

## Slide 20

- Kết thúc phần 4 từ các công việc liên quan đến **dữ liệu, nghiên cứu mô hình** thì nhóm em thu được kết quả thực nghiệm như sau.
- Đây là **một trong các biểu đồ** khi huấn luyện, đọc **nhận xét**

## Slide 21

- Để đánh giá độ chính xác của mô hình thì nhóm em sử dụng **độ đo F1-Score, kết quả được thể hiện như trên.**
- Chú thích:
  - **Kiến trúc:** 4 Kiến trúc được thử nghiệm dựa trên việc **học thông tin văn bản (Text Embedding)** và **học cấu trúc văn bản (Graph Embedding)**
  - **weight loss:** các bộ tham số tối ưu hàm loss. [1, 5, 5], [1, 9, 9], [1, 19, 19]: Là trọng số tương ứng với [O,B,I] . (In, Begin, Out of Keyword) . **Lý do: Nhãn O chiếm số lượng áp đảo, nên cần giảm xuống bằng cách thêm trọng số như trên.**
  - **d:** tổng chiều dài vector embedding của Graph Embedding gồm WV và FT . (BERT cố định là 768)
  - **th = threshold (ngưỡng)** . Trong abstrack + title sẽ loại bỏ đi các từ có tần suất xuất hiện nhỏ hơn ngưỡng. Sử dụng ngưỡng threshold, **giúp giảm độ nhiễu và tăng chất lượng của mô hình học.**
  - **walks = len\_random\_walks** . **Số lượng đường đi ngẫu nhiên trên đồ thị graph để thu thập về thông tin cấu trúc của nó.** Mỗi random walks sẽ tạo ra một **chuỗi các node** . (Sử dụng ở bước 4)
  - Lưu ý là BERT không chỉ là **học thông tin văn bản** mà nó cũng đã là học **ngữ cảnh của đoạn văn bản.**

Kiến trúc tổng	Weight của loss	kq (d=200+th=3+ walks=1e4)	kq (d=400+th=3+ walks = 1e4)	kq (d=600+th=3+ walks=1e4)	kq (d=600+th=3+ walks = 1e5)	kq (d=600+th=2+ walks = 1e5)
BERT	[1, 9, 9]	0.3021	x	x	x	x
	[1, 5, 5]	<b>0.3051</b>	x	x	x	x
	[1, 19, 19]	0.2549	x	x	x	x
BERT + WV	[1, 9, 9]	0.2947	0.3224	0.3274	0.3203	0.3290
	[1, 5, 5]	0.2959	0.3185	<b>0.3308</b>	0.3237	0.3251
	[1, 19, 19]	0.2801	0.3163	0.3175	0.3151	0.3223
BERT + FT	[1, 9, 9]	0.2855	0.3239	0.3296	0.3232	0.3291
	[1, 5, 5]	0.2905	0.3168	<b>0.3325</b>	0.3267	0.3256
	[1, 19, 19]	0.2760	0.3021	0.3164	0.3167	0.3220
BERT + WVFT	[1, 9, 9]	0.2922	0.3289	0.3308	0.3273	0.3324
	[1, 5, 5]	0.2975	0.3205	<b>0.3341</b>	0.3297	0.3324
	[1, 19, 19]	0.2778	0.3082	0.3192	0.3221	0.3275

- Từ bảng kết quả ta có thể rút ra: **Nhận xét & Giải thích như sau:**
  1. Kết quả cao nhất trên **thang đo F1-score** là 0.3341 với kiến trúc sử dụng gồm BERT Transformer kết hợp với Graph Embedding cả Word2Vec và FastText có tổng số chiều là 600, ngưỡng chọn từ phổ biến là 3 và số lần random walks là 10000, đồng thời có bộ trọng số của loss là [1, 5, 5].  
⇒ Cho thấy đây là bộ tham số tối ưu tốt nhất hiện tại.
  2. Nếu chỉ sử dụng BERT không sử dụng graph embedding thì độ chính xác 0.3051 thấp hơn 0.0290 so

với kết quả cao nhất khi sử dụng graph embedding.

⇒ **Từ đó cho thấy:** Hiệu quả mà graph embedding mang lại **đã cải thiện độ chính xác đáng kể**, thay **vì chỉ sử dụng ngữ cảnh của từ trong đoạn hiện tại**, ta **sẽ** sử dụng nhiều ngữ cảnh và cấu trúc khác nhau trong nhiều đoạn khác nhau trong abstract+title bằng cách sử dụng graph embedding.

3. Các kết quả trên BERT+WV và BERT+FT được cho là giá trị trung gian. Cả WV và FT nếu đứng riêng lẻ cùng kết hợp với BERT đều mang đến cải thiện tốt. Dẫn đến việc kết hợp cả 2 graph embedding này lại đạt hiệu quả cao nhất.
4. Với 3 trọng số [1, 5, 5], [1, 9, 9], [1, 19, 19] đều cho kết quả cao nhất trên bộ trọng số thấp [1, 5, 5].  
⇒ Bộ trọng số thấp [1, 5, 5] sẽ phù hợp để **thay thế cho việc** không sử dụng bộ trọng số (tức là trọng số [1, 1, 1]) còn các trọng số [1, 9, 9], [1, 19, 19] **gây chênh lệch lớn** trong việc tối ưu loss function, do đó mang đến kết quả không hiệu quả bằng [1, 5, 5].
5. Các kết quả cũng phản ánh vector\_size (độ dài của graph embedding) = 600 sẽ đạt hiệu quả cao nhất, cho thấy vector\_size **tỉ lệ thuận với F1-score** của mô hình.  
⇒ **Vector size càng lớn thì thông tin ngữ cảnh của từ trong graph embedding được biểu diễn rõ ràng** hơn, do đó độ dài của vector nên **đủ lớn để lưu thông tin ngữ cảnh đó**.  
⇒ Khi xem xét **đến các yếu tố về độ dài vector** thì 768 (độ dài của BERT) và 600 (độ dài của graph embedding (WC + FT)) **được cho là cân bằng và mang lại hiệu quả**.
6. Với tập dữ liệu hiện tại thì ngưỡng là 3 và số random walks 10000 sẽ phù hợp với bộ dữ liệu nhất. Tùy vào tập dữ liệu mà ta chọn ngưỡng và random walks cho hợp lý.

## Slide 22

- Đọc kết luận và hướng phát triển

## Slide 23

- Đọc kết luận và hướng phát triển

## Slide 24

- Tiếp đến là phần Ứng dụng, đây là phần tổng quản về hệ thống



