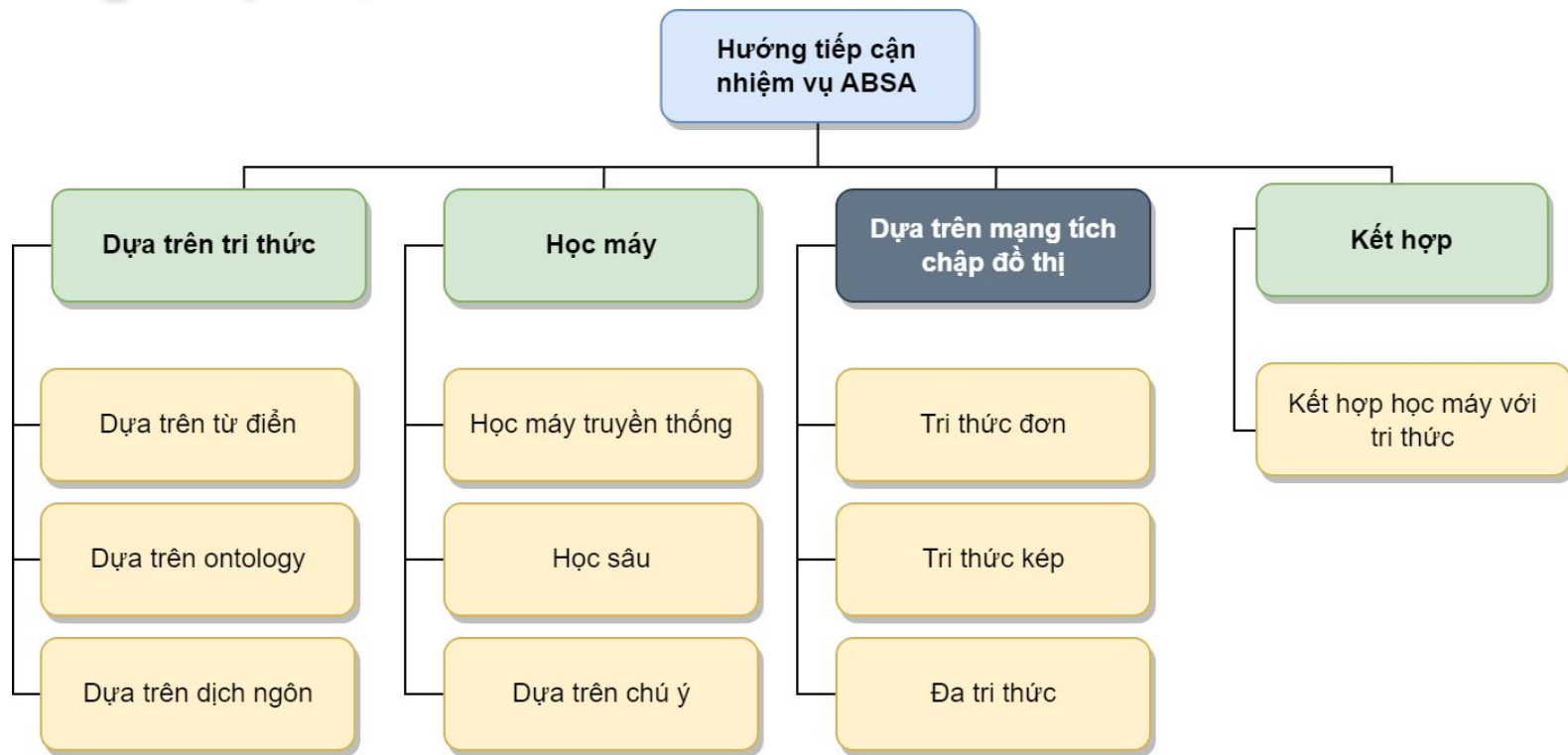
Hữu ích



Thảo luận

Đã dùng khoảng 1 ngày ?

Hướng tiếp cận



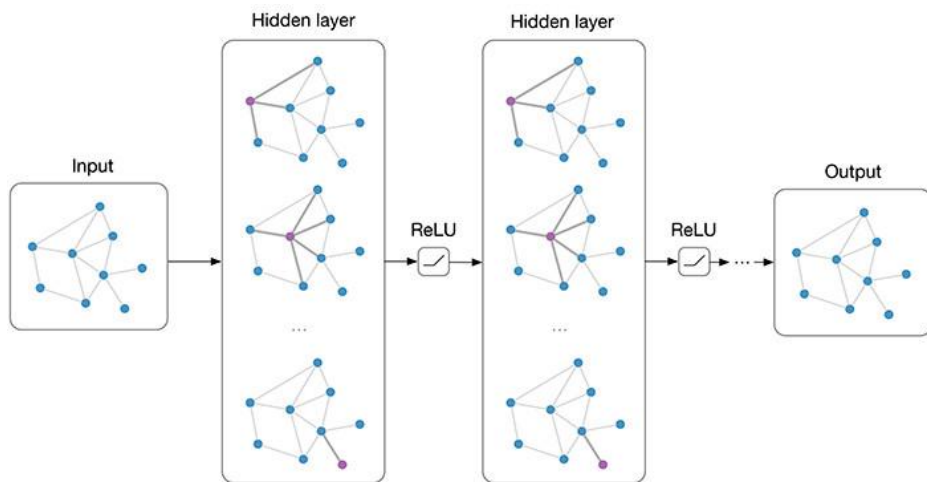
Bài toán khoá luận

- **Đầu vào:** Câu đánh giá tiếng Việt $s = \{w_1, w_2, \dots, w_n\}$ có tập các khía cạnh đã được trích xuất $a = \{a_1, a_2, \dots, a_m\}$
- **Đầu ra:** Quan điểm tương ứng với mỗi khía cạnh
$$p = \{p_1, p_2, \dots, p_m\}$$
- **Phương pháp:** Áp dụng mạng tích chập đồ thị nâng cao ngữ nghĩa và cú pháp.

2.

Phương pháp nghiên cứu

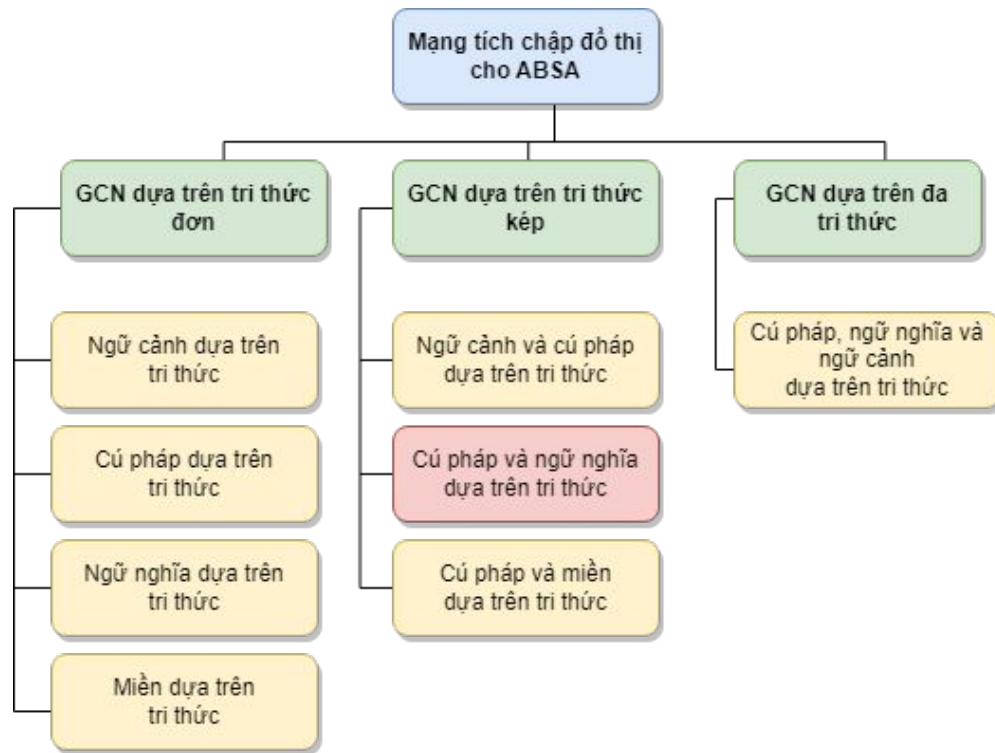
Mạng tích chập đồ thị



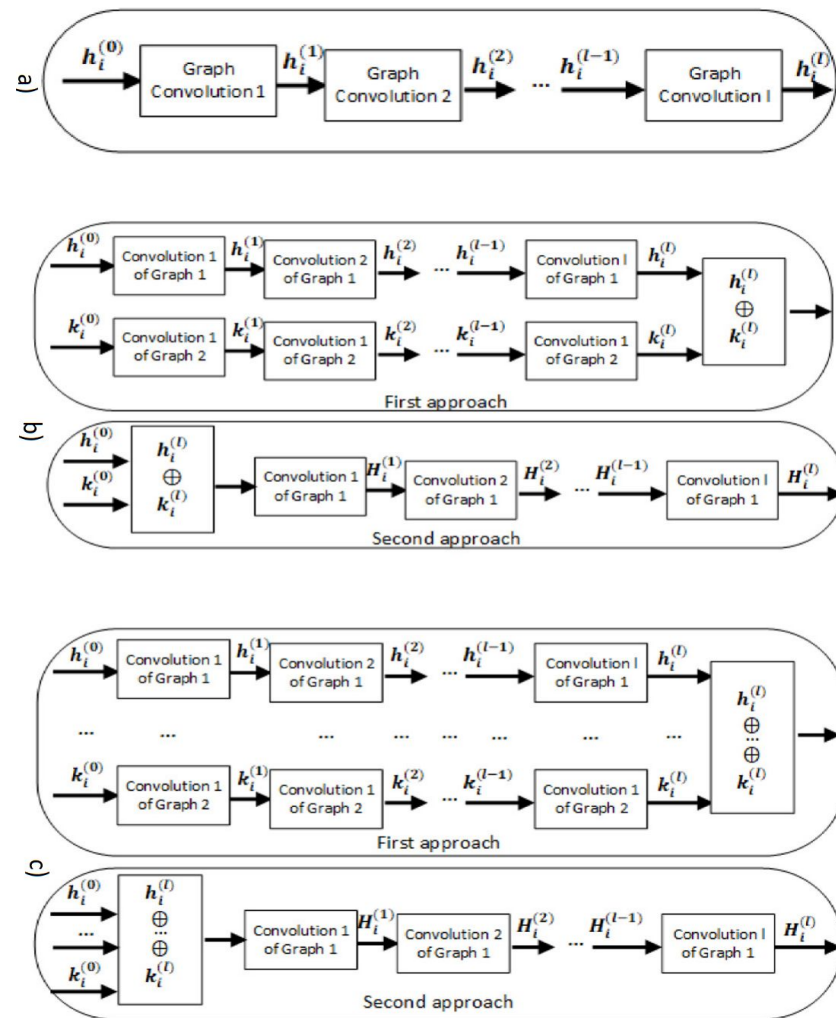
$$h_i^l = \sigma\left(\sum_{j=1}^n A_{ij} W^l h_j^{l-1} + b^l\right)$$

- Ý tưởng của GCN là tạo ra mạng nơon tích chập làm việc trên đồ thị.
- Mô hình hoạt động bằng cách truyền thông tin giữa các nút lân cận trong đồ thị.
- Mô hình cập nhật lặp lại biểu diễn nút dựa trên nút lân cận.

Mạng tích chập đồ thị cho ABSA



[6] H. T. Phan, N. T. Nguyen, and D. Hwang, "Aspect-level sentiment analysis: A survey of graph convolutional network methods," *Information Fusion*, vol. 91, pp.149–172, 2023



Ý tưởng khoá luận

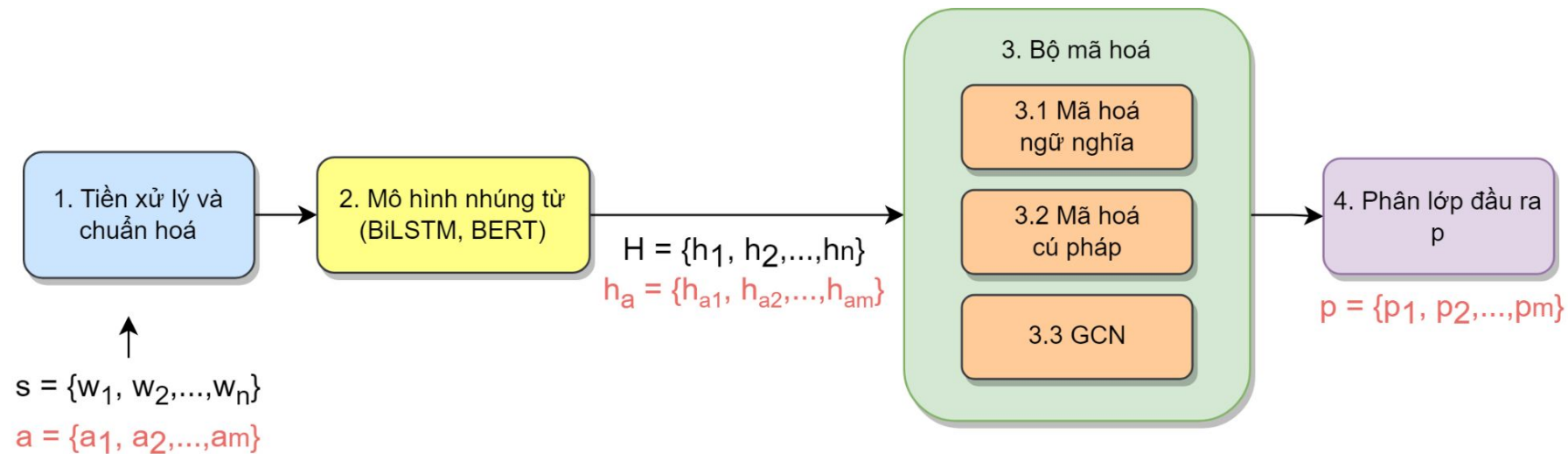
- Khóa luận ứng dụng mạng tích chập đồ thị dựa trên tri thức kép dựa trên ý tưởng mô hình SSEGCN [13].
- Mô hình kết hợp tri thức ngữ nghĩa và cú pháp.
- Đánh giá bộ phân lớp của mô hình trên bộ dữ liệu tiếng Việt.

[13] Z. Zhang, Z. Zhou, and Y. Wang, "SSEGCN: Syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies Association for Computational Linguistics, 2022*, pp. 4916–4925

3.

Mô hình khoá luận

Kiến trúc mô hình



3.1 Mã hoá ngữ nghĩa

- Mô hình sử dụng **p** ma trận điểm chú ý tương ứng với tự chú ý **p** đầu. Đối với đầu chú ý thứ **i**, ta có:

Chú ý nhận thức khía cạnh

$$A_{asp}^i = \tanh(H_a W^a \times (KW^k)^T + b) \quad (3.1.1)$$

Tự chú ý

$$A_{self}^i = \frac{QW^Q \times (KW^K)^T}{\sqrt{d}} \quad (3.1.2)$$

$$A^i = A_{self}^i + A_{asp}^i \quad (3.1.3)$$

3.2 Mã hoá cú pháp

- Ma trận đường đi ngắn nhất giữa hai nút trong đồ thị:

$$D(i, j) = \min d(v_i, v_j) \quad (3.2.1)$$

- Xây dựng p ma trận mặt nạ cú pháp với ngưỡng k :

$$M_{ij}^k = \begin{cases} 0, D(i, j) \leq k, k \in [1, p] \\ -\infty, otherwise \end{cases} \quad (3.2.2)$$

$$M = M^1, \dots, M^k, \dots, M^p \quad (3.2.3)$$

$$A_{mask}^i = softmax(A^i + M^i) \quad (3.2.4)$$

3.3 Lớp GCN

- Công thức cập nhật nút:

$$h_i^l = \sigma(\sum_{j=1}^n A_{ij} W^l h_j^{l-1} + b^l) \quad (3.3.1)$$

- Đầu ra tại lớp GCN thứ l:

$$H^l = \{h_1^l, h_2^l, \dots, h_n^l\} \quad (3.3.2)$$

- Che giấu các từ không phải khóa cạnh ở đầu ra và áp dụng average pooling cho biểu diễn khóa cạnh h_a^l

$$h_a^l = f(h_{a_1}^1, h_{a_2}^1, \dots, h_{a_m}^1) \quad (3.3.3)$$

- Phân phối xác suất cuối cùng:

$$p(a) = \text{softmax}(W_p h_a^l + b_p) \quad (3.3.4)$$

4.

Thực nghiệm và kết quả

Dữ liệu và kịch bản thực nghiệm

- **Dữ liệu thực nghiệm:**

UIT-ViSFD: bộ dữ liệu tiếng Việt cho phân tích quan điểm mức khía cạnh, bao gồm các đánh giá về điện thoại di động, chưa có nhãn vị trí.

UIT-ViSD4SA: bộ dữ liệu tương tự UIT-ViSFD, đã có sẵn nhãn vị trí.

- **Kịch bản thực nghiệm:**

Đánh giá hiệu quả của bộ phân lớp khía cạnh SSEGCGN trên 2 tập dữ liệu.

Thống kê dữ liệu

- Thống kê trên bộ dữ liệu UIT-ViSFD:

Train	Dev	Test
7786	1112	2224

- Thống kê trên bộ dữ liệu UIT-ViSD4SA:

Train	Dev	Test
7784	1113	2225

Độ đo đánh giá

- Sử dụng 3 độ đo: Precision, Recall và F1

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$F1 = \frac{2 \times P \times R}{P+R}$$

Tham số mô hình

- Mô hình được huấn luyện trên CPU với các siêu tham số:

Hàm tối ưu hoá: Adam

Trình khởi tạo tham số: Xavier initializer

Tốc độ học: 0.002

Bộ điều chỉnh: L2

Alpha, beta: 0.25, 0.25

Số lượng epoch: 20

Kích thước batch: 16

Số đầu tự chú ý: 5

Số lớp GCN: 2

Kết quả thực nghiệm so với baseline

- Sử dụng bộ dữ liệu UIT-ViSFD

	Precision	Recall	F1 (macro-avg)
Naive Bayes	0.4907	0.3043	0.3756
SVM	0.1609	0.2335	0.1969
Random Forest	0.1711	0.2456	0.2017
CNN	0.3334	0.2292	0.2716
LSTM	0.5651	0.4839	0.5213
BiLSTM	0.6582	0.6053	0.6306
Mô hình khoá luận	0.8201	0.5712	0.6076
			0.7663 (weighted-avg)

Kết quả thực nghiệm so với baseline

- Sử dụng bộ dữ liệu UIT-ViSD4SA

	Precision	Recall	F1 (macro-avg)
BiLSTM-CRF	0.4684	0.4546	0.4570
Mô hình khoá luận	0.7943	0.5409	0.5670

Kết quả thực nghiệm

- Kết quả chi tiết trên bộ dữ liệu UIT-ViSFD và UIT-ViSD4SA

	Precision	Recall	F1	support
positive	0.7654	0.9714	0.8562	4192
negative	0.8766	0.5832	0.7004	2279
neutral	0.8182	0.1589	0.2662	453
accuracy			0.7904	6924
macro avg	0.8201	0.5712	0.6076	6924
weighted avg	0.8055	0.7904	0.7663	6924

	Precision	Recall	F1	support
positive	0.7610	0.9684	0.8522	4333
negative	0.8672	0.5592	0.6799	2289
neutral	0.7547	0.0950	0.1688	421
accuracy			0.7832	7043
macro avg	0.7943	0.5409	0.5670	7043
weighted avg	0.7951	0.7832	0.7554	7043

5. Kết luận

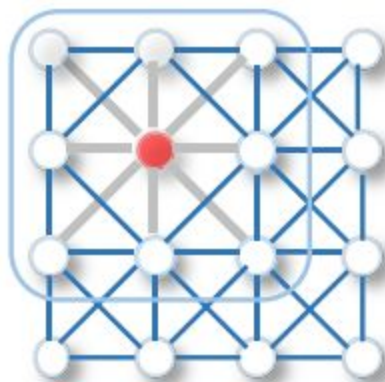
Kết quả:

- Tiếp cận và khai thác phương pháp sử dụng mạng tích chập đồ thị cho phân tích quan điểm mức khía cạnh.
- Áp dụng mô hình cho bộ dữ liệu tiếng Việt.

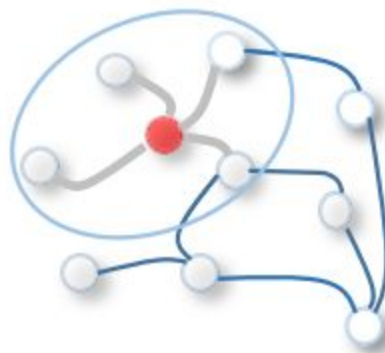
Hướng nghiên cứu tiếp theo:

- Cải thiện hiệu suất mô hình bằng cách cải thiện nhãn.
- Thực nghiệm thêm với một số mô hình dựa trên mạng tích chập đồ thị.

Cảm ơn quý thầy cô và các bạn
đã lắng nghe!



(a)



(b)

Fig. 1. 2-D convolution versus graph convolution. (a) 2-D convolution: analogous to a graph, each pixel in an image is taken as a node where neighbors are determined by the filter size. The 2-D convolution takes the weighted average of pixel values of the red node along with its neighbors. The neighbors of a node are ordered and have a fixed size. (b) Graph convolution: to get a hidden representation of the red node, one simple solution of the graph convolutional operation is to take the average value of the node features of the red node along with its neighbors. Different from the image data, the neighbors of a node are unordered and variable in size.