Xử lý ngôn ngữ tự nhiên

Các chủ đề nâng cao

Nội dung

1. Ứng dụng các mô hình NLP trong hệ gợi ý tin tức (News neural recommendation)

2. Dịch máy đa ngôn ngữ (Multi-lingual machine translation)

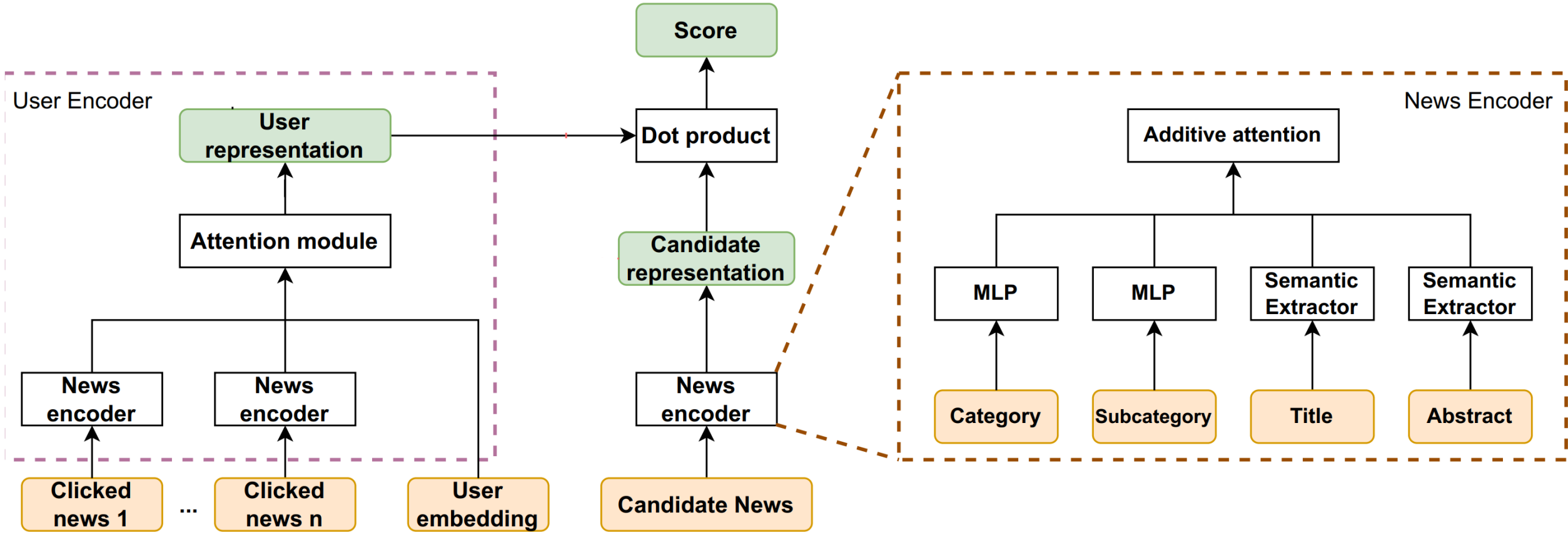
3. Hệ thống truy hồi thông tin dựa trên mô hình sinh ngôn ngữ

1. News neural recommendation

- Phát biểu bài toán:
  - Items: Là các bản tin (news)
    - Title: Tiêu đề bản tin
    - Snippet, content: Là nội dung của bản tin
    - Category: Chuyên mục
    - Sub-categories: Các chuyên mục con
  - Users: Là các người dùng đọc tin tức
    - Mỗi người dung gồm có danh sách các bản tin đã đọc
  - Mục tiêu: Gợi ý các bản tin phù hợp cho người dùng

Kiến trúc tổng quan

- Cấu tạo bởi 3 thành phần: News encoder, user encoder, click predictor



1.1. Mô hình gợi ý đa góc nhìn

- Một bài báo thường chứa thông tin ở nhiều phần khác nhau: tiêu đề, nội dung, thể loại.
  - Các thông tin này được dung để tìm ra sở thích và xu hướng đọc báo của người dùng.
- Các thành phần này có các tính chất khác nhau: tiêu đề ngắn và súc tích, nội dung thì dài và chi tiết, thể loại thì thường là cụm từ. Cách sử dụng khác nhau:

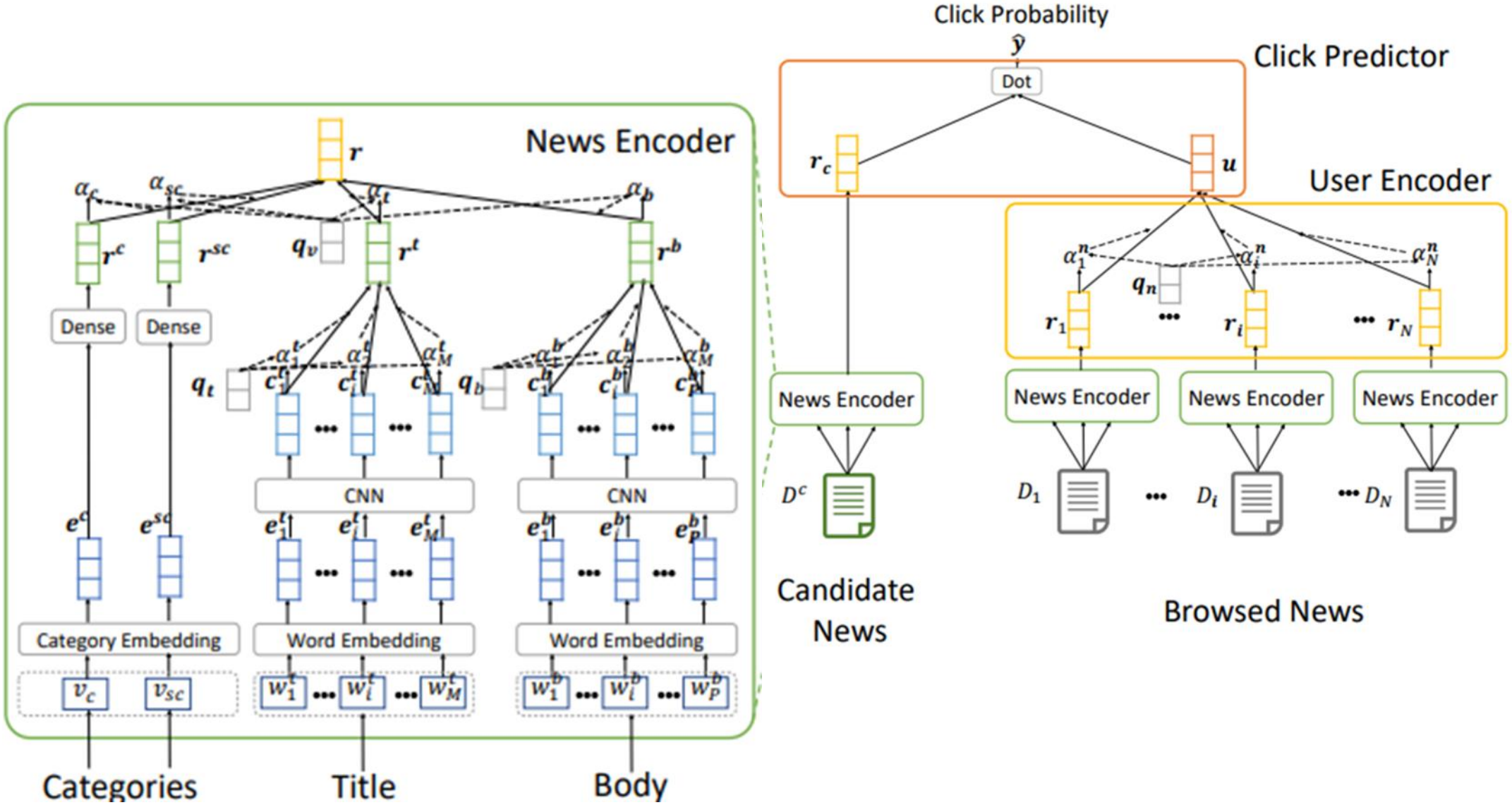
Category	Sports	Entertainment
Title	Astros improve outfield, agree to 2-year deal with Brantley	The best games of 2018
Body	Outfielder Michael Brantley agreed to a two-year, $32 million contract with Houston, sources familiar with the deal told Yahoo Sports, bringing his steady left-handed bat to the top of an Astros ...	The Best Games of 2018 Superheroes, super-dads, and Super Mario parties brought the joy in 2018 to millions of players who ate up the year's impressive achievements in gaming ...

Mô hình gợi ý đa góc nhìn

- Các bài báo khác nhau thì chứa lượng thông tin khác nhau.
- Từ ngữ trong bài có mức độ quan trọng khác nhau.
- Mức độ phổ biến của các bài mà user đọc có thể giúp tìm ra sở thích
  - tin "thời tiết" có nhiều người khác cũng xem, đóng góp ít trong việc tìm sở thích user
  - tin "Olympic Toán" có ít người khác cùng xem, đóng góp nhiều hơn, cho thấy user quan tâm vấn đề này

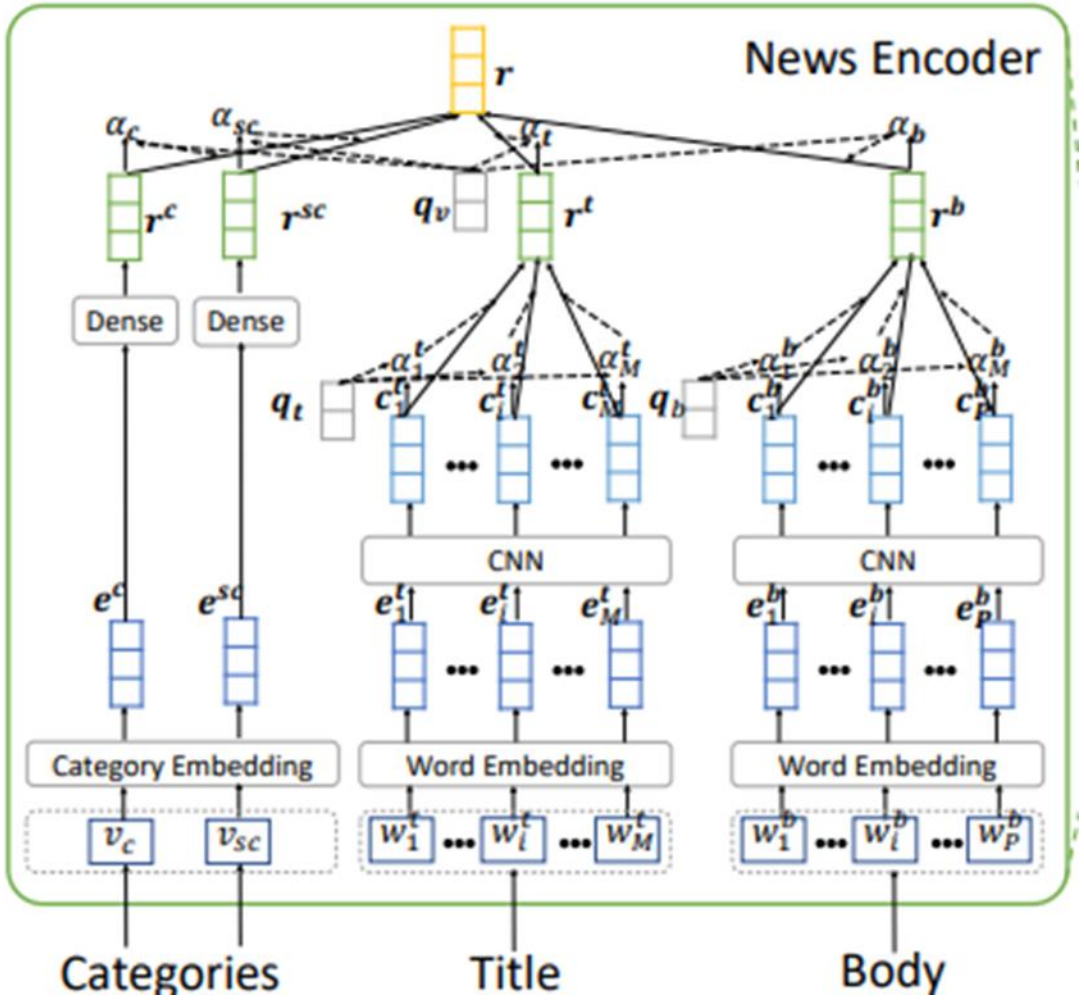
Category	Sports	Entertainment
Title	Astros improve outfield, agree to 2-year deal with Brantley	The best games of 2018
Body	Outfielder Michael Brantley agreed to a two-year, $32 million contract with Houston, sources familiar with the deal told Yahoo Sports, bringing his steady left-handed bat to the top of an Astros ...	The Best Games of 2018 Superheroes, super-dads, and Super Mario parties brought the joy in 2018 to millions of players who ate up the year's impressive achievements in gaming ...

Mô hình gợi ý đa góc nhìn



Mô hình gợi ý đa góc nhìn: News encoder

- Gồm 3 thành phần
  - Title encoder
    - Word embedding
    - Convolution neural network
    - Attention
  - Body encoder
    - Word embedding
    - Convolution neural network
    - Attention
  - Categories encoder
    - Categories embedding
    - Dense



Mô hình gợi ý đa góc nhìn: News encoder

- Title encoder
  - Word embedding: dictionary look-up
  - Convolution neural network:
  - Attention mức từ:

$$a_i^t = \mathbf{q}_t^T \tanh(\mathbf{V}_t \times \mathbf{c}_i^t + \mathbf{v}_t)$$

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

  - Biểu diễn của title

$$\mathbf{r}^t = \sum_{j=1}^{M} \alpha_j^t \mathbf{c}_j^t$$

Mô hình gợi ý đa góc nhìn: News encoder

- Body encoder thực hiện tương tự title encoder
- Categories encoder
  - Embedding
  - Dense

$$\mathbf{r}^c = \mathrm{ReLU}(\mathbf{V}_c \times \mathbf{e}^c + \mathbf{v}_c),$$
$$\mathbf{r}^{sc} = \mathrm{ReLU}(\mathbf{V}_s \times \mathbf{e}^{sc} + \mathbf{v}_s),$$

- Attention cho biểu diễn của title, categories, body

$$a_t = \mathbf{q}_v^T \tanh(\mathbf{U}_v \times \mathbf{r}^t + \mathbf{u}_v),$$
$$\alpha_t = \frac{\exp(a_t)}{\exp(a_t) + \exp(a_b) + \exp(a_c) + \exp(a_{sc})}$$

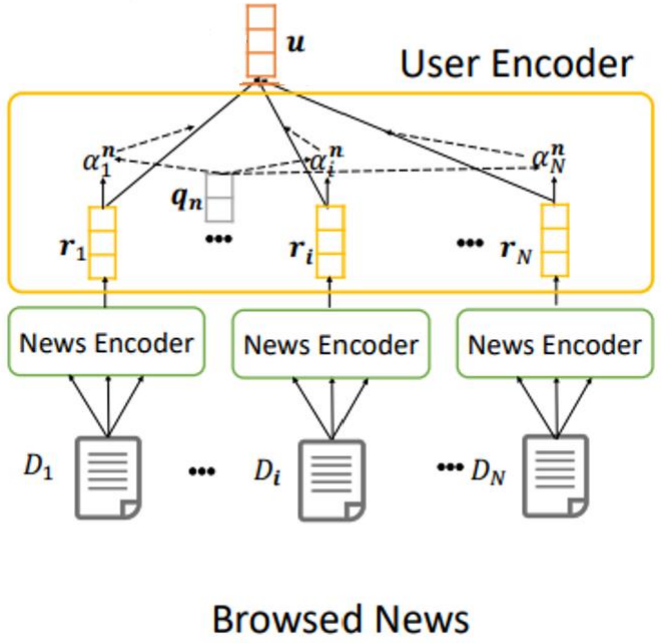
- Biểu diễn item:

$$\mathbf{r} = \alpha_c \mathbf{r}^c + \alpha_{sc} \mathbf{r}^{sc} + \alpha_t \mathbf{r}^t + \alpha_b \mathbf{r}^b.$$

Mô hình gợi ý đa góc nhìn: User encoder

- Áp dụng attention cho biểu diễn item user đã xem

$$a_i^n = \mathbf{q}_n^T \tanh(\mathbf{W}_n \times \mathbf{r}_i + \mathbf{b}_n),$$
$$\alpha_i^n = \frac{\exp(a_i^n)}{\sum_{j=1}^{N} \exp(a_j^n)},$$



- User được biểu diễn thông qua biểu diễn của các item đã xem

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

Mô hình gợi ý đa góc nhìn: Click prediction

- Dự đoán khả năng user click vào item:

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

- Xây dựng hàm loss: Cho mỗi user, với mỗi tin được đọc (positive news) $\hat{y}^+$, K tin chưa được đọc (negative news) [$\hat{y}_1^-$, $\hat{y}_2^-$, …, $\hat{y}_K^-$]sẽ được lấy mẫu ngẫu nhiên => K+1 news
- Hàm loss:

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

Trong đó: S là tập positive news và

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

Mô hình gợi ý đa góc nhìn: Click prediction

- Dự đoán khả năng user click vào item:

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

- Xây dựng hàm loss: Cho mỗi user, với mỗi tin được đọc (positive news) $\hat{y}^+$, K tin chưa được đọc (negative news) [$\hat{y}_1^-$, $\hat{y}_2^-$, …, $\hat{y}_K^-$]sẽ được lấy mẫu ngẫu nhiên => K+1 news
- Hàm loss:

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

Trong đó: S là tập positive news và

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

Mô hình gợi ý đa góc nhìn: Thử nghiệm

- Dữ liệu MSN news (https://msnews.github.io/) trong 1 tháng:

# users	10,000	**avg. # words per title**	11.29
# news	42,255	**avg. # words per body**	730.72
# impressions	445,230	**# positive samples**	489,644
# samples	7,141,584	**# negative samples**	6,651,940

- Kết quả:

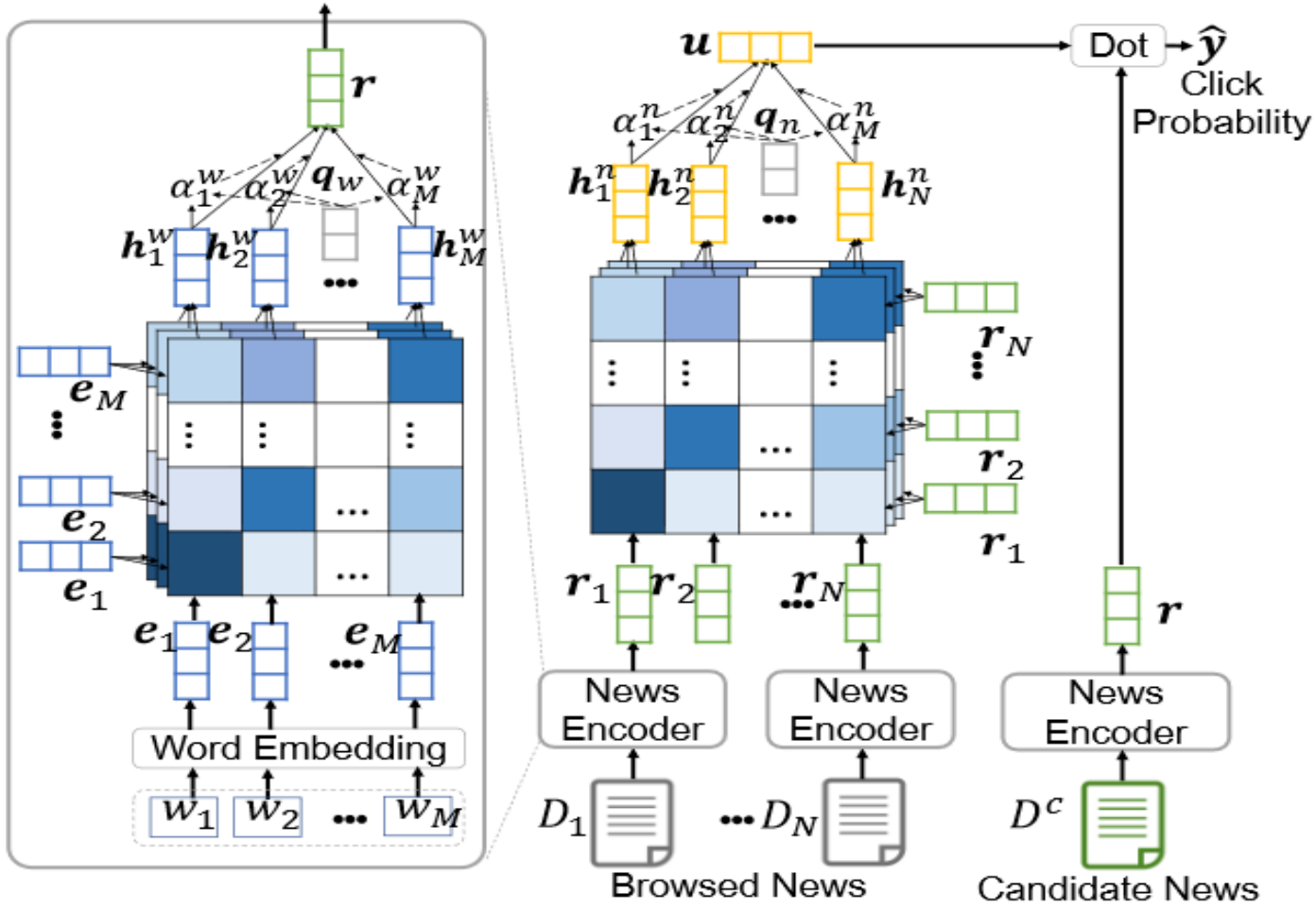
Methods	**AUC**	**MRR**	**nDCG@5**	**nDCG@10**
LibFM	0.5880	0.3054	0.3202	0.4090
CNN	0.5909	0.3068	0.3221	0.4109
DSSM	0.6114	0.3188	0.3261	0.4185
Wide&Deep	0.5846	0.3009	0.3167	0.4062
DeepFM	0.5896	0.3066	0.3221	0.4117
DFM	0.5996	0.3133	0.3288	0.4165
DKN	0.5966	0.3113	0.3286	0.4165
NAML*	**0.6434**	**0.3411**	**0.3670**	**0.4501**

1.2. Mô hình gợi ý với cơ chế tự chú ý

- Sự tương tác giữa các từ trong tiêu đề tin tức rất quan trọng để hiểu được tin tức
- Các bài báo khác nhau được duyệt bởi cùng một người dùng sẽ có liên quan
- Các từ khác nhau có tầm quan trọng khác nhau trong việc đại diện cho tin tức
- Các bài báo khác nhau được duyệt bởi cùng một người dùng có tầm quan trọng khác nhau trong việc đại diện cho người dùng

Mô hình gợi ý với cơ chế tự chú ý: Kiến trúc

- Gồm 3 thành phần
  - News encoder
  - User encoder
  - Click predictor



Mô hình gợi ý với cơ chế tự chú ý: News encoder

- News encoder

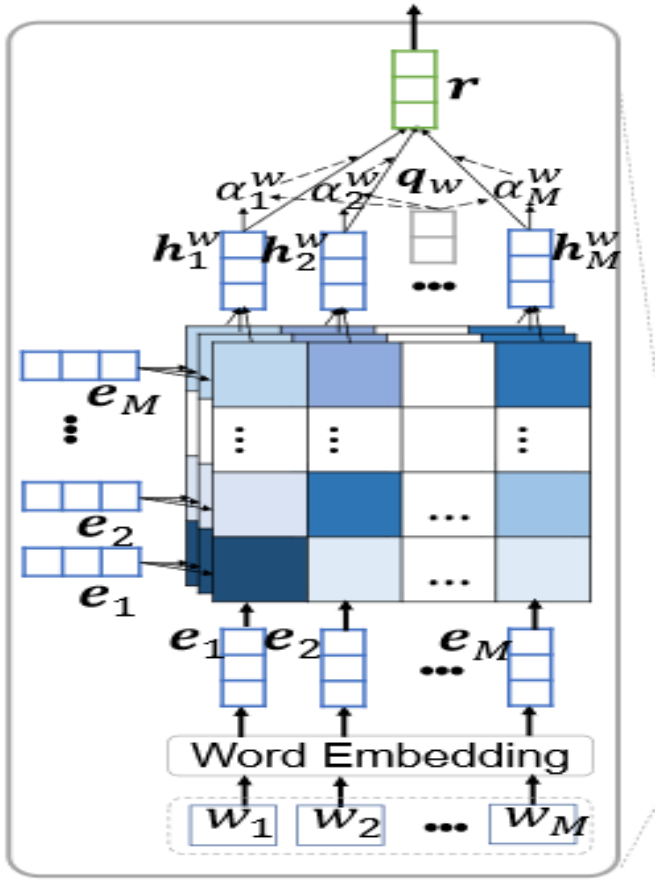
$$\alpha_{i,j}^k = \frac{\exp(\mathbf{e}_i^T \mathbf{Q}_k^w \mathbf{e}_j)}{\sum_{m=1}^{M} \exp(\mathbf{e}_i^T \mathbf{Q}_k^w \mathbf{e}_m)},$$

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

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

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

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



Mô hình gợi ý với cơ chế tự chú ý: User encoder

- User encoder

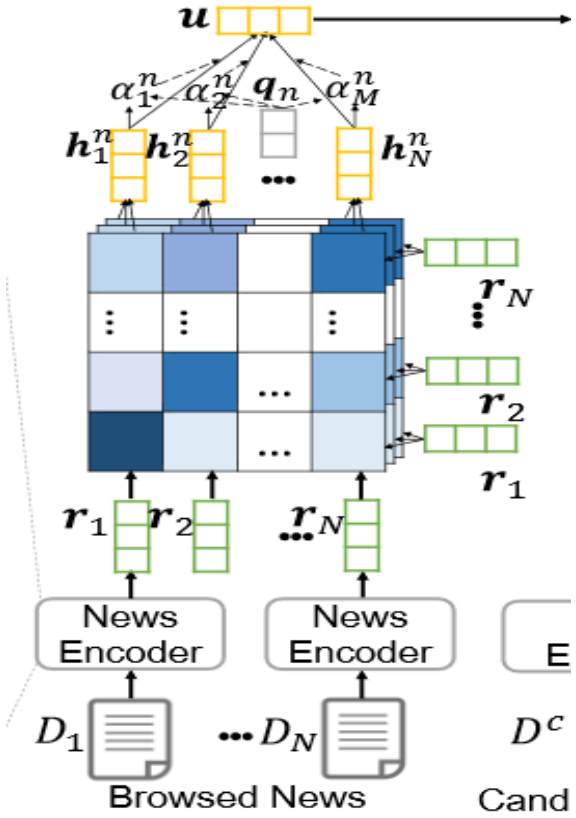
$$\beta_{i,j}^{k} = \frac{\exp(\mathbf{r}_i^T \mathbf{Q}_k^n \mathbf{r}_j)}{\sum_{m=1}^{M} \exp(\mathbf{r}_i^T \mathbf{Q}_k^n \mathbf{r}_m)},$$

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

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

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

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



Mô hình gợi ý với cơ chế tự chú ý: Click prediction

- Dự đoán khả năng user click vào item:

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

- Xây dựng hàm loss: Cho mỗi user, với mỗi tin được đọc (positive news) $\hat{y}^+$, K tin chưa được đọc (negative news) $[\hat{y}_1^-, \hat{y}_2^-, \ldots, \hat{y}_K^-]$sẽ được lấy mẫu ngẫu nhiên => K+1 news
- Hàm loss:

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

Trong đó: S là tập positive news và

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

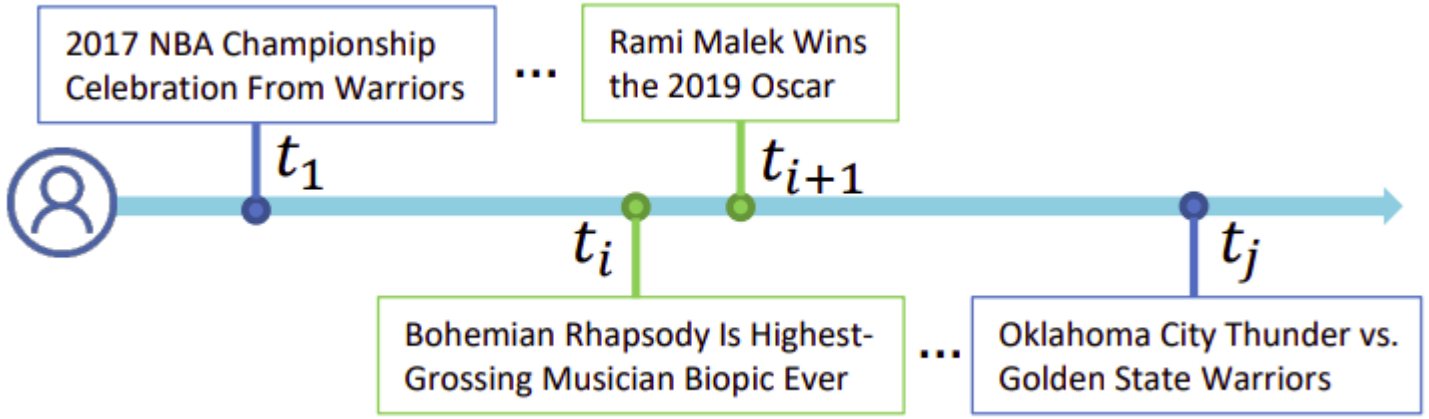
Mô hình gợi ý với cơ chế tự chú ý: Đánh giá

Methods	AUC	MRR	nDCG@5	nDCG@10
LibFM	0.5661	0.2414	0.2689	0.3552
DSSM	0.5949	0.2675	0.2881	0.3800
Wide&Deep	0.5812	0.2546	0.2765	0.3674
DeepFM	0.5830	0.2570	0.2802	0.3707
DFM	0.5861	0.2609	0.2844	0.3742
DKN	0.6032	0.2744	0.2967	0.3873
Conv3D	0.6051	0.2765	0.2987	0.3904
GRU	0.6102	0.2811	0.3035	0.3952
NRMS*	**0.6275**	**0.2985**	**0.3217**	**0.4139**

1.3. Mô hình gợi ý dựa trên sở thích ngắn hạn và dài hạn

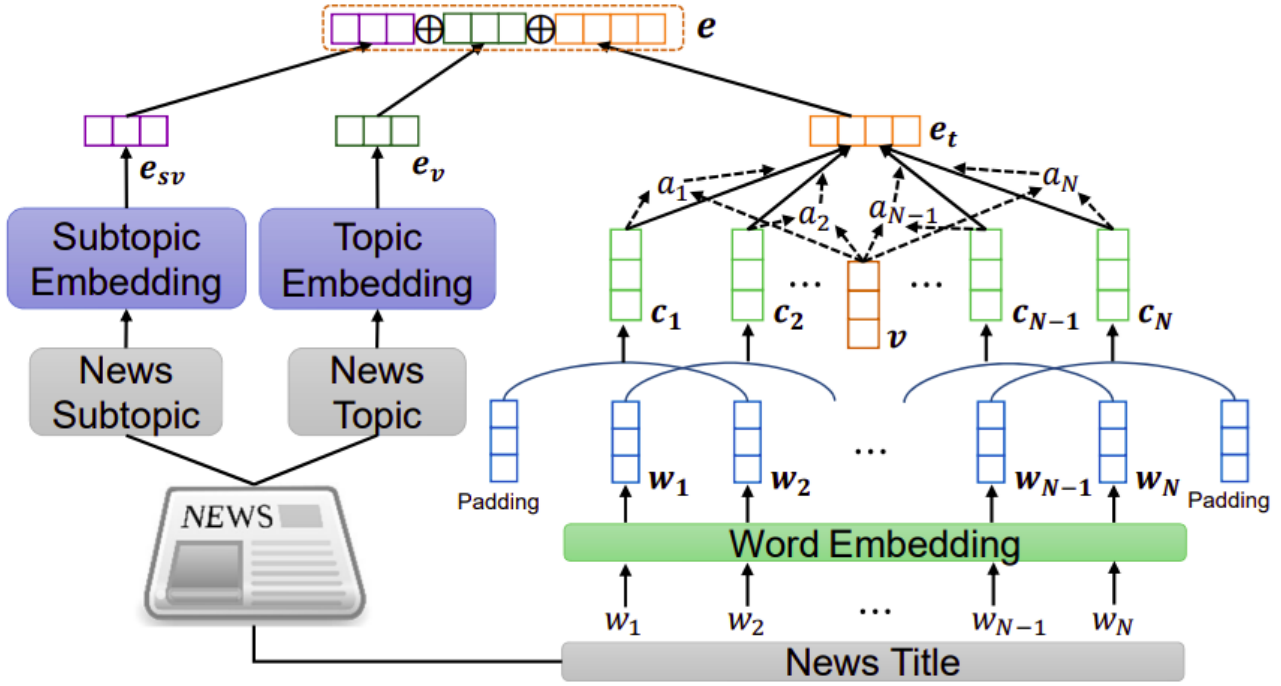
- User có sở thích dài hạn và ngắn hạn. => Cần phải khai thác cả 2.

=> Tiếp cận theo hướng biểu diễn user theo long-term và short-term



Mô hình gợi ý dựa trên sở thích ngắn hạn và dài hạn

- News encoder



Mô hình gợi ý dựa trên sở thích ngắn hạn và dài hạn

- News encoder: Sử dụng thông tin title, category (topic), subcategories
- Title encoder.
  - Title được embedding (dùng pre-train word2vec,..)
  - Sau khi embedding, đưa qua mạng CNN để trích xuất context:

$$\boldsymbol{c}_i = \text{ReLU}(\boldsymbol{C} \times \boldsymbol{w}_{[i-M:i+M]} + \boldsymbol{b}),$$

  - Sau đó đưa qua attention weight:
    - Với v,v_b là các tham số được học trong quá trình training.

$$a_i = \tanh(\boldsymbol{v} \times \boldsymbol{c}_i + v_b),$$

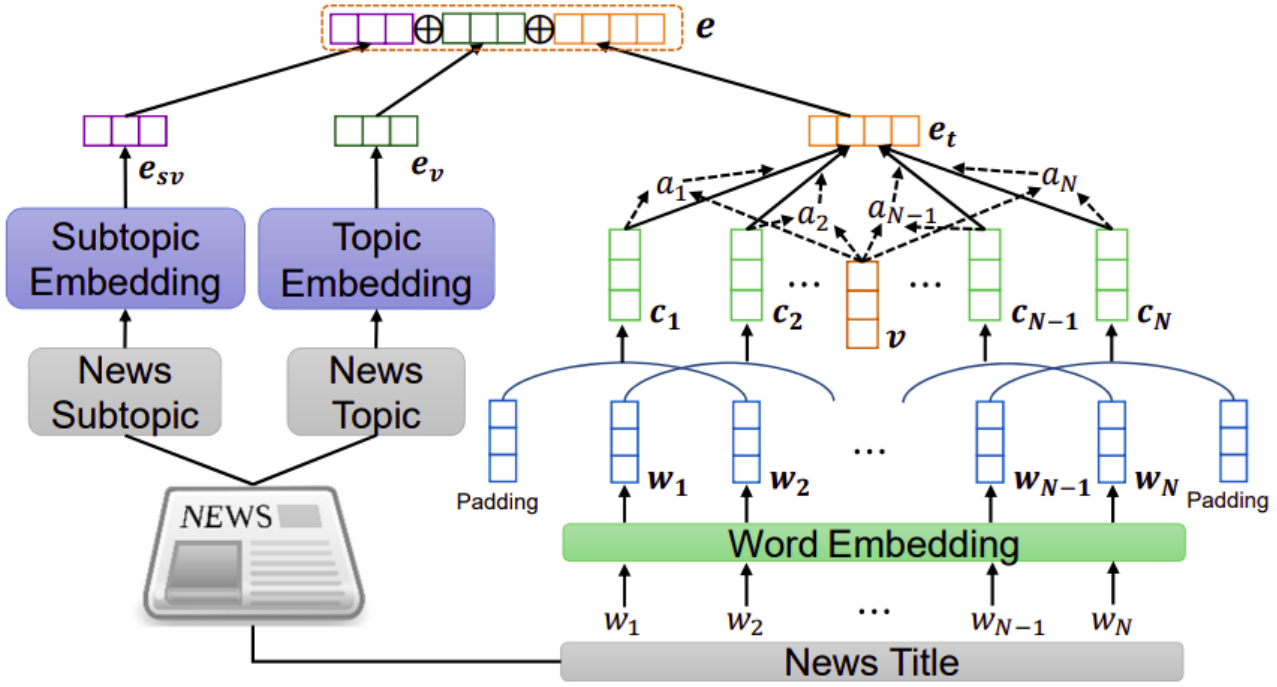
$$\alpha_i = \frac{\exp(a_i)}{\sum_{j=1}^{N} \exp(a_j)},$$

  - Sau cùng, vector đại diện cho title được lấy tổng có trọng số của các c_i

$$\boldsymbol{e}_t = \sum_{i=1}^{N} \alpha_i \boldsymbol{c}_i.$$

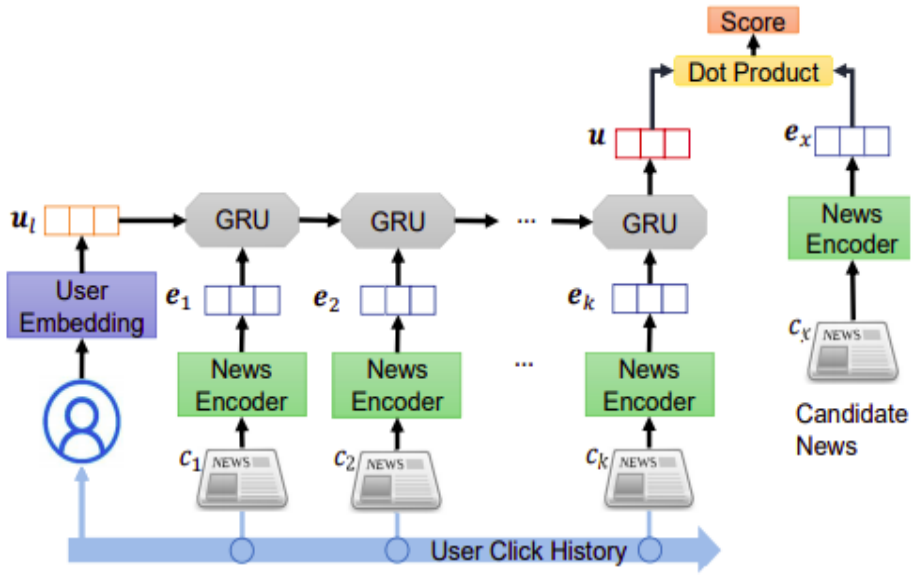
Mô hình gợi ý dựa trên sở thích ngắn hạn và dài hạn

- News encoder: Sử dụng thông tin title, category (topic), subcategories
- Topic encoder
  - topic và subtopic categories được embedding (được học trong quá trình training)
- Sau cùng concat các vector lại sẽ thu được vector biểu diễn của news

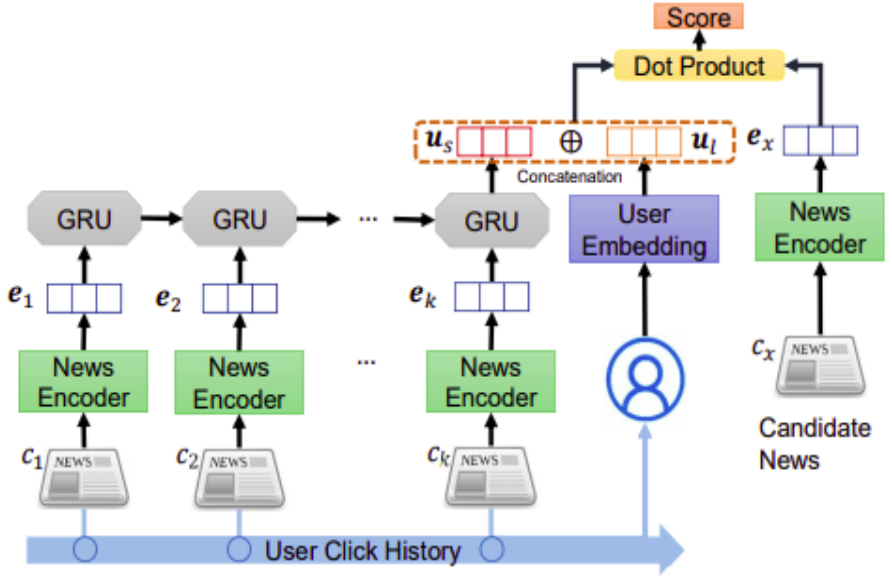


Mô hình gợi ý dựa trên sở thích ngắn hạn và dài hạn

User encoder: Short-term và Long-term
Sử dụng mạng GRU và user embedding



(a) LSTUR-ini.



(b) LSTUR-con.

Mô hình gợi ý dựa trên sở thích ngắn hạn và dài hạn

Methods	AUC	MRR	nDCG@5	nDCG@10
LibFM	56.52 ± 1.31	25.53 ± 0.81	26.66 ± 1.04	34.72 ± 0.95
DeepFM	58.13 ± 1.69	27.01 ± 0.20	28.37 ± 0.57	36.78 ± 0.62
Wide & Deep	58.07 ± 0.55	27.07 ± 0.37	28.51 ± 0.45	36.93 ± 0.43
DSSM	58.43 ± 0.58	27.25 ± 0.49	28.31 ± 0.60	36.91 ± 0.54
CNN	61.13 ± 0.77	29.44 ± 0.73	31.44 ± 0.87	39.51 ± 0.74
DKN	61.25 ± 0.78	29.47 ± 0.64	31.54 ± 0.79	39.59 ± 0.67
GRU	62.69 ± 0.16	30.24 ± 0.13	32.56 ± 0.17	40.55 ± 0.13
LSTUR-con	63.47 ± 0.10	30.94 ± 0.14	33.43 ± 0.13	41.34 ± 0.13
LSTUR-ini	**63.56 ± 0.42**	**30.98 ± 0.32**	**33.45 ± 0.39**	**41.37 ± 0.36**

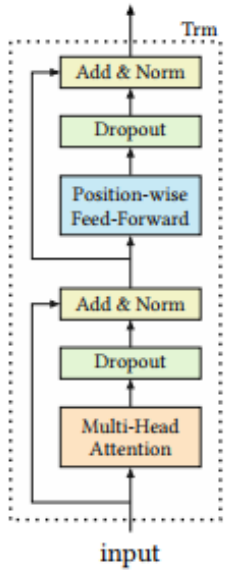
1.4. Một số kiến trúc khác

- Các phương pháp gợi ý dựa trên chuỗi:
    - RNN, Bert dùng cho dữ liệu dạng chuỗi.
    - Ví dụ: **Chuỗi**: x1 -> x2 -> x3 -> x4 -> x5 -> x6
    - Mỗi phần tử của chuỗi nằm ở một bước thời gian (**time step**) khác nhau và có quan hệ về mặt thứ tự với nhau
    - Mô hình hóa quan hệ thứ tự giữa các phần tử trong chuỗi

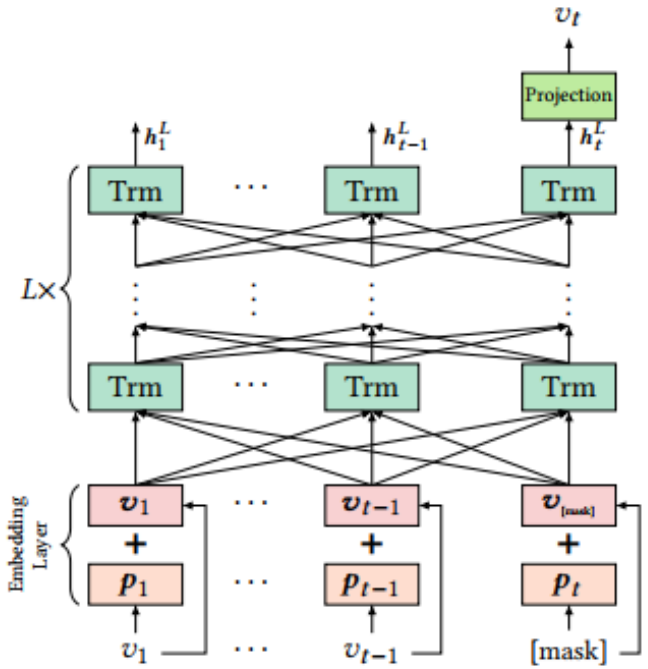
=> Có thể áp dụng cho hệ gợi ý với chuỗi các items trong session: Dự đoán item tiếp theo

1.4. Một số kiến trúc khác

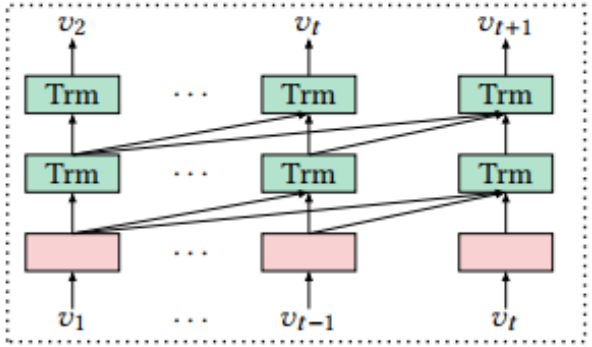
- RNN, SASRec, Bert4rec:



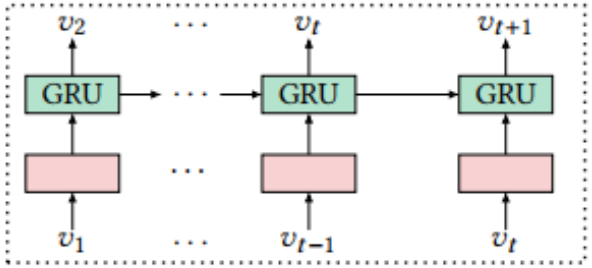
(a) Transformer Layer.



(b) BERT4Rec model architecture.



(c) SASRec model architecture.



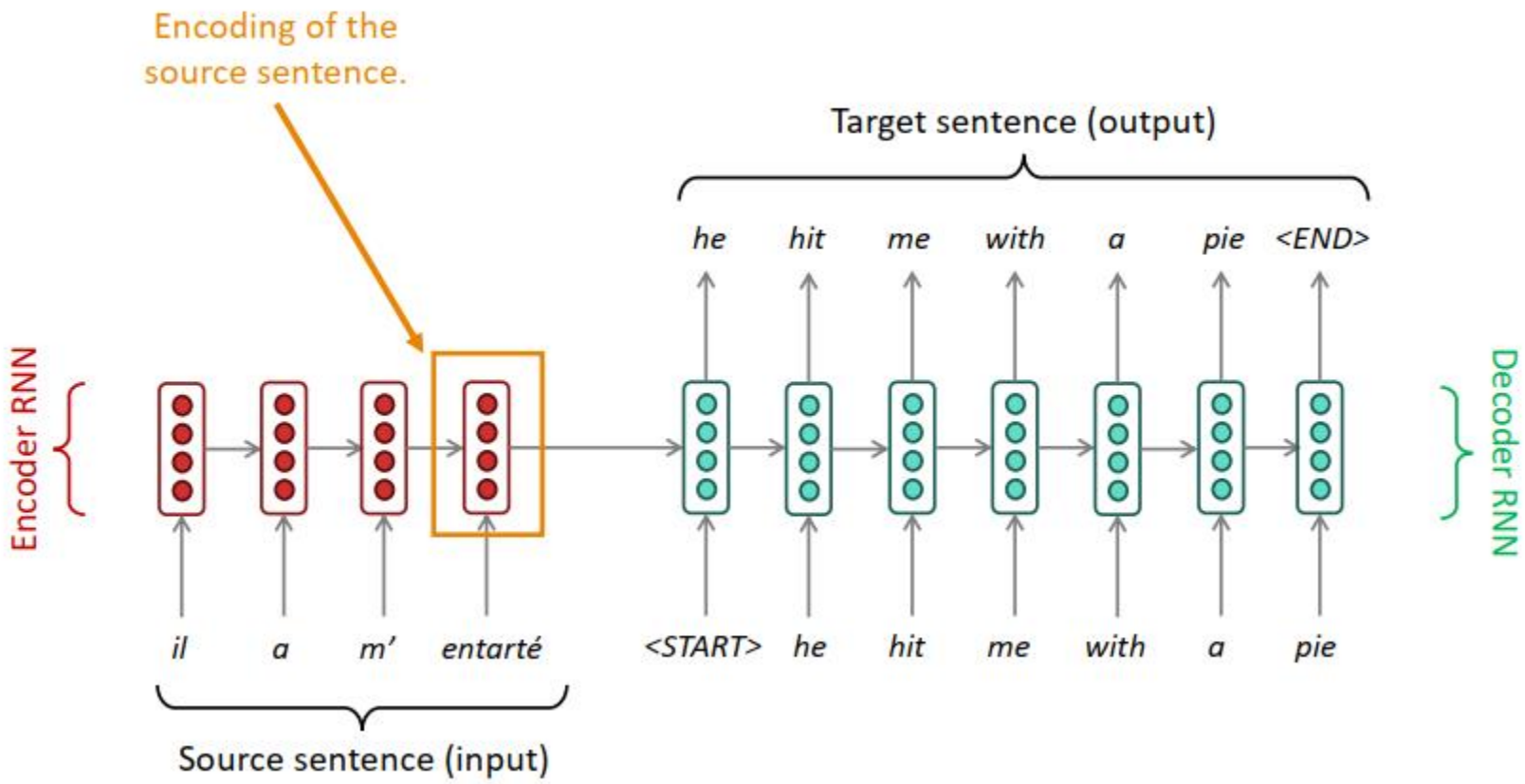
(d) RNN based sequential recommendation methods.

2. Multi-lingual Machine Translation

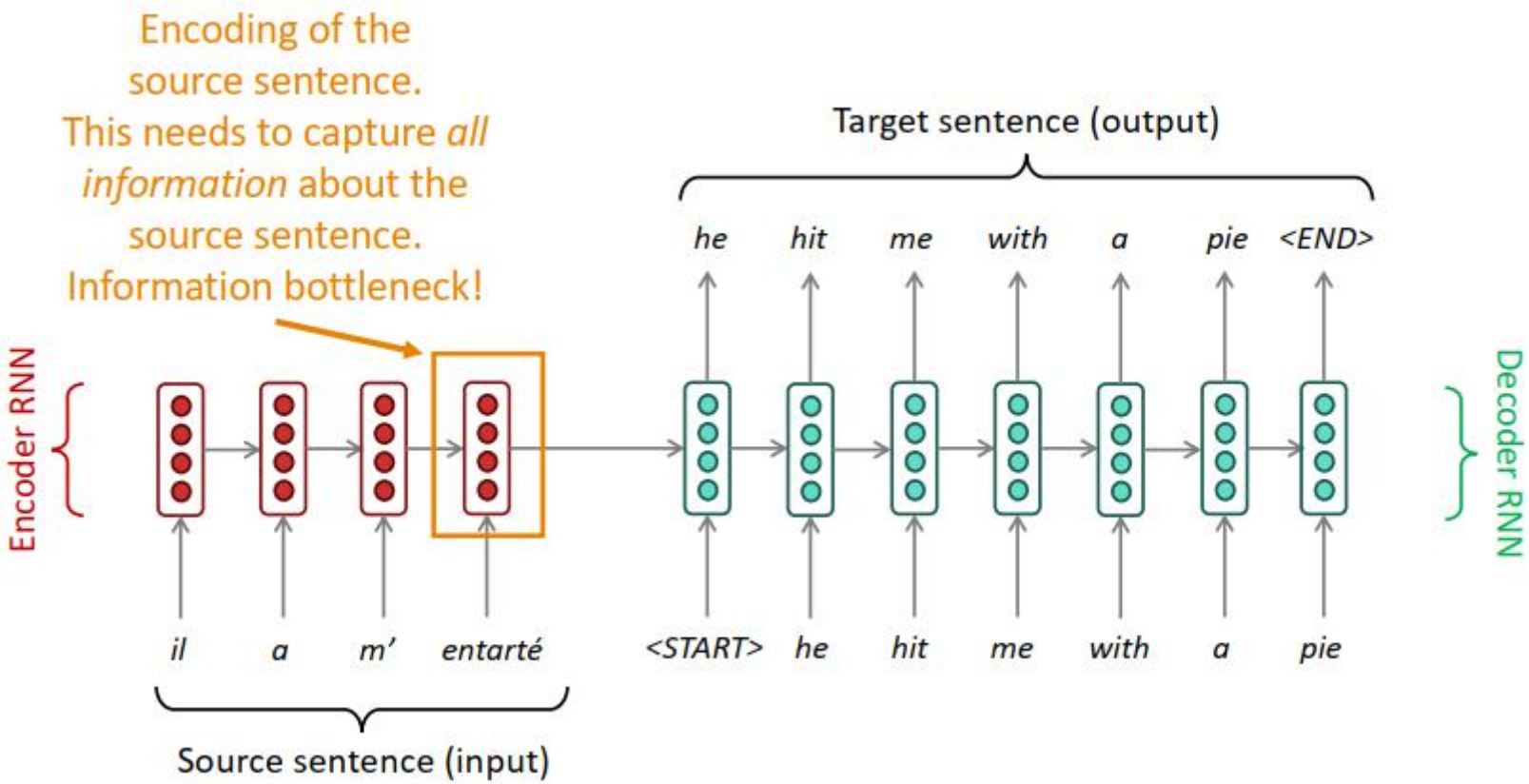
1. Nhắc lại mô hình dịch máy với Seq2seq
2. Transformer
3. Cách pre-train Transformer?

   BART và T5 model
4. Multilingual neural machine translation
   1. Opus-MT
   2. mBart
   3. M2M

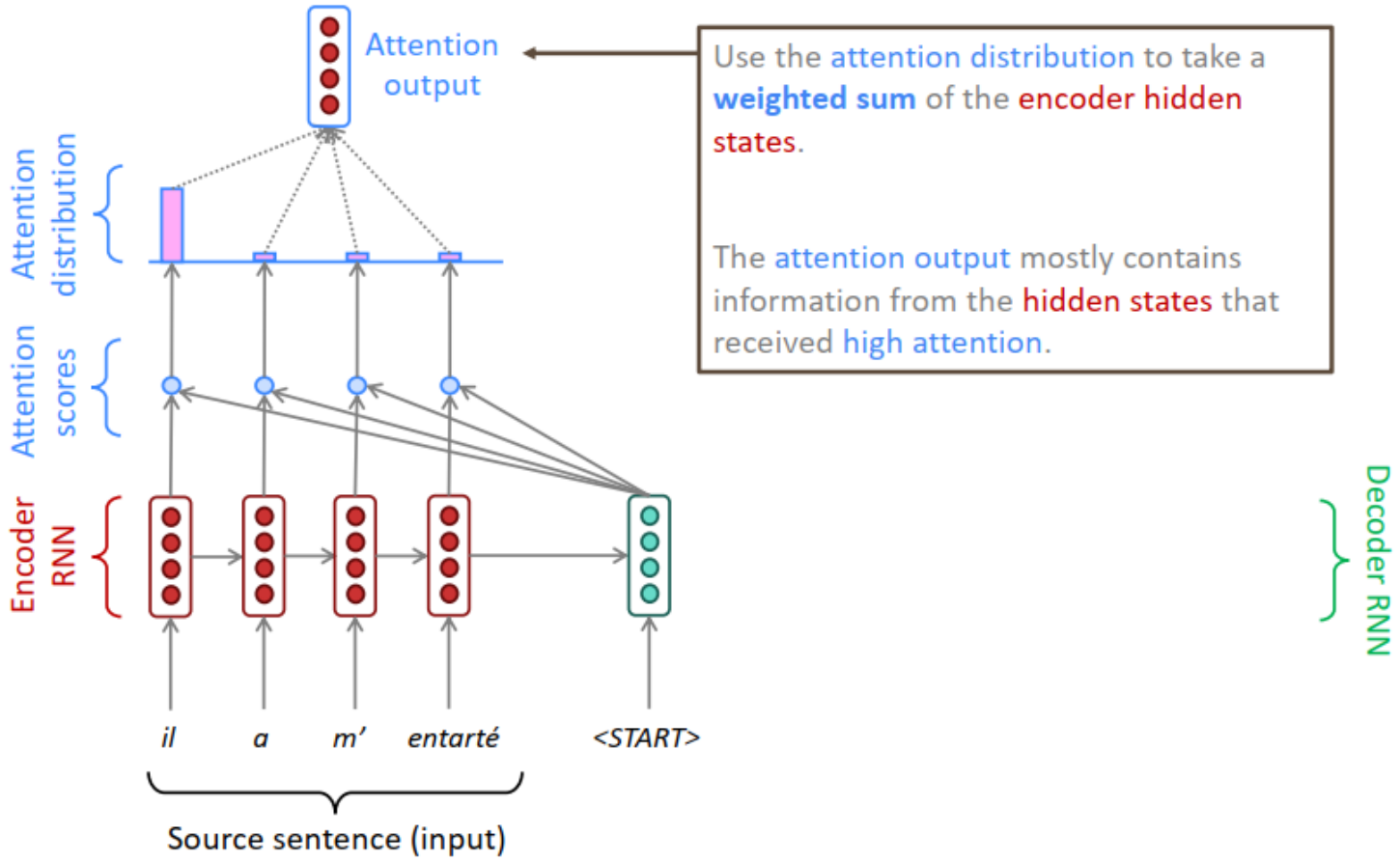
1. Mô hình Sequence-to-Sequence: Recurrent neural network



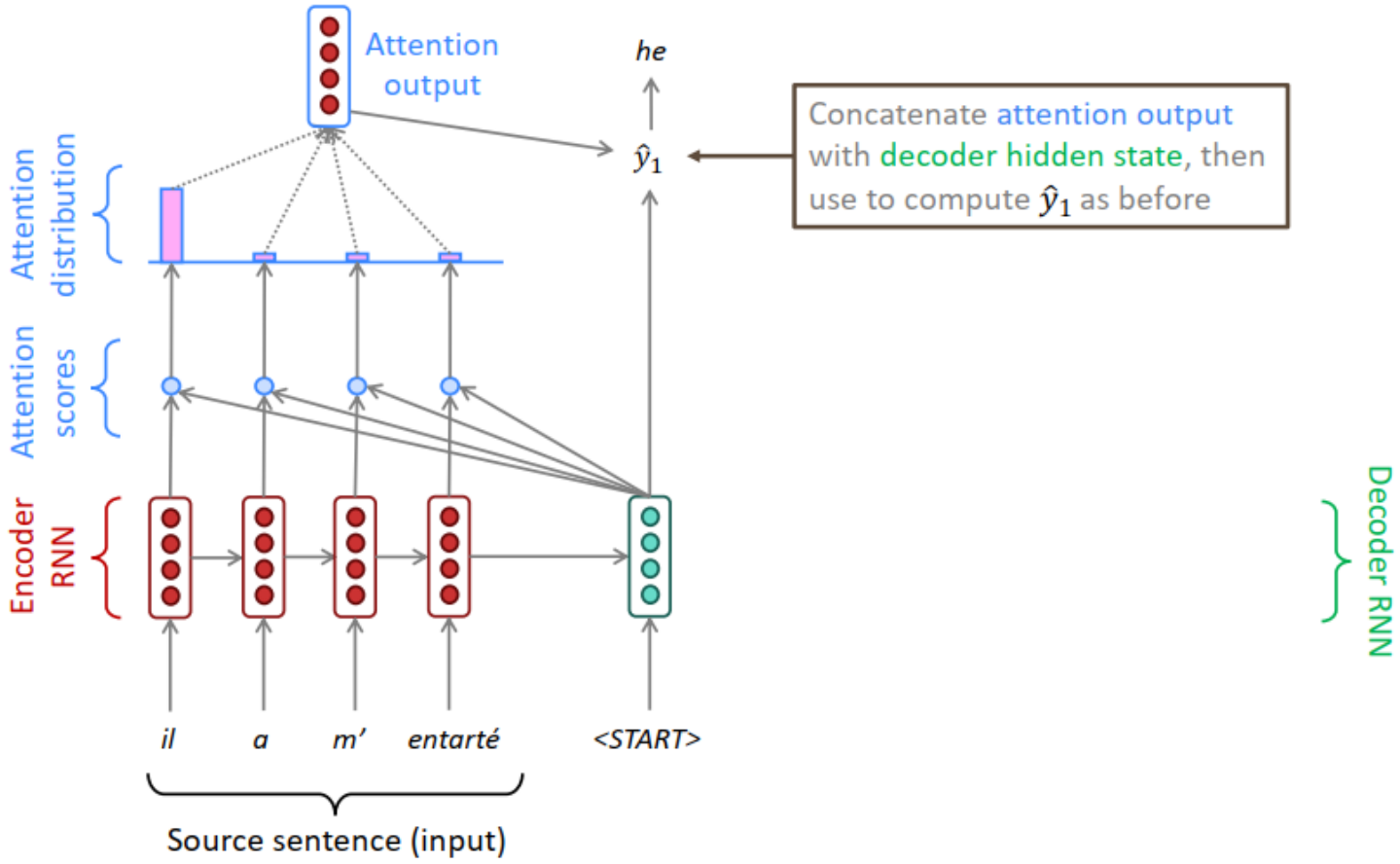
1. Mô hình Sequence-to-Sequence: The bottleneck problem



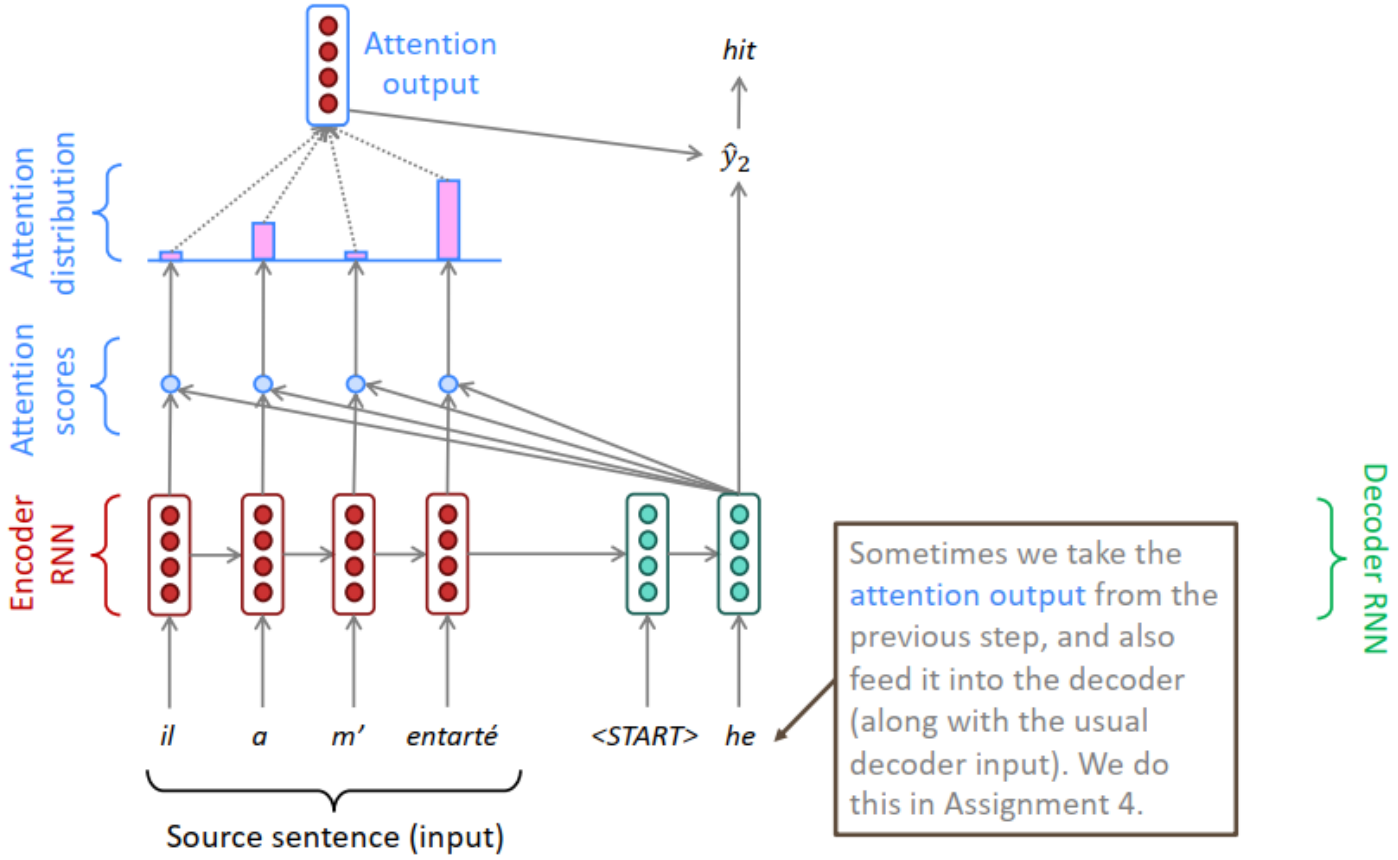
Mô hình Sequence-to-Sequence với cơ chế chú ý



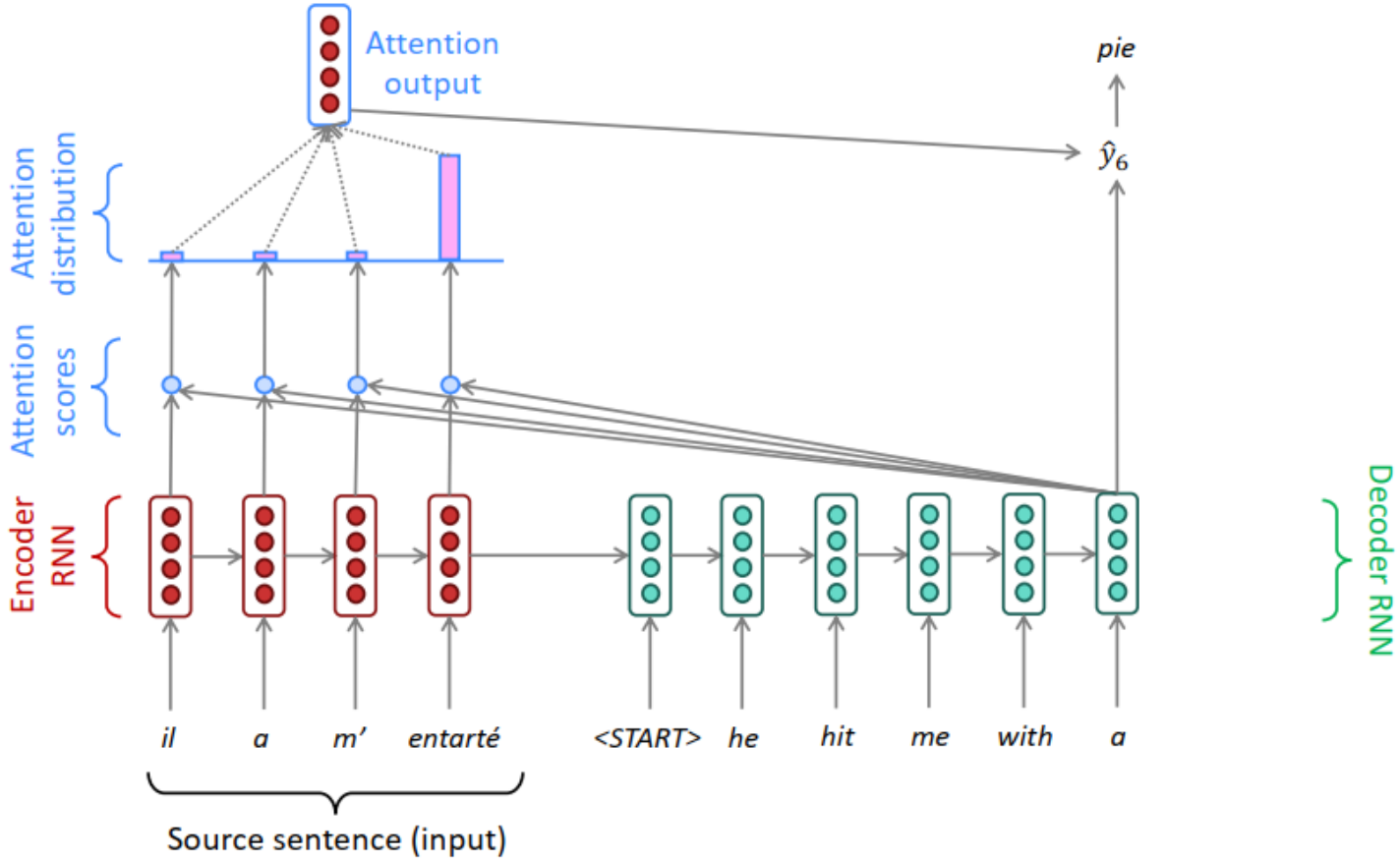
Mô hình Sequence-to-Sequence với cơ chế chú ý



Mô hình Sequence-to-Sequence với cơ chế chú ý



Mô hình Sequence-to-Sequence với cơ chế chú ý



2. Transformer

- **Transformer** [Vaswani et al., 2017] là một giải pháp thay thế RNN khi xử lý dữ liệu dạng chuỗi
  > Loại bỏ liên kết hồi quy
  > Bắt phụ thuộc dài

- Kiến trúc gồm
  > Encoder
  > Decoder

Transformer

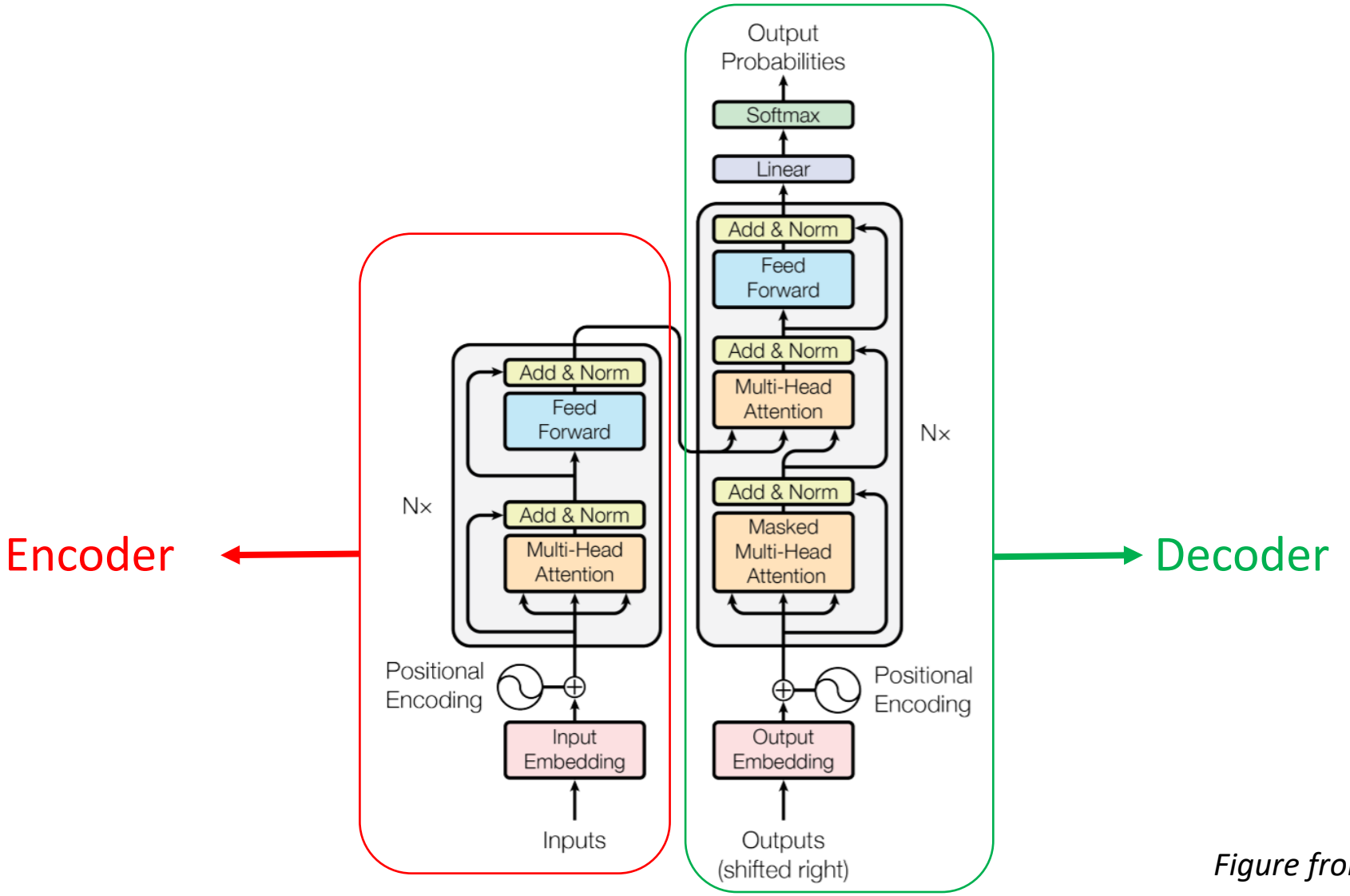
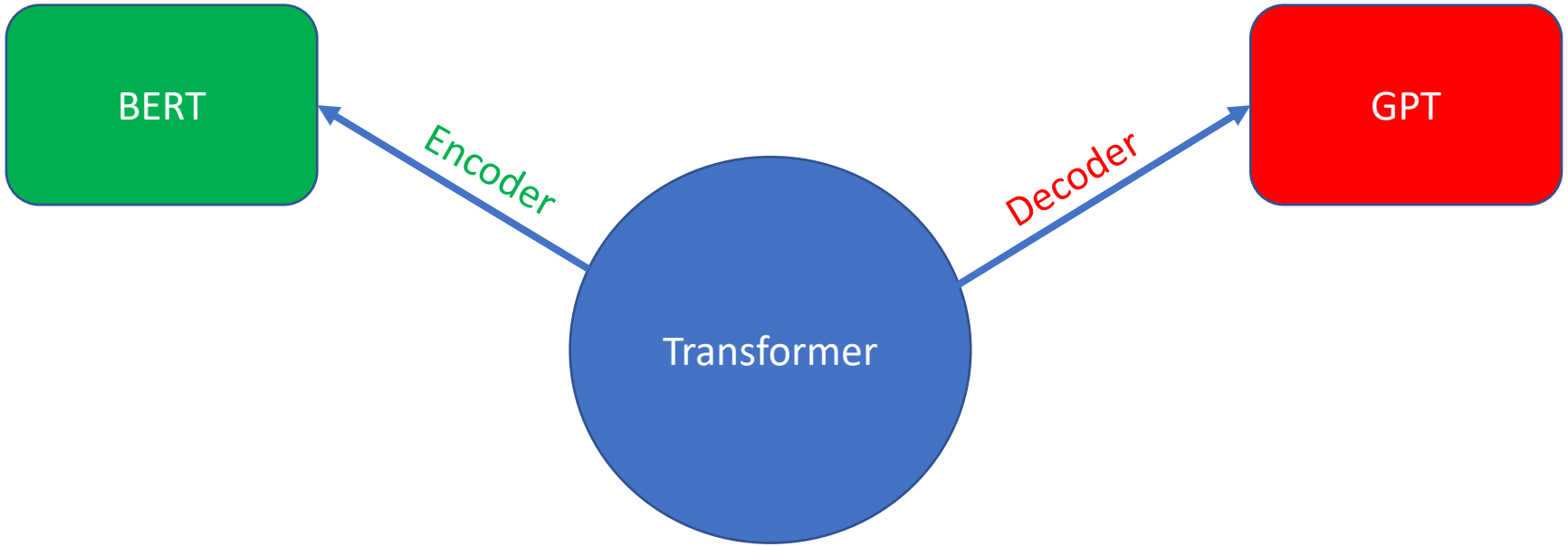


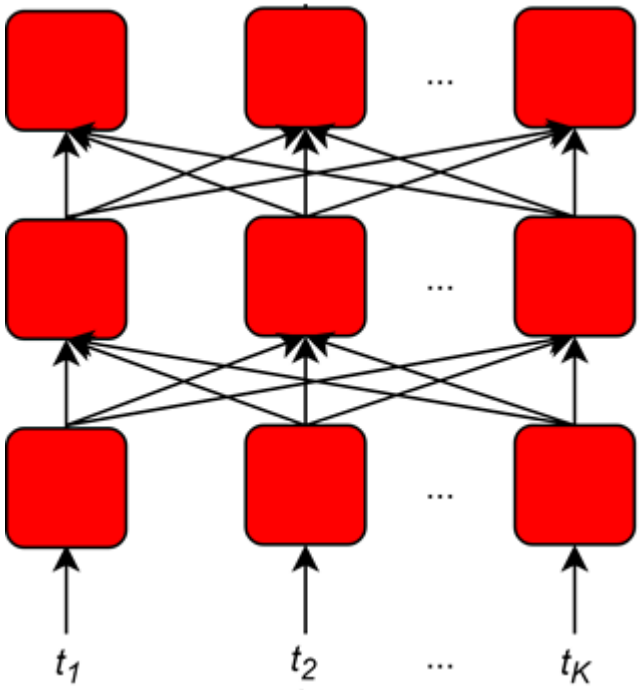
Figure from the original paper

Transformer

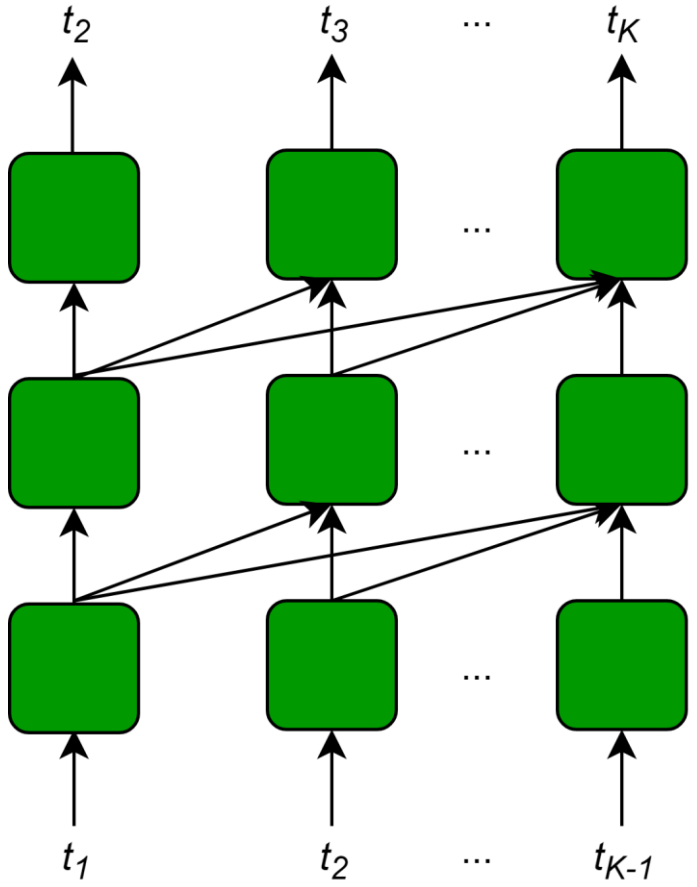


BERT - **B**idirectional **E**ncoder **R**epresentations from **T**ransformer [Devlin et al., 2018]
GPT - **G**enerative **P**retrained **T**ransformer [Radford et al., 2018]

Transformer: Encoder and decoder

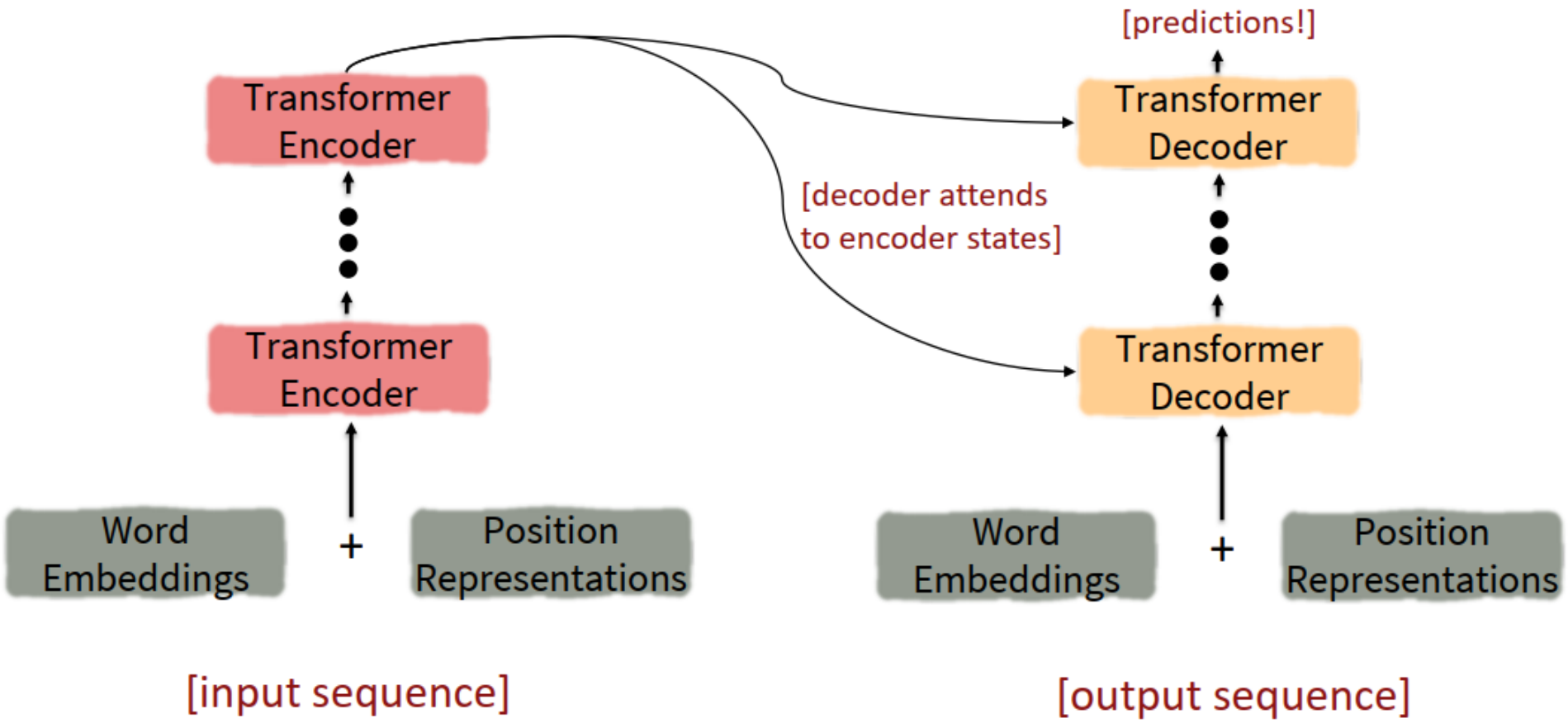


Encoder

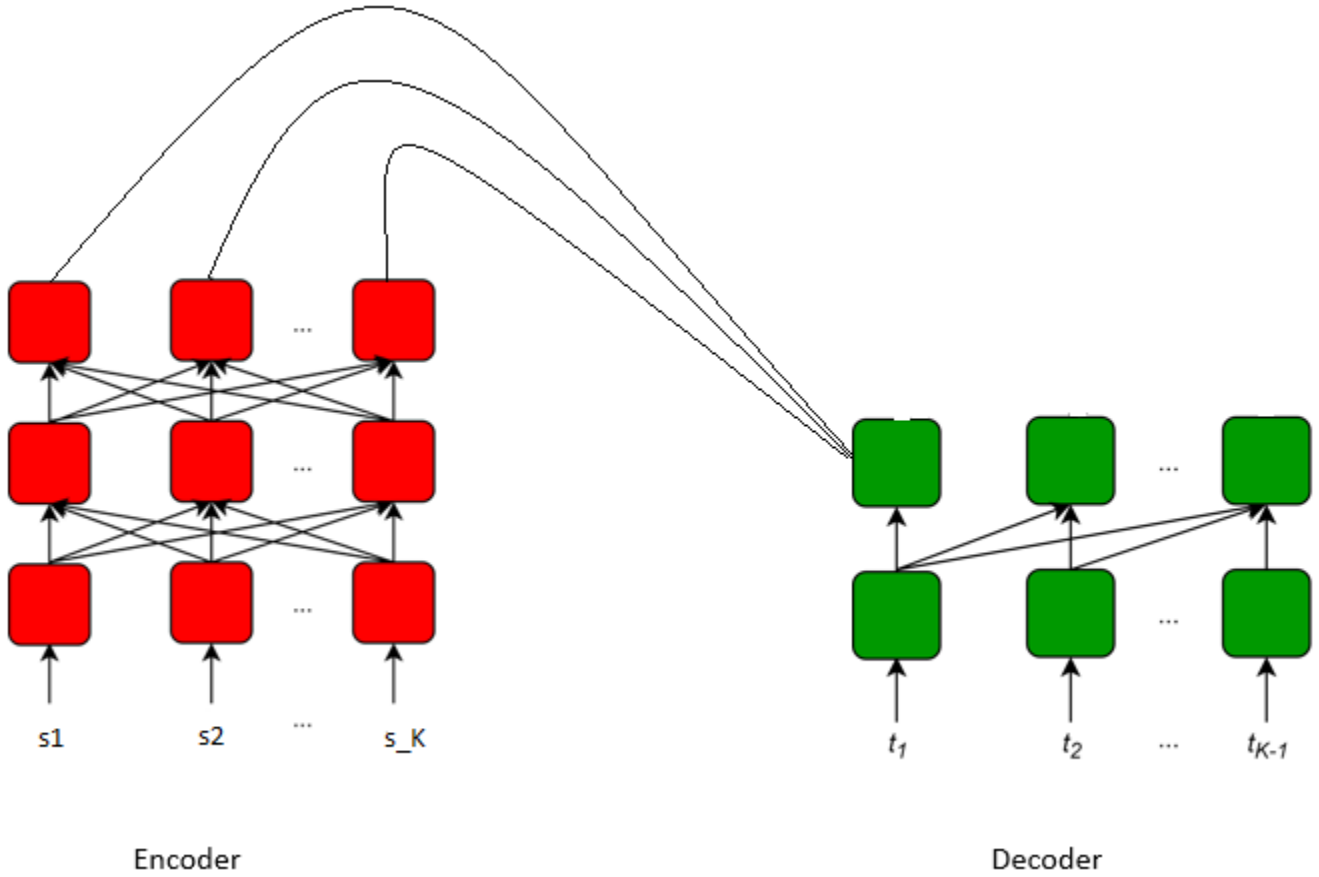


Decoder

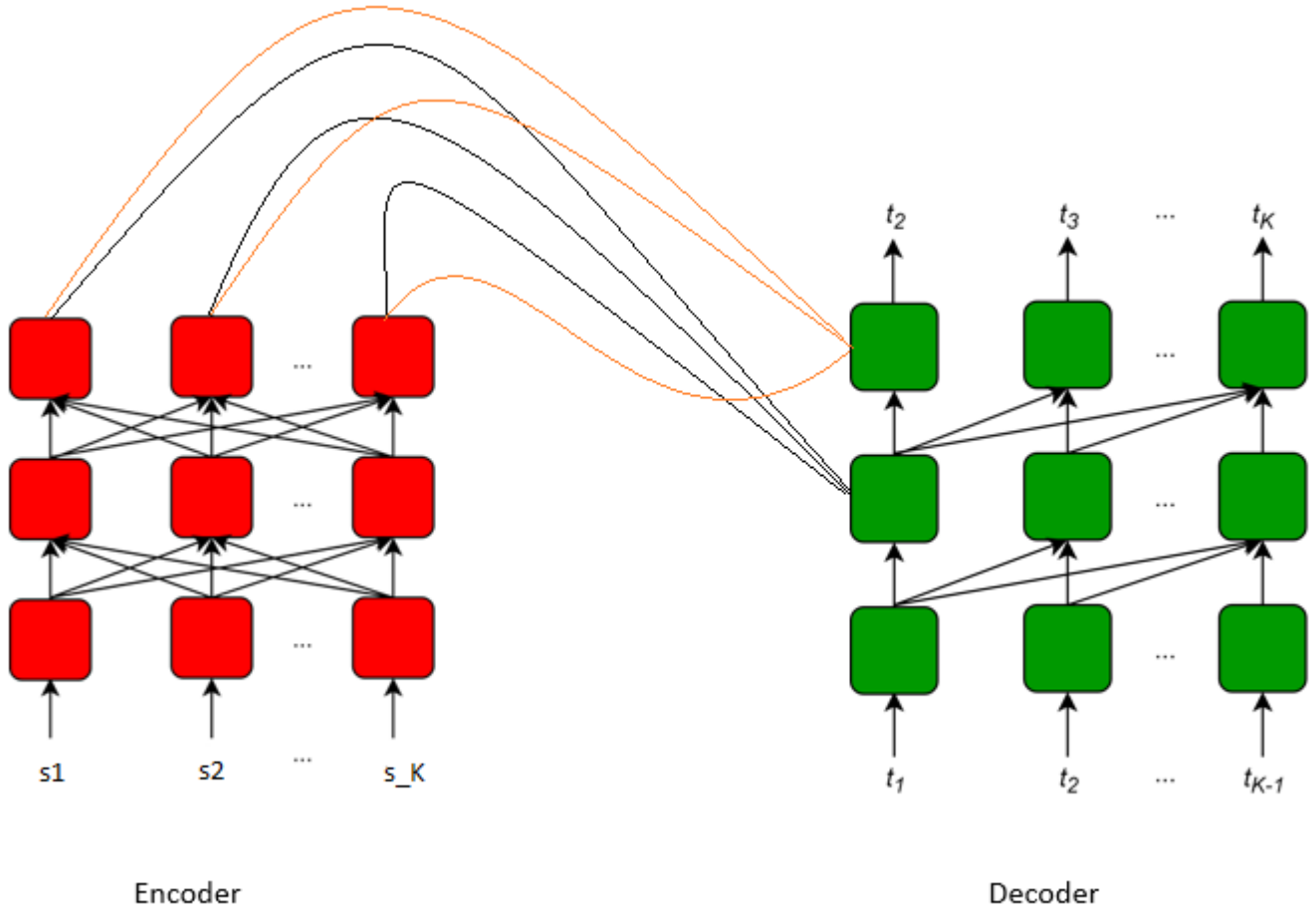
Transformer



Transformer



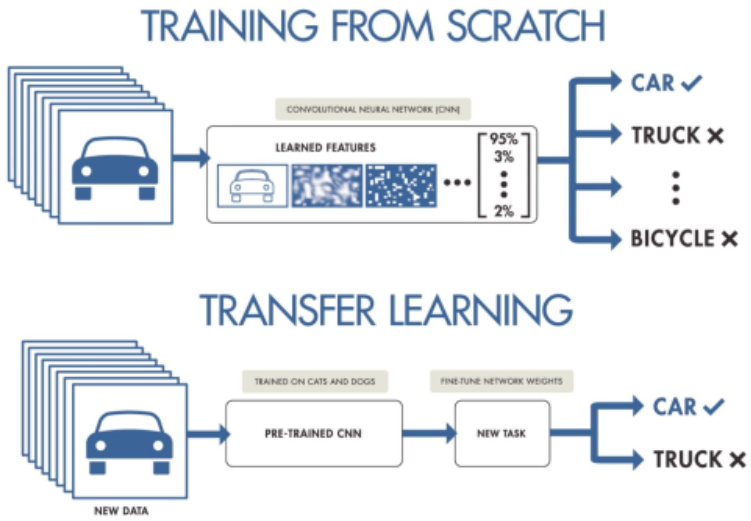
Transformer



3. Cách tiền huấn luyện transformer

Pre-training vs Fine Tuning

- Training an encoder/decoder to learn representation of data in UL fashion
- Use weights of a trained network as initialization for new but related task (transfer learning)



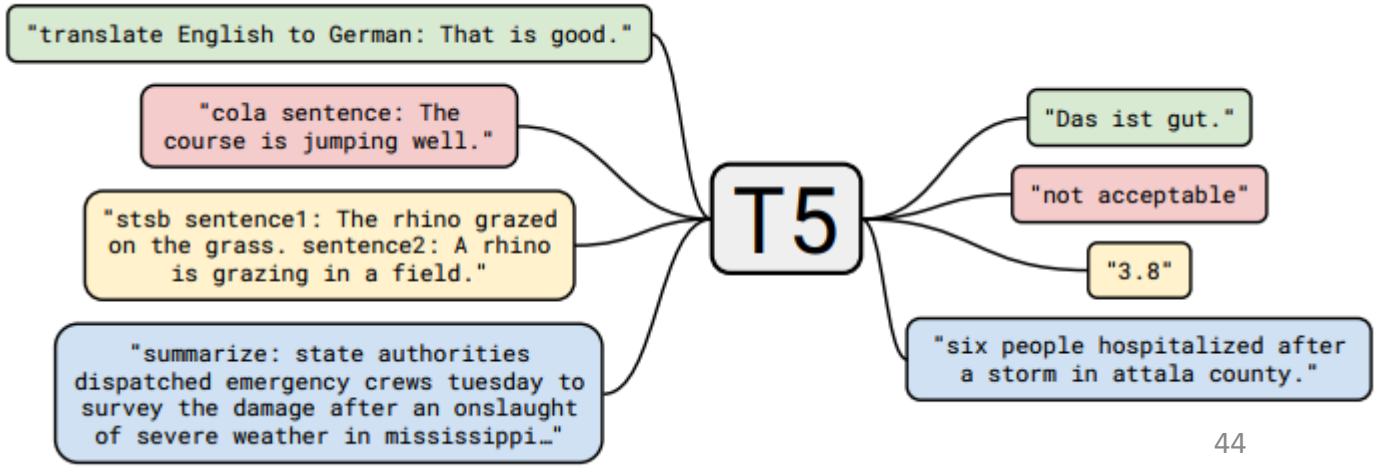
https://pl.pinterest.com/pin/672232681856247783/

Cách tiền huấn luyện transformer : T5 (et.al colin 2020)

- Pre-training: Unsupervised objectives on big datasets

Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style Devlin et al. (2018)	Thank you <M> <M> me to your party apple week .	*(original text)*
Deshuffling	party me for your to . last fun you inviting week Thank	*(original text)*
MASS-style Song et al. (2019)	Thank you <M> <M> me to your party <M> week .	*(original text)*
I.i.d. noise, replace spans	Thank you <X> me to your party <Y> week .	<X> for inviting <Y> last <Z>
I.i.d. noise, drop tokens	Thank you me to your party week .	for inviting last
Random spans	Thank you <X> to <Y> week .	<X> for inviting me <Y> your party last <Z>

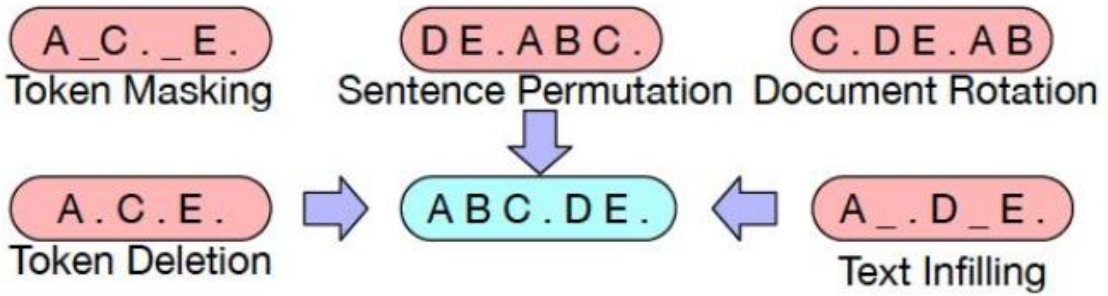
- Fine-tuning: Multi-tasks



Cách tiền huấn luyện transformer: Bart (lewis et al 2020)

Pre-training

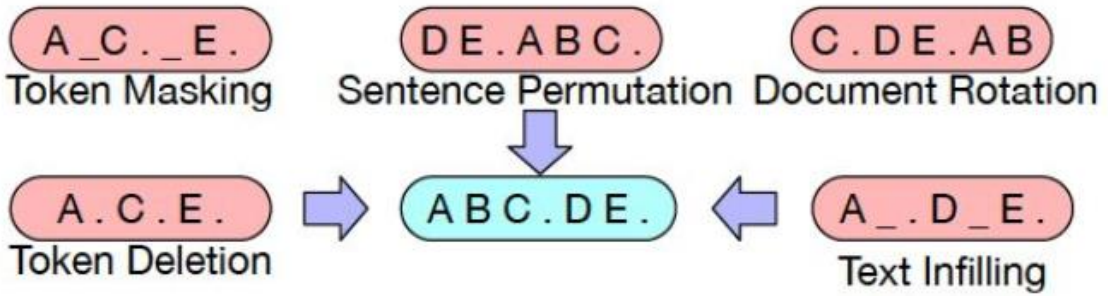
- Corrupt input sequence and minimize cross entropy between input and reconstruction
- Any noising scheme goes !
- Token masking - random token replaced with MASK
- Text infilling - model predicts how many tokens are missing from a span
- Document rotation - trains model to id start of document



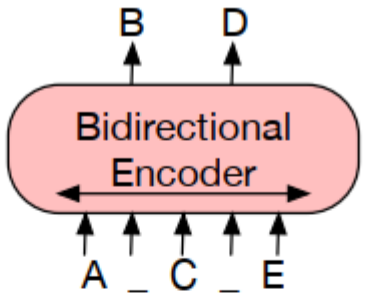
Cách tiền huấn luyện transformer: Bart (lewis et al 2020)

Pre-training

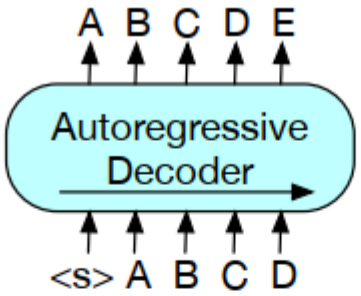
- Corrupt input sequence and minimize cross entropy between input and reconstruction
- Any noising scheme goes !
- Token masking - random token replaced with MASK
- Text infilling - model predicts how many tokens are missing from a span
- Document rotation - trains model to id start of document



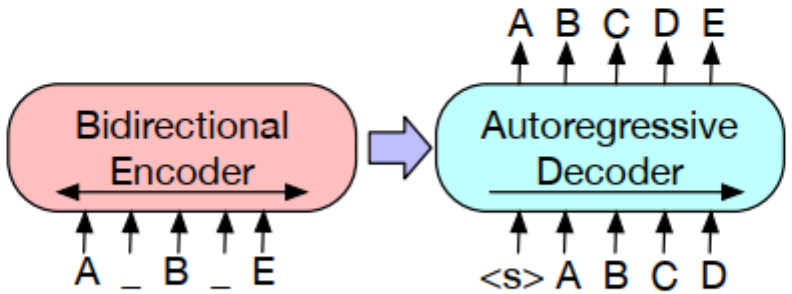
Cách tiền huấn luyện transformer: Bart (lewis et al 2020)



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



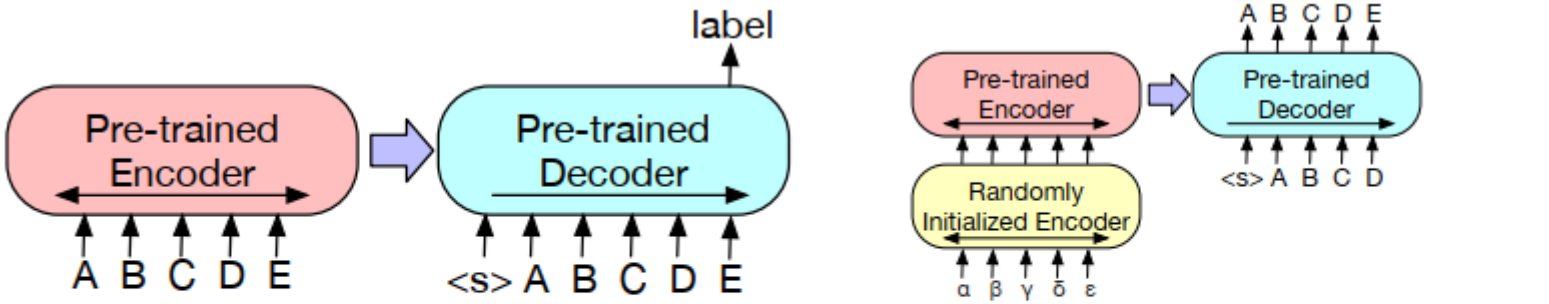
(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.



(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

Cách tiền huấn luyện transformer: Bart (lewis et al 2020)

- Fine-tuning



(a) To use BART for classification problems, the same input is fed into the encoder and decoder, and the representation from the final output is used.

(b) For machine translation, we learn a small additional encoder that replaces the word embeddings in BART. The new encoder can use a disjoint vocabulary.

Figure 3: Fine tuning BART for classification and translation.

Cách tiền huấn luyện transformer: Bart (lewis et al 2020)

- Some experimental results

	SQuAD 1.1 EM/F1	SQuAD 2.0 EM/F1	MNLI m/mm	SST Acc	QQP Acc	QNLI Acc	STS-B Acc	RTE Acc	MRPC Acc	CoLA Mcc
BERT	84.1/90.9	79.0/81.8	86.6/-	93.2	91.3	92.3	90.0	70.4	88.0	60.6
UniLM	-/-	80.5/83.4	87.0/85.9	94.5	-	92.7	-	70.9	-	61.1
XLNet	**89.0**/94.5	86.1/88.8	89.8/-	95.6	91.8	93.9	91.8	83.8	89.2	63.6
RoBERTa	88.9/**94.6**	**86.5/89.4**	**90.2/90.2**	96.4	92.2	94.7	**92.4**	86.6	**90.9**	**68.0**
BART	88.8/**94.6**	86.1/89.2	89.9/90.1	**96.6**	**92.5**	**94.9**	91.2	**87.0**	90.4	62.8

Table 2: Results for large models on SQuAD and GLUE tasks. BART performs comparably to RoBERTa and XLNet, suggesting that BART's uni-directional decoder layers do not reduce performance on discriminative tasks.

	CNN/DailyMail			XSum		
	R1	R2	RL	R1	R2	RL
Lead-3	40.42	17.62	36.67	16.30	1.60	11.95
PTGEN (See et al., 2017)	36.44	15.66	33.42	29.70	9.21	23.24
PTGEN+COV (See et al., 2017)	39.53	17.28	36.38	28.10	8.02	21.72
UniLM	43.33	20.21	40.51	-	-	-
BERTSUMABS (Liu & Lapata, 2019)	41.72	19.39	38.76	38.76	16.33	31.15
BERTSUMEXTABS (Liu & Lapata, 2019)	42.13	19.60	39.18	38.81	16.50	31.27
BART	**44.16**	**21.28**	**40.90**	**45.14**	**22.27**	**37.25**

Table 3: Results on two standard summarization datasets. BART outperforms previous work on summarization on two tasks and all metrics, with gains of roughly 6 points on the more abstractive dataset.

Cách tiền huấn luyện transformer: Bart (lewis et al 2020)

- Some experimental results:

	SQuAD 1.1 EM/F1	SQuAD 2.0 EM/F1	MNLI m/mm	SST Acc	QQP Acc	QNLI Acc	STS-B Acc	RTE Acc	MRPC Acc	CoLA Mcc
BERT	84.1/90.9	79.0/81.8	86.6/-	93.2	91.3	92.3	90.0	70.4	88.0	60.6
UniLM	-/-	80.5/83.4	87.0/85.9	94.5	-	92.7	-	70.9	-	61.1
XLNet	**89.0**/94.5	86.1/88.8	89.8/-	95.6	91.8	93.9	91.8	83.8	89.2	63.6
RoBERTa	88.9/**94.6**	**86.5/89.4**	**90.2/90.2**	96.4	92.2	94.7	**92.4**	86.6	**90.9**	**68.0**
BART	88.8/**94.6**	86.1/89.2	89.9/90.1	**96.6**	**92.5**	**94.9**	91.2	**87.0**	90.4	62.8

Table 2: Results for large models on SQuAD and GLUE tasks. BART performs comparably to RoBERTa and XLNet, suggesting that BART's uni-directional decoder layers do not reduce performance on discriminative tasks.

	CNN/DailyMail			XSum		
	R1	R2	RL	R1	R2	RL
Lead-3	40.42	17.62	36.67	16.30	1.60	11.95
PTGEN (See et al., 2017)	36.44	15.66	33.42	29.70	9.21	23.24
PTGEN+COV (See et al., 2017)	39.53	17.28	36.38	28.10	8.02	21.72
UniLM	43.33	20.21	40.51	-	-	-
BERTSUMABS (Liu & Lapata, 2019)	41.72	19.39	38.76	38.76	16.33	31.15
BERTSUMEXTABS (Liu & Lapata, 2019)	42.13	19.60	39.18	38.81	16.50	31.27
BART	**44.16**	**21.28**	**40.90**	**45.14**	**22.27**	**37.25**

Table 3: Results on two standard summarization datasets. BART outperforms previous work on summarization on two tasks and all metrics, with gains of roughly 6 points on the more abstractive dataset.

	ELI5		
	R1	R2	RL
Best Extractive	23.5	3.1	17.5
Language Model	27.8	4.7	23.1
Seq2Seq	28.3	5.1	22.8
Seq2Seq Multitask	28.9	5.4	23.1
BART	**30.6**	**6.2**	**24.3**

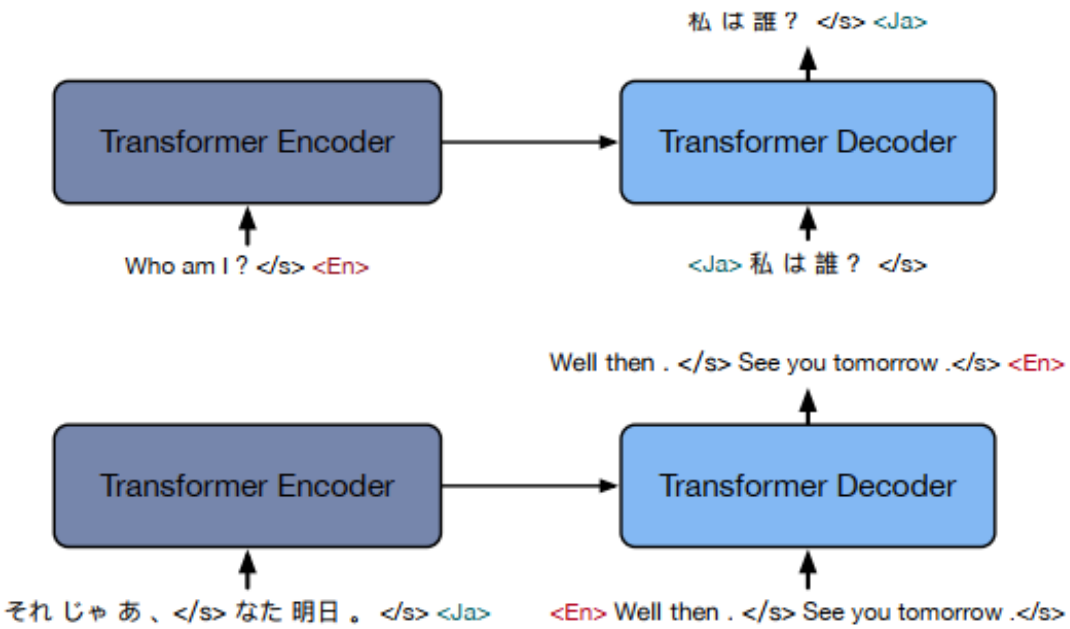
Table 5: BART achieves state-of-the-art results on the challenging ELI5 abstractive question answering dataset. Comparison models are from Fan et al. (2019).

	RO-EN
Baseline	36.80
Fixed BART	36.29
Tuned BART	**37.96**

Table 6: The performance (BLEU) of baseline and BART on WMT'16 RO-EN augmented with back-translation data. BART improves over a strong back-translation (BT) baseline by using monolingual English pre-training.

4. Multilingual neural machine translation (Opus-MT)

- Create vocabulary built with SentencePiece on the full multilingual dataset
- A special language id token is added at both the encoder and decoder
- Architecture: original Transformer
- Do not consider pre-training task



4. Multilingual neural machine translation (mBart)

- Architecture: Original Bart/Transformer
- Pre-train on multilingual datasets

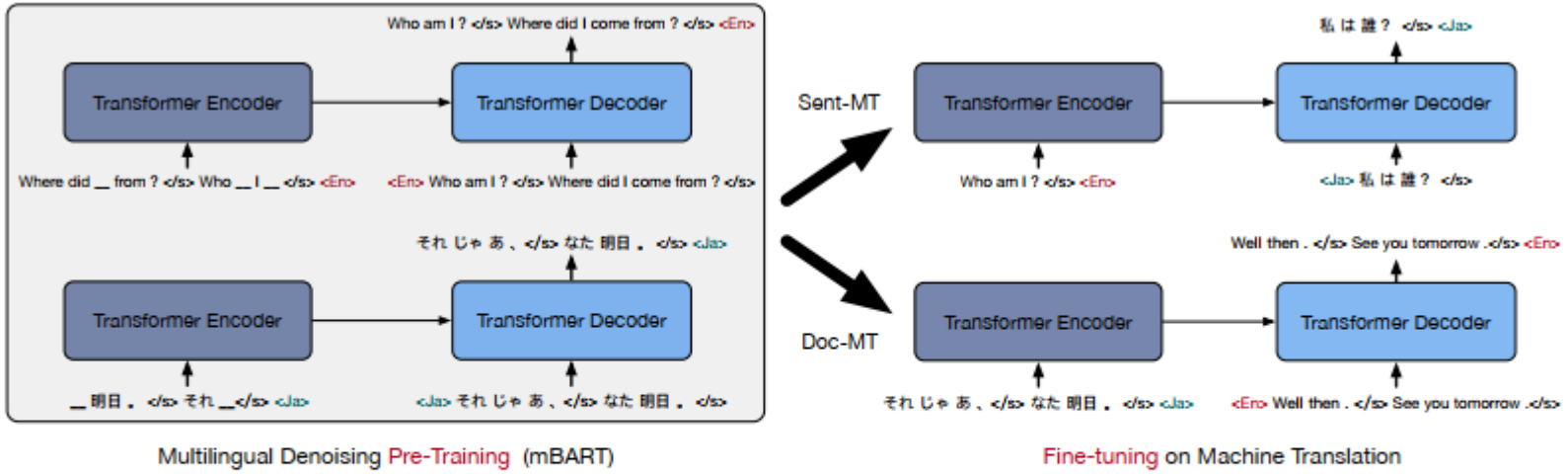


Figure 1: Framework for our Multilingual Denoising Pre-training (left) and fine-tuning on downstream MT tasks (right), where we use (1) sentence permutation (2) word-span masking as the injected noise. A special language id token is added at both the encoder and decoder. One multilingual pre-trained model is used for all tasks.

4. Multilingual neural machine translation (mBart)

- Architecture: Original Bart/Transformer
- Pre-train on multilingual datasets

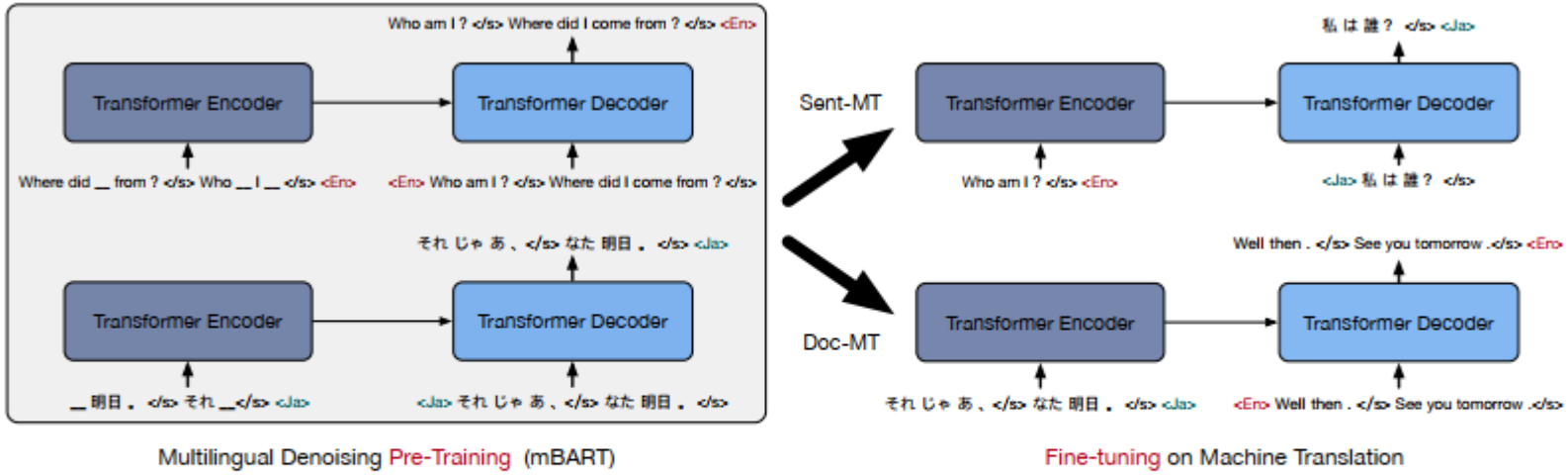


Figure 1: Framework for our Multilingual Denoising Pre-training (left) and fine-tuning on downstream MT tasks (right), where we use (1) sentence permutation (2) word-span masking as the injected noise. A special language id token is added at both the encoder and decoder. One multilingual pre-trained model is used for all tasks.

4. Multilingual neural machine translation (mBart (Liu et.al 2020))

- Architecture: Original Bart/Transformer
- Pre-train on multilingual datasets
- Finetune on bitext (Bilingual dataset)

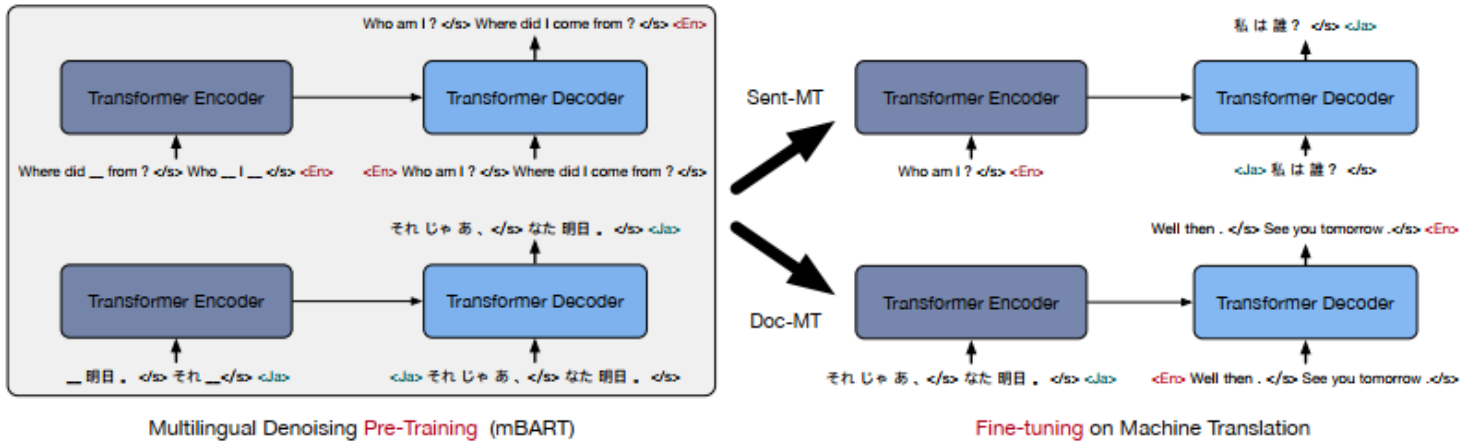


Figure 1: Framework for our Multilingual Denoising Pre-training (left) and fine-tuning on downstream MT tasks (right), where we use (1) sentence permutation (2) word-span masking as the injected noise. A special language id token is added at both the encoder and decoder. One multilingual pre-trained model is used for all tasks.

5. Multilingual neural machine translation (mBart-m2e, e2m,m2m (Tang et.al 2021))

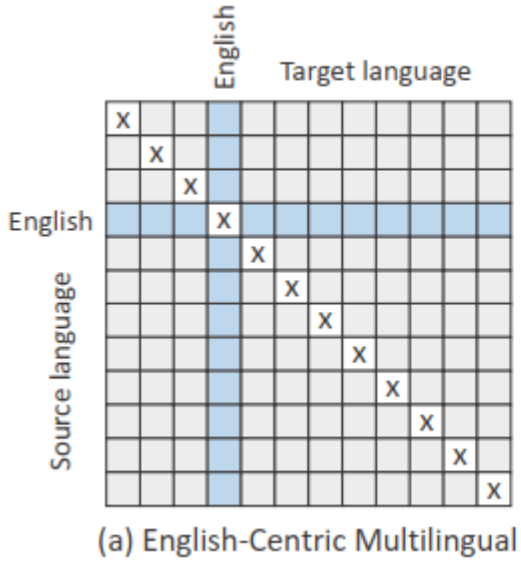
- Use the same architecture and multilingual pre-training task as mBart (Liu et.al 2020)), but conduct multilingual finetuning instead of finetuning on bitext

- 3 configurations to create different versions of multilingual translation models: *Many-to-one*, *one-to-Many*, and *Many-to-Many* via a pivot language
  - Many-to-one model encodes *N* languages and decodes to English
  - One-to-Many model encodes English and decodes into *N* languages.
  - Many-to-Many model encodes and decodes *N* languages using pivot data through English

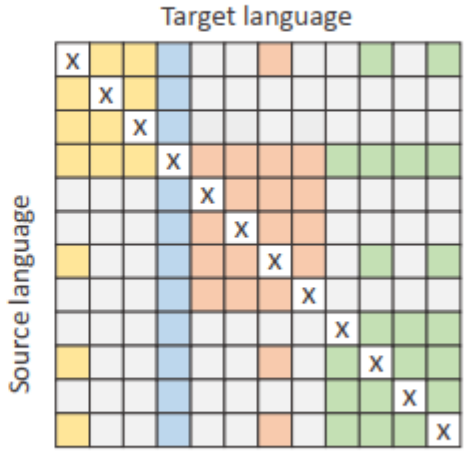
- Extend to incorporate a greater number of languages — 50 instead of 25
  - Build datasets with 25 additional languages
  - Build vocabulary and extend token embeddings for new tokens

4. Multilingual neural machine translation (M2M(Fan et.al 2021))

- Create a true Many-to-Many multilingual translation model that can translate directly between any pair of 100 languages
- Build multilingual datasets with 100 languages



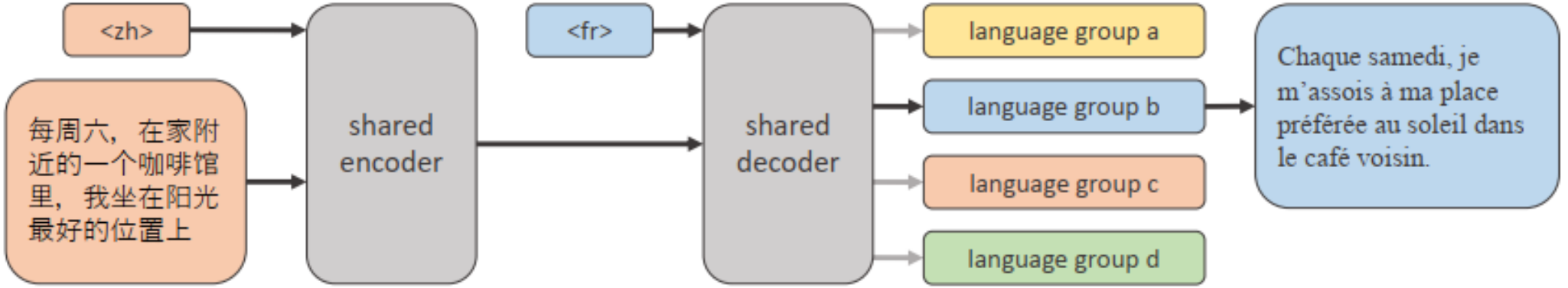
(a) English-Centric Multilingual



(b) M2M-100: Many-to-Many Multilingual Model

4. Multilingual neural machine translation (M2M(Fan et.al 2021))

- Explore how to effectively increase model capacity through a combination of dense scaling and language-specific parameters to create high quality models
- Some strategies to group language: **Frequency and Similarity**



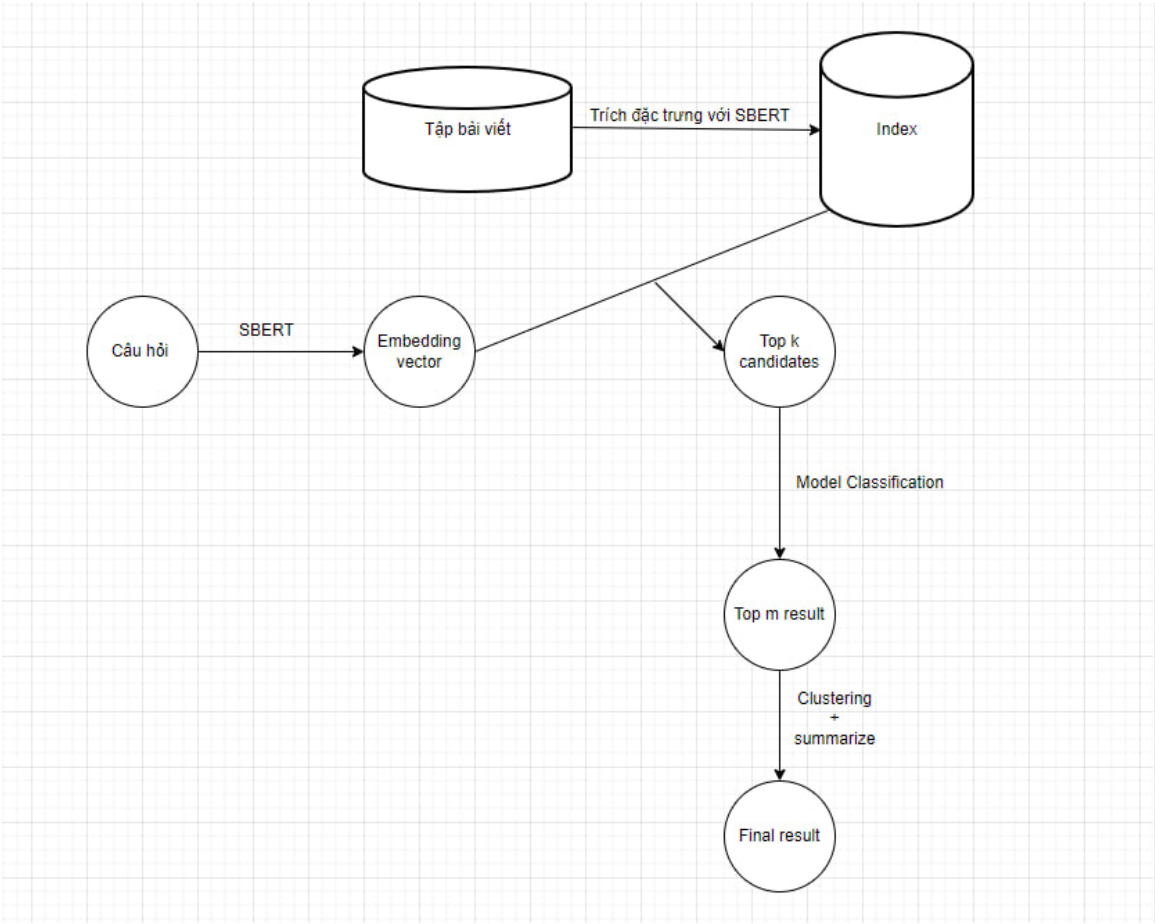
(c) Translating from Chinese to French with Dense + Language-Specific Sparse Model

Tài liệu tham khảo

- Attention Is All You Need. Vaswani et al. 2017.
- Improving Language Understanding by Generative Pre-Training. Radford et al. 2018.
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Devlin et al. 2018.
- Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Colin et all 2020
- BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension lewis et al.2020
- Multilingual Denoising Pre-training for Neural Machine Translation Liu et.al 2020
- Multilingual Translation with Extensible Multilingual Pretraining and Finetuning Tang.et.al 2021
- Beyond English-Centric Multilingual Machine Translation Fan et.al 2021

3. Hệ thống truy hồi thông tin dựa trên mô hình sinh ngôn ngữ

- Tổng quan kiến trúc



Retrieval

- Dữ liệu:

+ Nguồn crawl thường xuyên: mapping với câu hỏi dựa trên top relevant (sử dụng BM25)

+ Nguồn CSE

-> Thu được tập bài viết

Retrieval

- Xử lý dữ liệu:

+ Clean text

+ Tách thành các đoạn văn

+ Sử dụng SBERT để trích đặc trưng -> Lưu trữ (Elasticsearch)

Retrieval: Bi-encoder và Cross-encoder

- Tìm kiếm:

+ Câu hỏi đầu vào: sử dụng SBERT -> vector đặc trưng (bi-encoder)

+ Tìm kiếm những đoạn văn có độ tương đồng cao nhất với câu hỏi đầu vào dựa trên cosine_similarity giữa vector đặc trưng của các đoạn văn với vector đặc trưng của câu hỏi (tìm kiếm xấp xỉ HNSW)

-> Top k đoạn văn

+ Sử dụng mô hình phân loại (cross-encoder) xem đoạn văn có phù hợp với câu hỏi hay không -> Top m đoạn văn ($m \leq k$)

Module sinh văn bản:

- Module sinh văn bản:

+ Tạo data trả lời câu hỏi dựa trên context với chatGPT

+ Pretrained: BARTpho, Flan-T5

Kết luận

- ✗ Các mô hình NLP được sử dụng trong hệ gợi ý
- ✗ Các mô hình dịch đa ngôn ngữ
- ✗ Hệ thống truy vấn thông tin dựa trên mô hình sinh