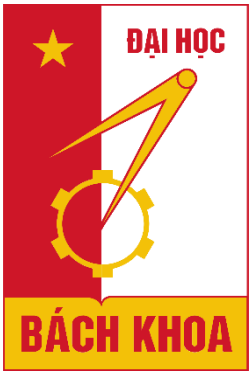
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI



ĐỒ ÁN MÔN HỌC

Hệ gợi ý

Gợi ý tin tức trên tập dữ liệu MIND

Chu Đình Đức

duc.cd194021@sis.hust.edu.vn

Nguyễn Xuân Cường

cuong.nx190040@sis.hust.edu.vn

Đỗ Minh Hiệp

hiep.dm190048@sis.hust.edu.vn

Ngành: Khoa học máy tính

Giảng viên hướng dẫn: ThS. Ngô Văn Linh

HÀ NỘI, 02/2023

MỤC LỤC

LỜI CẢM ƠN

Đồ án môn học Hệ gợi ý đã được hoàn thiện đầy đủ, chúng em xin dành lời cảm ơn chân thành tới thầy giáo phụ trách môn học ThS. Ngô Văn Linh đã có những góp ý quan trọng trong quá trình chúng em thực hiện đồ án, đồng thời đã mang đến những tiết học chất lượng trên giảng đường truyền tải những kiến thức, chia sẻ quý giá đến sinh viên.

Trong quá trình thực hiện đồ án, chúng em đã học hỏi được nhiều kiến thức quý báu, cũng như trau dồi thêm những kỹ năng quan trọng khác. Tuy nhiên đồ án môn học khó tránh khỏi những thiếu sót, chúng em rất mong nhận được nhận xét, chỉ dạy thêm từ thầy để hoàn thiện đồ án, và hoàn thiện bản thân mình trong tương lai.

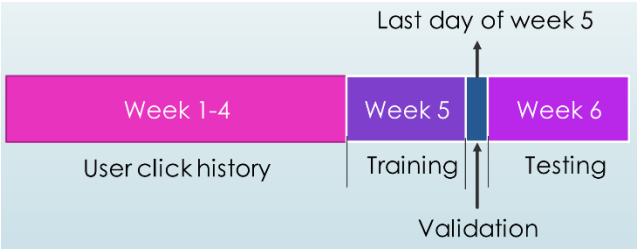
Em xin chân thành cảm ơn!

Chương I. Giới thiệu về cuộc thi và tập dữ liệu

1. Tập dữ liệu MIND

MIND (MIcrosoft News Dataset) là tập dữ liệu quy mô lớn và có chất lượng cao cho bài toán gợi ý tin tức. MIND được hình thành từ nhật ký hành vi người dùng tại trang Microsoft News. Tác giả lấy mẫu ngẫu nhiên nhật ký tương tác của 1 triệu người dùng có ít nhất 5 click trong 6 tuần từ 12/10 tới 22/11/2019.

Tác giả sử dụng dữ liệu 4 tuần đầu tiên tạo nên lịch sử click, dữ liệu tuần thứ 5 cho training và dữ liệu tuần thứ 6 cho testing trong đó dữ liệu ngày cuối cùng của tuần thứ 5 dùng cho validation.



Tập train và tập validation bao gồm 4 file khác nhau.

Tên file	Mô tả
behaviors.tsv	Lịch sử click và nhật ký impression của các user
news.tsv	Thông tin của các bản tin
entity_embedding.vec	Embedding của các thực thể trong bản tin
relation_embedding.vec	Embedding của quan hệ giữa các thực thể

Tập test cũng bao gồm 4 file như trên, tuy nhiên file behaviors.tsv của tập test sẽ không có nhãn, thông tin chi tiết được trình bày ở phía bên dưới.

1.1. behaviors.tsv

File behaviors.tsv gồm 5 cột chứa lịch sử click và nhật ký impression của các user. Ví dụ:

Cột	Nội dung
Impression ID	91
User ID	U397059
Time	11/15/2019 10:22:32 AM
History	N106403 N71977 N97080 N102132 N97212 N121652
Impressions	N129416-0 N26703-1 N120089-1 N53018-0 N89764-0

1.2. news.tsv

File news.tsv gồm 7 cột chứa thông tin chi tiết của các bài báo trong file behaviors.tsv. Ví dụ:

Cột	Nội dung
News ID	N37378

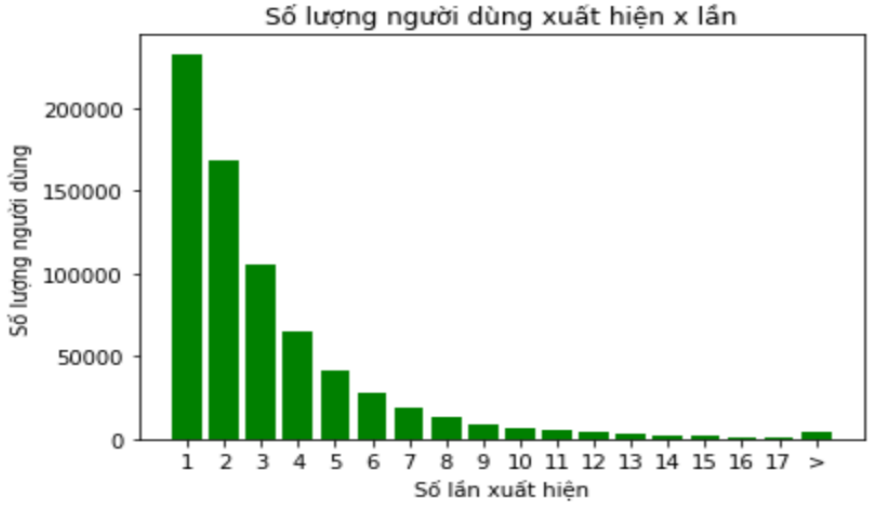
Category	sports
SubCategory	golf
Title	PGA Tour winners
Abstract	A gallery of recent winners on the PGA Tour.
URL	https://www.msn.com/en-us/sports/golf/pga-tour-winners/ss-A ...
Title Entities	[{"Label":"PGA Tour", "Type":"O", "WikidataId":"Q910409", …
Abstract Entites	[{"Label":"PGA Tour","Type": "O","WikidataId":"Q910409", …

1.3. entity_embedding.vec và relation_embedding.vec

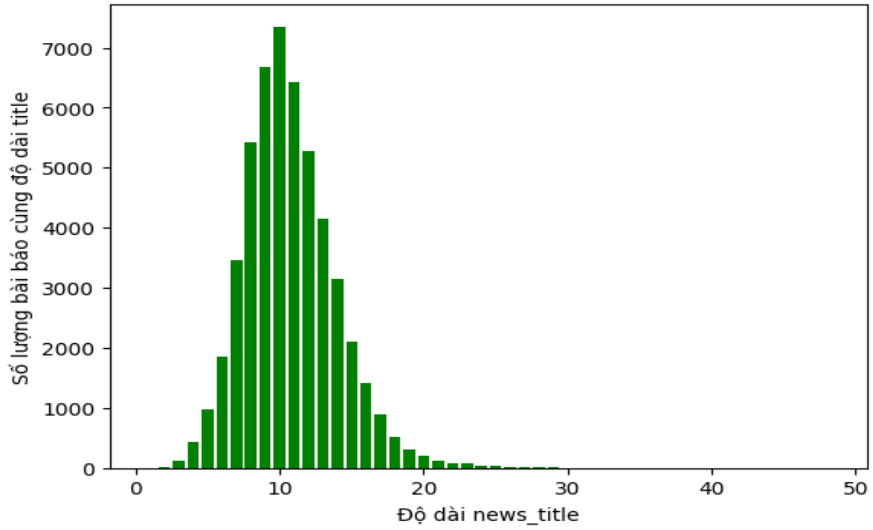
File entity_embedding.vec và relation_embedding.vec chứa các vector 100 chiều của các entity và relation được học từ subgraph (từ đồ thị tri thức WikiData) bằng phương pháp TransE. Ví dụ:

ID	Embedding Values
Q42306013	0.014516 -0.106958 0.024590 ... -0.080382

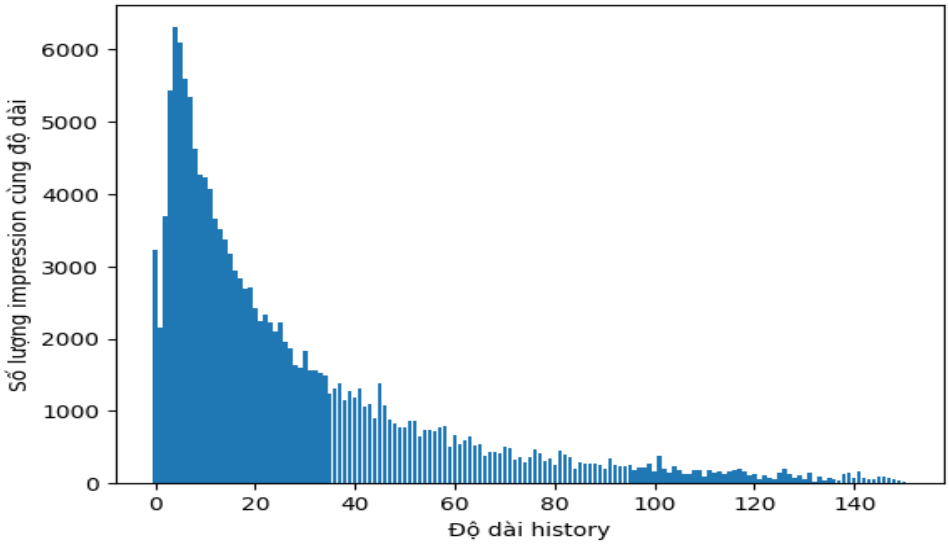
2. Thống kê dữ liệu



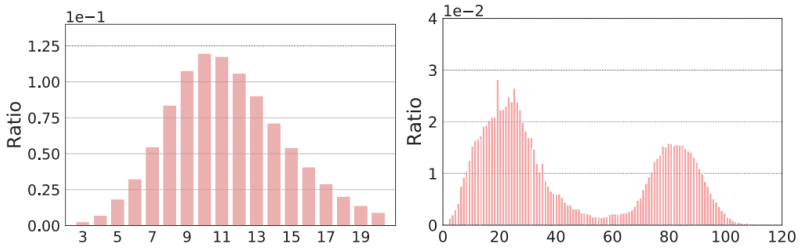
Có hơn 200000 user xuất hiện 1 lần, hơn 150000 user xuất hiện 2 lần và giảm dần, đa số các user xuất hiện từ 1 – 7 lần.



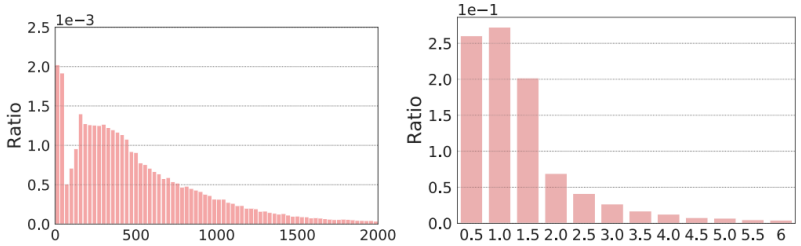
Độ dài của tiêu đề bài báo nằm trong khoảng 1 – 30, đây cũng chính là lý do mô hình NRMS mặc định chọn độ dài tiêu đề bài báo mặc định là 30.



Độ dài lịch sử click của user nằm trong khoảng 0 – 150 và chủ yếu tập trung vào khoảng 0 – 75.

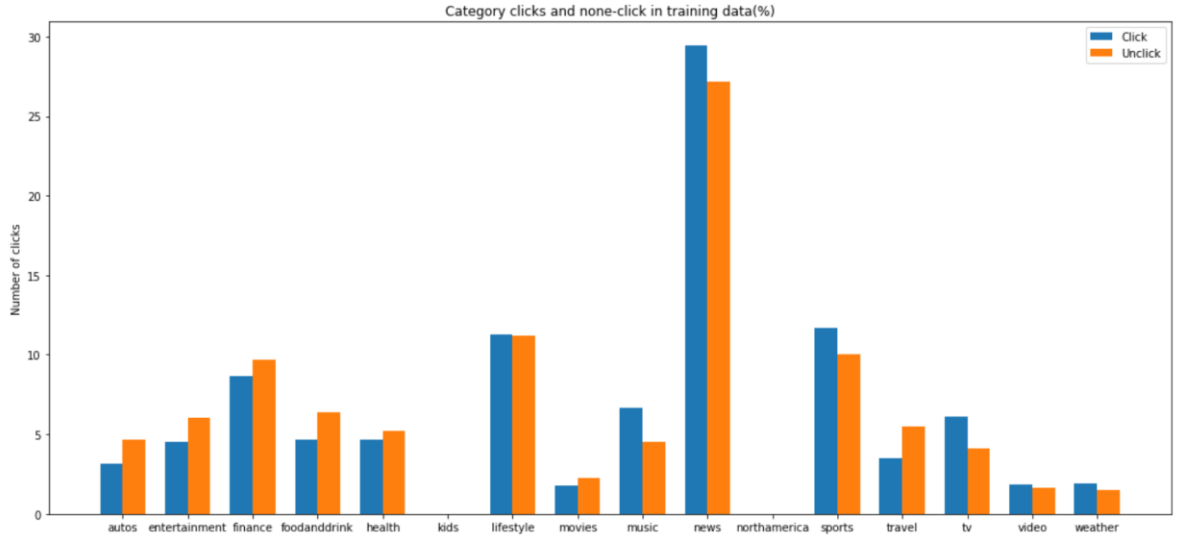


(a) Title Length (b) Abstract Length



(c) Body Length (d) Survival Time

Một số thống kê cơ bản khác trên tập MIND. Cuối cùng là biểu đồ số lượng click và non-click theo category. Lượng non-click chiếm ưu thế, tuy nhiên user có xu hướng click vào category như: news, sports, lifestyle, …



3. Cuộc thi gợi ý tin tức trên tập MIND

Thí sinh tham dự cuộc thi tại địa chỉ [MIND News Recommendation Competition]. Cách thức tham gia cuộc thi:

- Đọc thông tin chi tiết trên website
- Tham gia cuộc thi trên Codalab
- Huấn luyện và đánh giá mô hình gợi ý tin tức của bạn trên tập dữ liệu MIND
- Nộp kết quả dự đoán của bạn trên tập kiểm tra tới Codalab để có được kết quả chính thức

Ví dụ 24481 [4,1,3,2] là kết quả dự đoán của impression dưới đây

ImpressionID	Candidate News
24481	N125045 N87192 N73556 N20417

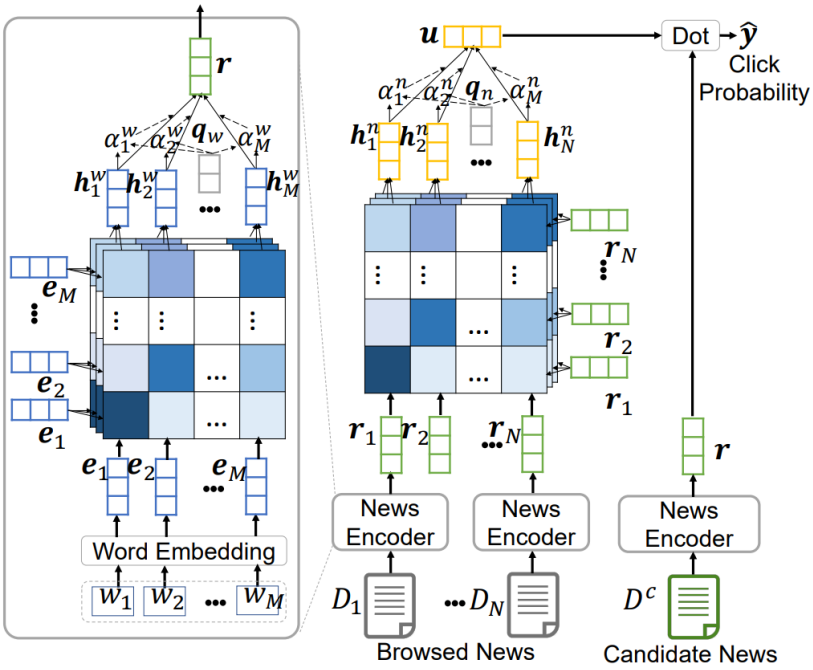
Quá trình submission:

- Điều hướng tới ‘Participate’
- Viết tóm tắt về mô hình của bạn (tùy chọn)
- Click vào nút ‘Submit / View Results’
- Upload file nén kết quả
- Chờ cho tới khi trạng thái chuyển sang ‘Finished’ hoặc ‘Failed’

Chương II. Mô hình sử dụng

1. Mô hình NRMS

NRMS (Neural News Recommendation with Multi-Head Self-Attention) gồm 4 phần chính: News Encoder, User Encoder, Click Predictor, Negative Log-Likelihood.



1.1. News Encoder

News Encoder gồm 3 lớp được sử dụng để học ra biểu diễn của bài báo từ tiêu đề bài báo. Lớp đầu tiên là lớp word embedding có tác dụng tách tiêu đề bài báo thành các token sau đó chuyển chuỗi các token này thành chuỗi các vector. Giả sử tiêu đề bài báo gồm M từ $[w_1, w_2, \dots, w_M]$ sau khi đi qua lớp này sẽ trở thành M vector $[e_1, e_2, \dots, e_M]$.

Lớp tiếp theo là lớp multi-head self-attention học ra biểu diễn ngữ cảnh của từ thông qua sự tương tác giữa các từ trong tiêu đề. Biểu diễn của từ thứ i được học bởi attention head thứ k có công thức như sau:

$$\alpha_{i,j}^k = \frac{\exp(\mathbf{e}_i^T \mathbf{Q}_k^w \mathbf{e}_j)}{\sum_{m=1}^{M} \exp(\mathbf{e}_i^T \mathbf{Q}_k^w \mathbf{e}_m)},$$

$$\mathbf{h}_{i,k}^w = \mathbf{V}_k^w \left(\sum_{j=1}^{M} \alpha_{i,j}^k \mathbf{e}_j\right),$$

trong đó Q_k^w và V_k^w là tham số của self-attention head thứ k, và $\alpha_{i,j}^k$ là độ quan trọng của tương tác giữa từ thứ i và từ thứ j. Kết quả cuối cùng được tổng hợp bằng các nối liên tiếp kết quả của h self-attention head:

$$\mathbf{h}_i^w = [\mathbf{h}_{i,1}^w; \mathbf{h}_{i,2}^w; \dots; \mathbf{h}_{i,h}^w]$$

Lớp cuối cùng là lớp attention bổ sung vì những từ khác nhau sẽ có mức độ quan trọng khác nhau đối với biểu diễn tiêu đề bài báo. Trọng số attention α_i^w của từ thứ i được tính toán như sau:

$$a_i^w = \mathbf{q}_w^T \tanh(\mathbf{V}_w \times \mathbf{h}_i^w + \mathbf{v}_w),$$

$$\alpha_i^w = \frac{\exp(a_i^w)}{\sum_{j=1}^{M} \exp(a_j^w)},$$

trong đó V_w và v_w là tham số, q_w là vector query.

Biểu diễn cuối cùng của tiêu đề bài báo là tổng hợp có trọng số của các token:

$$\mathbf{r} = \sum_{i=1}^{M} \alpha_i^w \mathbf{h}_i^w$$

1.2. User Encoder

User Encoder gồm 2 lớp được dùng để học ra biểu diễn của user từ những bài báo mà user đã click. Lớp đầu tiên là lớp multi-head attention. Những bài báo được user click trong quá khứ thường có liên quan với nhau. Thêm vào đó một bài báo có thể tương tác với nhiều bài báo được click bởi cùng một user. Biểu diễn của bài báo thứ i được học bởi attention head thứ k được tính toán như sau:

$$\beta_{i,j}^k = \frac{\exp(\mathbf{r}_i^T \mathbf{Q}_k^n \mathbf{r}_j)}{\sum_{m=1}^{M} \exp(\mathbf{r}_i^T \mathbf{Q}_k^n \mathbf{r}_m)},$$

$$\mathbf{h}_{i,k}^n = \mathbf{V}_k^n (\sum_{j=1}^{M} \beta_{i,j}^k \mathbf{r}_j),$$

trong đó Q_k^n và V_k^n là tham số của self-attention head thứ k, $\beta_{i,j}^k$ là độ quan trọng của tương tác giữa bài báo thứ i và bài báo thứ j. Biểu diễn của bài báo i sau khi thực hiện h self-attention head thu được bằng phép nối như sau:

$$\mathbf{h}_i^n = [\mathbf{h}_{i,1}^n; \mathbf{h}_{i,2}^n; ...; \mathbf{h}_{i,h}^n]$$

Lớp tiếp theo là lớp attention bổ sung vì bài báo khác nhau sẽ có ảnh hưởng khác nhau đến biểu diễn user. Trọng số attention của bài báo thứ i được tính toán như sau:

$$a_i^n = \mathbf{q}_n^T \tanh(\mathbf{V}_n \times \mathbf{h}_i^n + \mathbf{v}_n),$$

$$\alpha_i^n = \frac{\exp(a_i^n)}{\sum_{j=1}^{N} \exp(a_j^n)},$$

trong đó V_n, v_n và q_n là tham số, và N là số lượng bài báo được click.

Biểu diễn cuối cùng của user là phép tổng hợp có trọng số của biểu diễn các bài báo được lick bởi user:

$$\mathbf{u} = \sum_{i=1}^{N} \alpha_i^n \mathbf{h}_i^n$$

1.3. Click Predictor

Module này tính xác suất click bài báo của user.

$$\hat{y} = \mathbf{u}^T \mathbf{r}^c$$

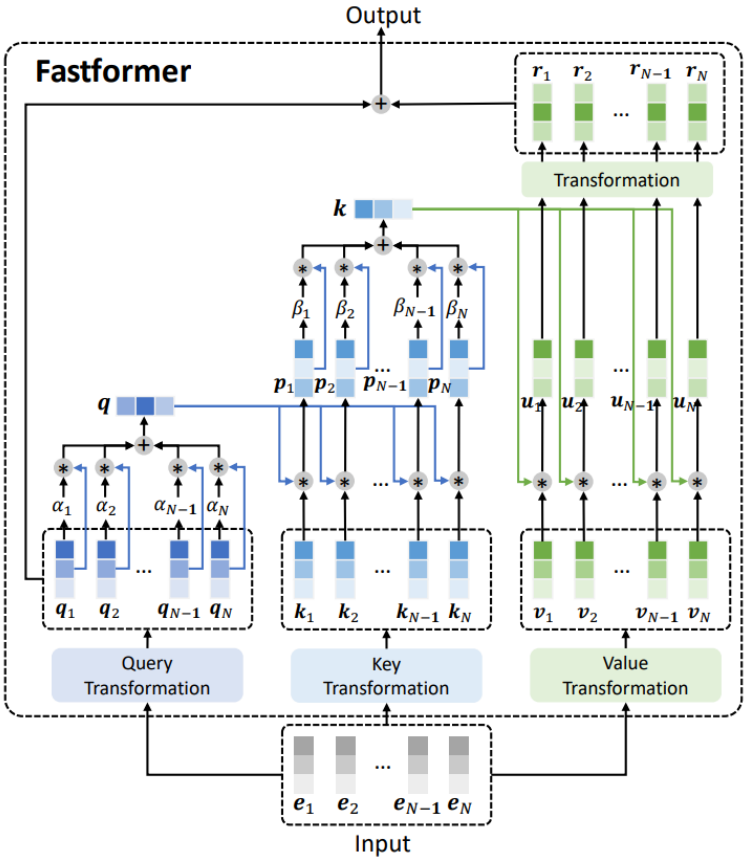
1..4. Negative Log-Likehood

Kỹ thuật lấy mẫu âm tính được sử dụng cho quá trình huấn luyện. Cụ thể, đối với mỗi bài báo được user click (mẫu dương), chúng ta sẽ lựa chọn ra K bài báo user không click (mẫu âm) với xác suất click là $\hat{y}^+$ và $[\hat{y}_1^-, \hat{y}_2^-, \ldots, \hat{y}_K^-]$

$$p_i = \frac{\exp(\hat{y}_i^+)}{\exp(\hat{y}_i^+) + \sum_{j=1}^{K} \exp(\hat{y}_{i,j}^-)}$$

$$\mathcal{L} = -\sum_{i \in \mathcal{S}} \log(p_i)$$

2. Mô hình Fastformer + PLM - NR



Kiến trúc của mô hình Fastformer được mô tả như hình phía trên. Đầu tiên nó sử dụng attention bổ sung để tóm tắt chuỗi query thành query toàn cục, sau đó mô hình hóa tương tác giữa vector query toàn cục với các key thông qua phép element-wise product và tổng hợp các key thành một key toàn cục thông qua attention bổ sung, tiếp theo mô hình hóa tương tác giữa key toàn cục với các value thông qua element-wise product và sử dụng 1 lớp linear để học ra các value toàn cục chứa tri thức ngữ cảnh và cuối cùng là cộng các value này tương ứng với các query ban đầu để tạo nên đầu ra cuối cùng. Thực hiện theo cách này độ phức tạp tính toán là tuyến tính đối với chiều dài chuỗi đầu vào. Ký hiệu ma trận đầu vào là E.

$$E \in R^{N \times d} \xrightarrow{\text{linear transformation}} Q, K, V \in R^{N \times d}$$

2.1. Vector query toàn cục

Trọng số α_i của vector query thứ i được tính toán như sau:

$$\alpha_i = \frac{\exp(w_q^T q_i / \sqrt{d})}{\sum_{j=1}^{N} \exp(w_q^T q_j / \sqrt{d})}$$

trong đó vector w_q là tham số. Và vector query toàn cục được tính toán như sau:

$$q = \sum_{i=1}^{N} \alpha_i q_i$$

2.2. Vector key toàn cục

Chúng ta ký hiệu vector thứ i trong ma trận tri thức ngữ cảnh toàn cục là p_i, vector này được tính toán theo công thức:

$$p_i = q * k_i$$

trong đó * thể hiện phép toán element-wise product. Trọng số attention β_i của vector key thứ i được tính toán như sau:

$$\beta_i = \frac{\exp(w_k^T p_i / \sqrt{d})}{\sum_{j=1}^{N} \exp(w_k^T p_j / \sqrt{d})}$$

trong đó w_k là tham số. Và vector key toàn cục được tính bằng cách:

$$k = \sum_{i=1}^{N} \beta_i p_i$$

2.3. Ma trận value tri thức ngữ cảnh toàn cục

Ký hiệu vector thứ i trong ma trận tương tác key - value là u_i:

$$u_i = k * v_i$$

Sau khi thu được ma trận U, chúng ta áp dụng 1 lớp linear lên ma trận U để học được biểu diễn của ma trận. Ký hiệu ma trận đầu ra là R:

$$R = [r_1, r_2, ..., r_N] \in R^{N \times d}.$$

$$U \in R^{N \times d} \xrightarrow{\text{linear transformation}} R \in R^{N \times d}$$

2.4. Output và đánh giá

Ma trận R được cộng cùng với ma trạn query ban đầu để hình thành đầu ra của mô hình Fastformer:

$$R \in R^{N \times d} + Q \in R^{N \times d} \rightarrow O \in R^{N \times d}$$

Bằng cách xếp nhiều lớp Fastformer lên nhau, chúng ta có thể hoàn toàn học được thông tin ngữ cảnh của câu một cách đầy đủ. Sử dụng kỹ thuật chia sẻ tham số, Fastformer chia sẻ các tham số value và query transformation. Thêm vào đó tham số giữa các lớp Fastformer cũng được chia sẻ đẻ giảm số lượng tham số và overfitting. Độ phức tạp tính toán và tổng số tham số của mô hình là $O(N.d)$ và $3hd^2 + 2hd$.

2.5. Fastformer + PLM – NR

Trong mô hình gợi ý tin tức phía dưới, thay module attention bằng module fastformer ta thu được mô hình Fastformer + PLM -NR.

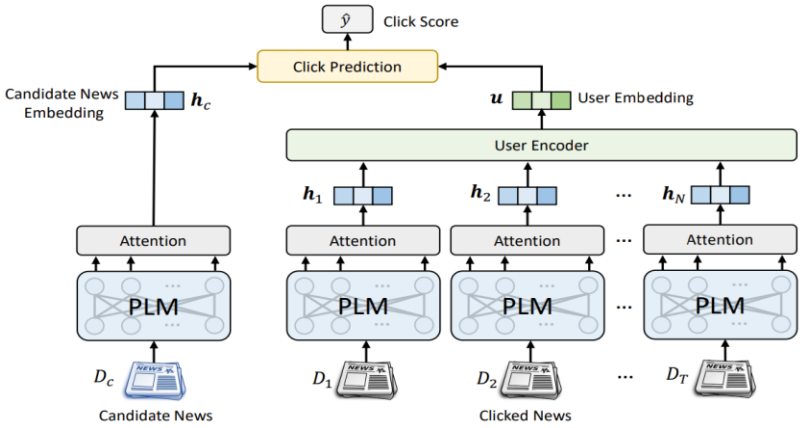
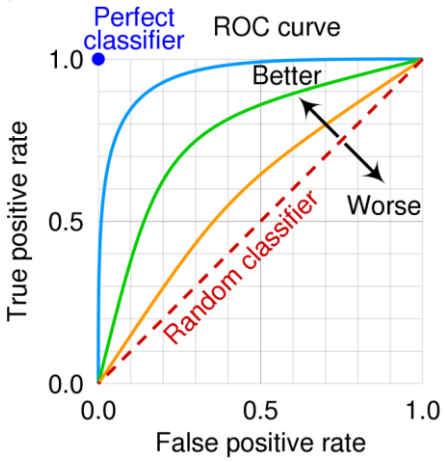


Figure 2: The framework of PLM empowered news recommendation.

Chương III. Độ đo đánh giá

1. AUC (Area Under Curve)



ROC (Receiving Operating Curve) là đường cong đồng biến biểu diễn khả năng phân loại của model tại các ngưỡng khác nhau dựa trên 2 chỉ số:

- TPR (True Positive Rate): recall
- FPR (False Positive Rate): tỷ lệ dự báo sai các trường hợp thực tế là negative thành positive trên tổng số các trường hợp thực tế là negative

$$FPR = \frac{FP}{total\_negative}$$

AUC (Area Under Curve) là phần diện tích phần nằm dưới ROC có giá trị trong đoạn [0, 1]. AUC càng lớn khi ROC càng tiệm cận đường y = 1 và khả năng phân loại của mô hình càng tốt. Khi ROC nằm sát đường chéo thì model là model phân loại ngẫu nhiên.

2. MRR (Mean Reciprocal Rank)

MRR là một trong những metrics đơn giản nhất trong việc đánh giá các ranking models. MRR tính trung bình của các thức hạng tương ứng của mục liên quan đầu tiên đối với tập các truy vấn Q, có thể định nghĩa nó như sau:

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

Dưới đây là một ví dụ về các tính độ đo MRR.

Query	Proposed Results	Correct response	Rank	Reciprocal rank
cat	catten, cati, **cats**	cats	3	1/3
torus	torii, **tori**, toruses	tori	2	1/2
virus	**viruses**, virii, viri	viruses	1	1

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} = \frac{1}{3}(\frac{1}{3}+\frac{1}{2}+\frac{1}{1}) = \frac{11}{18} \approx 0.61$$

3. nDCG (Normalized Discounted Cumulative Gain)

DCG là độ đo chất lượng xếp hạng. Nó thường được sử dụng trong các bài toán truy xuất thông tin như đo độ hiệu quả của thuật toán tìm kiếm bằng cách xếp hạng các bài viết mà nó hiển thị theo mức độ liên quan với từ khóa tìm kiếm. Cụ thể, hãy xét p = 5 bài viết output cùng mức độ liên quan với từ khóa tìm kiếm:

$$(D_1, 3), (D_2, 2), (D_3, 0), (D_4, 0), (D_5, 1)$$

$$CG_p = \sum_{i=1}^{p} rel_i = 3 + 2 + 0 + 0 + 1 = 6$$

$$DCG_p = \sum_{i=1}^{p}\frac{rel_i}{\log_2(i+1)} = \frac{3}{\log_2 2} + \frac{2}{\log_2 3} + \frac{0}{\log_2 4} + \frac{0}{\log_2 5} + \frac{1}{\log_2 6} \approx 4.67$$

$$IDCG_p = \sum_{i=1}^{p}\frac{rel_i\_order}{\log_2(i+1)} = \frac{3}{\log_2 2} + \frac{2}{\log_2 3} + \frac{1}{\log_2 4} + \frac{0}{\log_2 5} + \frac{0}{\log_2 6} \approx 4.76$$

$$nDCG@p = \frac{DCG_p}{IDCG_p} = \frac{4.67}{4.76} \approx 0.98$$

Chương IV. Thực nghiệm và kết quả

1. Nội dung thực nghiệm

Đối với NRMS, chúng em tiến hành một số điều chỉnh mô hình nhằm mục đích so sánh kết quả giữa các phiên bản khác nhau. Đầu tiên là việc thay thế pretrained word embedding GLoVe bằng pretrained BERT do BERT được huấn luyện trên bộ dữ liệu lớn và kiến trúc encoder của mô hình transformer có khả năng học ra những word embedding chất lượng, chứa nhiều thông tin về ngữ cảnh hơn GLoVe.

Tiếp theo chúng em tiến hành điều chỉnh độ dài tiêu đề bài báo và độ dài lịch sử click của user để tìm ra độ dài tối ưu của tiêu đề và lịch sử click mà ở độ dài đó độ chính xác đạt được là cao nhất.

Bên cạnh đó là việc sử dụng thêm phần văn bản tóm tắt (abstract) của bài báo để học ra biểu diễn chính xác hơn của bài báo, từ đó học ra được biểu diễn chính xác hơn của user.

Ngoài ra nhóm chúng em thực hiện việc trộn hai tập dữ liệu train và test thành một tập dữ liệu mới lớn hơn. Việc huấn luyện mô hình trên tập dữ liệu mới lớn hơn sẽ thu được mô hình có tính tổng quát và độ chính xác cao khi đem đi dự đoán trên tập dữ liệu mới.

Cuối cùng, việc thay thế các head self-attention của mô hình transformer gốc bằng các head fastformer khiến cho quá trình huấn luyện mô hình trở nên nhanh chóng hơn và tốn ít tài nguyên bộ nhớ hơn. Hơn thế nữa, trong bài báo fastformer gốc tác giả đã chứng minh hiệu hiệu quả của mô hình fastformer cao hơn transformer trong nhiệm vụ gợi ý tin tức, do đó có thể hi vọng rằng mô fastformer sẽ giúp mô hình học biểu diễn của news và user hiệu quả hơn.

Đối với mô hình fastformer cho bài toán gợi ý tin tức, chúng em sử dụng mô hình fastformer và pretrained language model UniLM (Fastformer + PLM – NR).

Các so sánh cụ thể được trình bày phía bên dưới.

2. Thiết lập mô hình

Tham số chung:

Tham số chung					
epoch	batch_size	learning_rate	loss	optimizer	npratio
1	32	0.0001	CE	adam	4

Thiết lập thực nghiệm:

Thiết lập thử nghiệm							
STT	Model	Word Embedding	Title Size	History Size	Mix Data	Abstract	Fastformer
1	NRMS v1 (base)	GLoVe	30	50	N	N	N
2	NRMS v2	GLoVe	30	75	N	N	N
3	NRMS v3	GLoVe	45	50	N	N	N
4	NRMS v4	GLoVe	30	75	Y	45	N
5	NRMS v5	BERT	30	75	Y	45	N
6	NRMS v6	BERT	30	75	Y	45	Y

3. Kết quả thực nghiệm

Kết quả thử nghiệm trên tập MINDsmall (MINDsmall_validation)					
STT	Model	AUC	MRR	nDCG@5	nDCG@10
1	NRMS v1 (base)	0.6383	0.2941	0.3202	0.3878
2	NRMS v2	0.6431	0.2985	0.327	0.3929
3	NRMS v3	0.6387	0.294	0.3221	0.3885
4	NRMS v4	0.6511	0.3073	0.3372	0.4016
5	NRMS v5	0.6651	0.3215	0.3486	0.4057
6	NRMS v6	0.6658	0.3224	0.3443	0.4063

4. Kết quả submission

Kết quả submission trên tập MINDlarge (MINDlarge_test)						
STT	Model	AUC	MRR	nDCG@5	nDCG@10	Rank (5/2/2023)
1	Fastformer + PLM-NR	**0.7201**	**0.3688**	**0.4059**	**0.4617**	**5**
2	NRMS v1 (base)	0.6844	0.3363	0.367	0.4238	27

Chương V. Kết luận và hướng phát triển

1. Kết luận

Trong đồ án môn học lần này chúng em đã tìm hiểu và thử nghiệm nhiều biến thể của mô hình NRMS và mô hình Fastformer + PLM – NR gốc. Kết luận ban đầu, độ dài tiêu đề tối ưu của bài báo là 30 và độ dài lịch sử click tối ưu là 75 do thống kê cho thấy rất ít bài báo có tiêu đề dài hơn 30 và độ dài lịch sử click có thể kéo dài tới hơn 100, tăng độ dài của lịch sử click độ chính xác của mô hình cũng tăng theo. Việc trộn dữ liệu train với validation, thay thế pretrained GLoVe bằng BERT và áp dụng fastformer cũng làm tăng đáng kể độ chính xác của mô hình.

2. Hướng phát triển

Hướng phát triển trong tương lai:

- Thay thế GLoVe, BERT bởi UniLM trong mô hình NRMS
- Thêm phần abstract và tăng độ dài lịch sử click trong mô hình Fastformer + PLM – NR

TÀI LIỆU THAM KHẢO

[1] MIND: A Large-scale Dataset for News Recommendation, Fangzhao Wu et al.

https://msnews.github.io/assets/doc/ACL2020_MIND.pdf

[2] Neural News Recommendation with Multi-Head Self-Attention, Chuhan Wu et al.

https://aclanthology.org/D19-1671.pdf

[3] Fastformer: Additive Attention Can Be All You Need, Chuhan Wu et al.

https://arxiv.org/pdf/2108.09084.pdf