Đại học Quốc gia Hà Nội

Đại học Khoa học Tự nhiên

THUYẾT TRÌNH CUỐI KỲ

Chủ đề: Summarize Paper

HỌC PHẦN

Nhập môn Trí Tuệ Nhân Tạo

Thành viên:

1. Phạm Bình Nghĩa
2. Phạm Đức Mạnh
3. Nguyễn Đình Phiên
4. Nguyễn Văn Tài
5. Mai Đình Thành

Các nội dung chính:

1. Động lực & Bối cảnh
2. Mục tiêu dự án
3. Các phương pháp tóm tắt truyền thống (Extractive)
4. Phương pháp tóm tắt sinh (Abstractive)
5. So sánh , Đánh giá kết quả thực nghiệm
6. Phương hướng phát triển

Phần 1: Động lực & Bối cảnh

1.Lý do chọn đề tài

- Lượng bài báo khoa học ngày càng lớn → khó đọc hết.
- Văn bản học thuật dài, phức tạp, chứa nhiều thông tin chuyên môn.
- Nhu cầu tự động tóm tắt để hỗ trợ nghiên cứu, học tập, giảng dạy.

2. Mục tiêu tổng quát:

- Tóm tắt lại ý chính các bài báo, văn bản

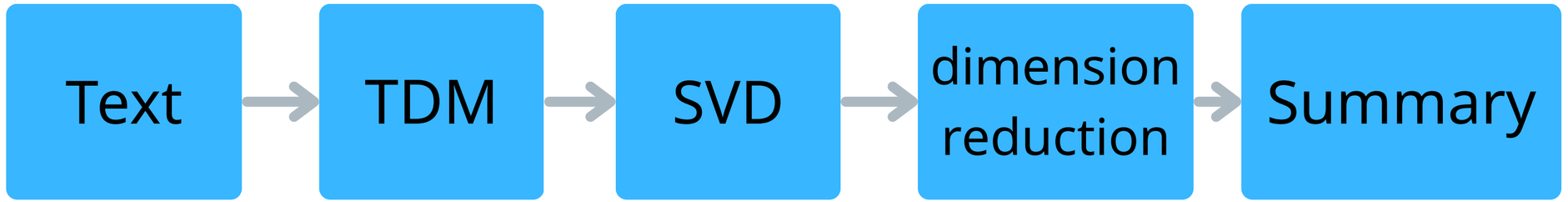
Phần 2: Mục tiêu dự án

- Xây dựng hệ thống tóm tắt bài báo khoa học tiếng Việt.
- Tích hợp hai hướng tiếp cận:
  - Tóm tắt trích rút (Extractive): TextRank, LSA, KL-Sum
  - Tóm tắt sinh (Abstractive): Fine-tune BARTpho
- So sánh, đánh giá hiệu quả giữa các phương pháp.
- Tìm ra mô hình cho chất lượng tóm tắt tối ưu nhất.

Phần 3: Phương pháp tóm tắt truyền thống

1. LSA (Latent Semantic Analysis)

- Latent Semantic Analysis (Phân tích ngữ nghĩa tiềm ẩn) là một kỹ thuật trong NLP được dùng để: Rút trích chủ đề (Topic); Tìm mức độ tương đồng giữa các văn bản; Tóm tắt văn bản; Giảm nhiễu (noise) trong dữ liệu văn bản; Giảm số chiều của không gian từ vựng.
- LSA dựa trên nền tảng chính là SVD (Singular Value Decomposition – phân rã giá trị kỳ dị).
- Lý thuyết "không gian ngữ nghĩa tiềm ẩn": Ý nghĩa thật sự của từ không nằm ở bản thân nó, mà nằm ở cách nó xuất hiện cùng với các từ khác trong nhiều văn bản.
- Quy trình:



1. LSA

1.1 Mô hình không gian vector (Ma trận TDM)

- Dữ liệu văn bản được biểu diễn dưới dạng ma trận Term-Document (TDM):
- Hàng: từ (terms)
- Cột: văn bản (documents)
- Ô: trọng số của từ trong văn bản (thường dùng TF-IDF thay vì TF vì TF-IDF loại bỏ từ phổ biến như “và”, “là”, ...)

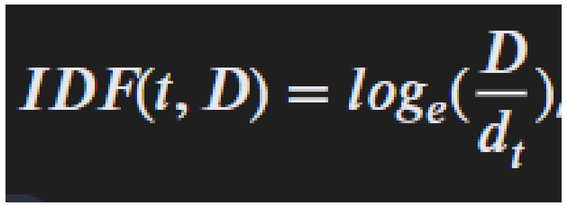
- D1 = "I like databases"
- D2 = "I dislike databases",

	I	like	dislike	databases
D1	1	1	0	1
D2	1	0	1	1

1. LSA

1.1 Mô hình không gian vector (Ma trận TDM)

- **TF (Tần suất thuật ngữ):** Số lần xuất hiện của một từ trong một tài liệu cụ thể.
- **IDF (Tần suất tài liệu nghịch đảo):** Một thước đo cho biết một từ xuất hiện trong bao nhiêu tài liệu trong một tập hợp.

Công thức IDF: 

trong đó D là tổng số tài liệu, dt là tổng số tài liệu chứa từ t.

- **TF-IDF:** Kết quả của việc nhân giá trị TF và IDF, cho biết mức độ quan trọng của một từ trong một tài liệu cụ thể so với toàn bộ tập hợp văn bản, dựa trên việc một từ xuất hiện nhiều trong tài liệu nhưng lại hiếm gặp trong các tài liệu khác.

1. LSA

1.2 Phân rã SVD (Singular Value Decomposition – phân rã giá trị kỳ dị)

- Phân rã giá trị kỳ dị là một phương pháp phân tích ma trận trong đó một ma trận A có thể được phân tích thành ba ma trận khác nhau:

$$\mathbf{A}_{m \times n} = \mathbf{U}_{m \times m} \mathbf{\Sigma}_{m \times n} (\mathbf{V}_{n \times n})^T$$

Trong đó:
U: biểu diễn các từ dưới dạng vector ngữ nghĩa
V: biểu diễn các văn bản
E: độ quan trọng của các chủ đề

1. LSA

1.3 Giảm chiều

- Ta chọn k thành phần quan trọng nhất:

$$A_k = U_k \Sigma_k V_k^T$$

- Ý nghĩa: các chủ đề yếu bị bỏ; các từ đồng nghĩa gom lại; không gian ngữ nghĩa mượt hơn, ít nhiễu.

1. LSA

1.4 Ứng dụng trong tóm tắt văn bản

Với tóm tắt:

- Mỗi câu được đưa vào ma trận TF-IDF
- LSA giúp tìm câu tiêu biểu cho chủ đề chính
- Chọn các câu có giá trị đóng góp lớn (hàng của Vk)

Cách đánh giá độ quan trọng của câu: $$score(c_i) = \sqrt{\sum_{j=1}^{k} (V_k[j,i])^2}$$

Câu nào có score cao nhất → quan trọng nhất → chọn vào summary.

1. LSA

1.5 Kết quả thực nghiệm

Mô hình LSA chỉ đạt mức trung bình–yếu:

- ROUGE-1: 0.3-0.5; ROUGE-2: thấp (< 0.15): cho thấy bản tóm tắt của LSA thường chỉ trùng lặp ở mức từ đơn, không biểu đạt được cụm từ hay ngữ đoạn đặc trưng của văn bản.
- Cosine similarity khá thấp (đa số < 0.25), phản ánh rằng dù LSA có chọn đúng một vài câu quan trọng, mức độ tương đồng ngữ nghĩa tổng thể giữa bản tóm tắt và bản gốc vẫn còn hạn chế.
- BERTScore lại khá cao, chứng tỏ văn bản tóm tắt khá chính xác với văn bản gốc.

Nguyên nhân chính nằm ở bản chất trích xuất của LSA: mô hình chỉ dựa trên phân rã không gian từ–câu, không hiểu ngữ nghĩa sâu, không nắm được quan hệ nhân quả hay mạch diễn đạt. Vì vậy, khi văn bản quá dài hoặc nhiều thông tin rẽ nhánh, LSA không thể "chọn đúng" các câu đại diện cho chủ đề trung tâm.

1. LSA

1.6 Ưu điểm và hạn chế

Ưu điểm của LSA

- Không cần huấn luyện, chạy nhanh, không yêu cầu GPU.
- Quy trình mô-đun (TF-IDF → SVD → chọn câu) dễ mở rộng và tùy chỉnh.
- Giảm nhiễu và rút trích chủ đề tốt nhờ SVD.
- Dễ giải thích, phù hợp nghiên cứu học thuật.

Hạn chế của LSA

- Phụ thuộc mạnh vào chất lượng tokenizer tiếng Việt.
- Chỉ tạo tóm tắt trích rút → đôi khi khô, kém tự nhiên.
- Khó xử lý văn bản dài hoặc nhiều chủ đề.
- Không hiểu ngữ cảnh sâu, đa nghĩa → thua kém mô hình học sâu hiện đại.

Phần 3: Phương pháp tóm tắt truyền thống

2. KL-Sum (Kullback-Leibler Sum)

- Thuật toán KL-Sum (Kullback–Leibler Sum) dựa trên việc lựa chọn các câu sao cho phân phối từ của bản tóm tắt gần nhất với phân phối từ của văn bản gốc. KL Divergence đo mức độ khác nhau giữa hai phân phối xác suất, qua đó giúp mô hình chọn được các câu đại diện tốt nhất cho nội dung chính.
- Nguyên lý hoạt động: KL_Sum chọn câu mà khi thêm vào bản tóm tắt sẽ giảm KL Divergence nhiều nhất, tức là làm cho phân phối từ bản tóm tắt gần nhất với phối từ văn bản gốc nhất. Quá trình lặp lại cho đến khi đạt số câu tối đa hoặc không còn cải thiện đáng kể.

2. KL-Sum

2.1 Kullback-Leibler Divergence

- **Kullback–Leibler Divergence (KL Divergence):** đo mức độ khác biệt giữa hai phân phối xác suất. Trong tóm tắt văn bản, KL Divergence đo khoảng cách giữa phân phối từ của bản tóm tắt và phân phối từ của toàn bộ văn bản gốc.
- Công thức:

$$D_{KL}(P||Q) = \sum_{w \in V} P(w) \log \frac{P(w)}{Q(w)}$$

Trong đó:

- P là phân phối từ văn bản gốc,
- Q là phân phối từ trong văn bản tóm tắt hiện tại
- V là tập từ vựng.

2. KL-Sum

2.2 Phân phối unigram

- Phân phối unigram là xác suất xuất hiện của từng từ trong văn bản, thường được tính với Laplace smoothing để tránh xác suất bằng 0.
- Công thức:

$$P(w) = \frac{count(w)+\alpha}{N+\alpha|V|}$$

Trong đó:

- Count(w) là số từ w xuất hiện
- N là tổng số từ
- |V| kích thước từ vựng
- α là hệ số làm mượt.

2. KL-Sum

2.3 Kết quả thực nghiệm

- Kết quả thực nghiệm trên toàn bộ tập dữ liệu cho thấy mô hình tóm tắt đạt chất lượng ổn định và đáng tin cậy
- ROUGE-1 đạt 0.5972 cho thấy mức độ bao phủ từ vựng quan trọng khá cao
- ROUGE-2 ở mức 0.5286 chứng tỏ mô hình giữ được mạch ý và cụm từ then chốt
- ROUGE-L đạt 0.5196 khẳng định sự tương đồng về cấu trúc và trật tự thông tin giữa bản gốc và bản tóm tắt.
- BERTScore F1 = 0.7565 cho thấy mô hình không chỉ dựa vào trùng lặp bề mặt mà còn nắm bắt được nội dung sâu của văn bản.
- Cosine Similarity đạt 0.8896 khẳng định bản tóm tắt duy trì rất tốt chủ đề và tinh thần chung của toàn văn bản.

→ Tổng hợp các kết quả, mô hình cho thấy hiệu năng mạnh trên cả hai khía cạnh: **độ chính xác từ vựng** và **độ tương đồng ngữ nghĩa**.

2. KL-Sum

2.4 Ưu điểm & hạn chế

a.Ưu điểm

- Giữ được tính bao quát chủ đề: do KL_Sum chọn câu dựa trên phân phối từ vựng chỉ chọn từ tồn tại trong văn bản chọn cho nên bản tóm tắt thường đầy đủ ý chính.
- Tránh trùng lặp thông tin: KL Divergence làm giảm nội dung trùng lặp trong bản tóm tắt
- Dễ triển khai: không cần mô hình huấn luyện không cần dữ liệu lớn chỉ cần xử lí từ, tính phân phối rồi chọn câu
- Hoạt động tốt với văn bản dài: văn bản càng dài phân phối từ càng ổn định, KL_Sum càng chính xác.

2. KL-Sum

2.4 Ưu điểm & hạn chế

b.Hạn chế

- Không hiểu ngôn ngữ sâu: chỉ đo xem xuất hiện bao nhiêu lần chứ không hiểu nghĩa hay ngữ cảnh.
- Không biết tạo tóm tắt trôi chảy: do chỉ chọn câu vây nên dễ tạo thành danh sách câu hơn là đoạn văn.
- Dễ chọn phải câu dài quá mức: câu dài chứa nhiều từ cho nên phân phối giống dễ được chọn nhiều hơn nhưng đôi khi không phải câu tốt nhất

Phần 3: Phương pháp tóm tắt truyền thống

3. TextRank

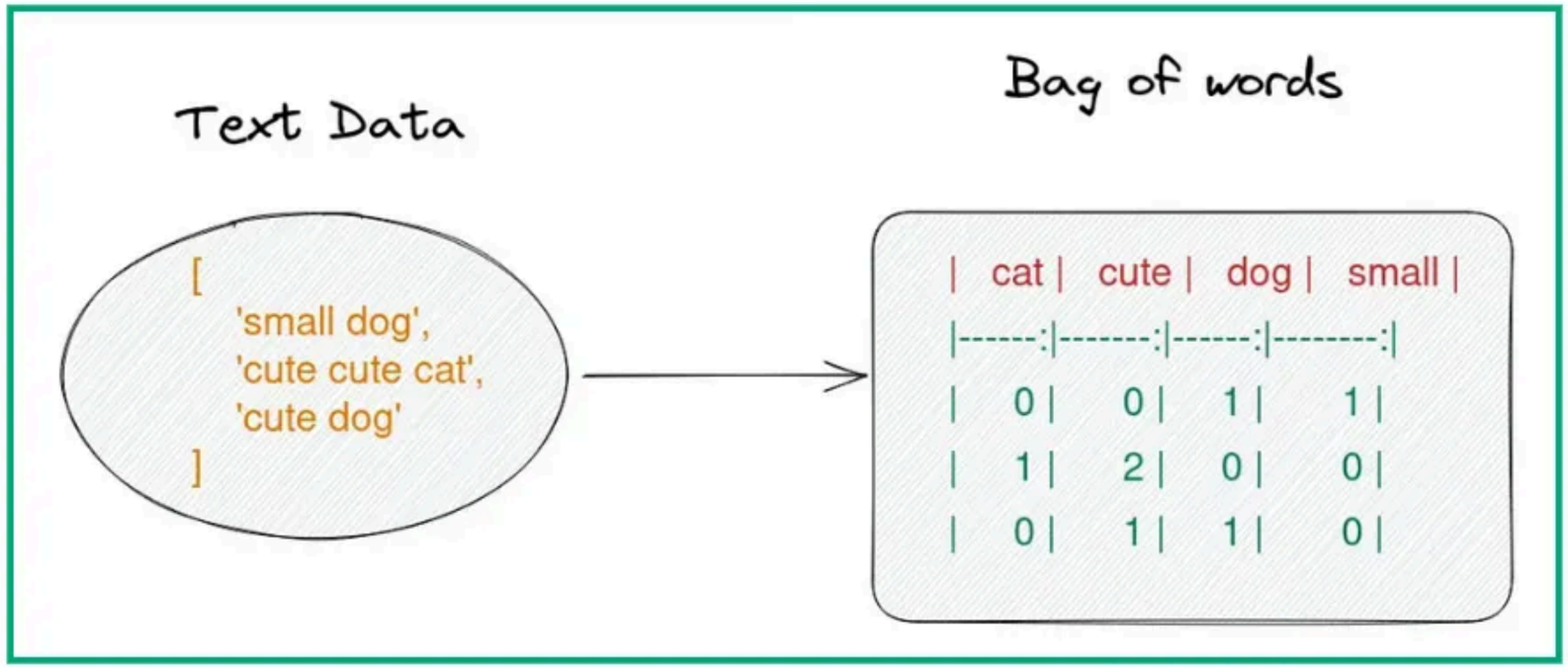
- TextRank là một thuật toán trích xuất thông tin dựa trên đồ thị (graph-based ranking) được đề xuất bởi Rada Mihalcea và Paul Tarau vào năm 2004. Thuật toán này được lấy ý tưởng từ PageRank – phương pháp xếp hạng trang web của Google.
- Mục tiêu của TextRank là:
  - Xác định mức độ quan trọng của các đơn vị ngôn ngữ (từ, câu)
  - Sử dụng chúng để tóm tắt văn bản hoặc trích xuất từ khóa
- Thuật toán gồm 4 bước chính:
  - Biểu diễn văn bản thành đồ thị
  - Tính độ tương đồng giữa các câu (Cosine Similarity)
  - Áp dụng công thức PageRank để tính điểm quan trọng
  - Chọn ra các câu quan trọng nhất

3. TextRank

3.1 Biểu diễn câu thành vector (Sentence Vectorization)

Mỗi câu được biểu diễn bằng vector theo mô hình Bag-of-Words:

- Xây dựng vocabulary gồm tất cả các từ xuất hiện trong văn bản
- Với mỗi câu, tạo vector trong đó:
  - Phần tử = số lần từ xuất hiện trong câu
- Các vector này được dùng để tính độ tương đồng câu – câu



3. TextRank

3.2 Tính độ tương đồng giữa các câu (Cosine Similarity)

- Độ tương đồng giữa hai câu được tính bằng cosine similarity
- Kết quả giúp đánh giá mức độ liên quan về nội dung giữa hai câu

$$\text{cosine}(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||}$$

3. TextRank

3.3 Tính điểm quan trọng bằng PageRank

- Công thức PageRank:

$$S(V_i) = (1 - d) + d \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_k w_{jk}} S(V_j)$$

Trong đó:

- V_i: câu cần tính điểm
- $In(V_i)$: tập các câu có liên kết tới V_i
- w_{ji}: độ tương đồng giữa câu j và câu i
- d: hệ số suy giảm, thường = 0.85
- $S(V_i)$: điểm quan trọng của câu i

Ý nghĩa trực quan: Một câu càng được nhiều câu quan trọng khác ủng hộ (liên kết tới), thì nó càng quan trọng. Điểm số sẽ hội tụ sau vài vòng lặp.

3. TextRank

3.4 Kết quả thực nghiệm

- Ba chỉ số ROUGE đều đạt mức trung bình khá, đặc biệt ROUGE-1 và ROUGE-2 lần lượt khoảng 0.57 và 0.54. Điều này cho thấy TextRank khôi phục được phần lớn thông tin chính xuất hiện trong bản tóm tắt chuẩn. Việc ROUGE-2 đạt giá trị tương đối cao là dấu hiệu cho thấy các câu được mô hình chọn chứa nhiều cụm từ quan trọng.
- ROUGE-L chỉ đạt 0.45 phản ánh một hạn chế cố hữu của phương pháp trích xuất: mô hình chưa tái tạo được cấu trúc liền mạch của bản tóm tắt chuẩn. Điều này là hợp lý vì TextRank chỉ lựa chọn các câu theo thứ hạng mà không thực hiện tổ chức lại bố cục.
- Điểm BERTScore F1 đạt 0.7953, cao hơn đáng kể so với các chỉ số ROUGE. Điều này chứng minh rằng, dù TextRank không sinh câu mới, nội dung tóm tắt vẫn duy trì được ý nghĩa cốt lõi so với tóm tắt chuẩn.
- Giá trị Cosine Similarity là 0.7394, tương đương mức độ tương đồng tương đối cao về mặt phân bố từ khóa. Chỉ số này chứng minh rằng các bản tóm tắt tạo bởi TextRank chứa phần lớn từ khóa quan trọng tương tự bản chuẩn.

3. TextRank

3.4 Ưu điểm và hạn chế

Ưu điểm:

- Không phụ thuộc vào dữ liệu huấn luyện (unsupervised)
- Dễ triển khai và chi phí tính toán thấp
- Khả năng áp dụng linh hoạt: TextRank có thể áp dụng cho nhiều ngôn ngữ mà không cần thay đổi cấu trúc thuật toán. Ngoài ra, phần tính độ tương đồng có thể được thay thế bằng: TF-IDF, word embedding, sentence embedding

Hạn chế:

- Không hiểu ngữ nghĩa sâu: TextRank dựa trên thông tin bề mặt (tần suất từ, vector đơn giản, độ tương đồng), nên không hiểu được ngữ nghĩa sâu hoặc nội dung suy luận.
- Phụ thuộc mạnh vào chất lượng tách câu: Nếu bước tách câu sai hoặc văn bản chứa câu quá dài, TextRank sẽ đánh giá mối quan hệ sai lệch, làm giảm chất lượng tóm tắt.
- Không phù hợp với văn bản rất dài

Phần 4: Phương pháp tóm tắt sinh (Abstractive)

1.BARTpho (Bidirectional and Auto-Regressive Transformers phở)

- BartPho là một mô hình tóm tắt tiếng Việt dựa trên kiến trúc BART
- BART là một bộ tự mã hóa khử nhiễu (denoising autoencoder) được xây dựng dưới dạng mô hình Seq2Seq. Nó tích hợp Transformer hai chiều (Bidirectional, giống như bộ mã hóa của BERT) và Transformer tự hồi quy (Auto-Regressive, trái sang phải, giống như bộ giải mã của GPT), cho phép mô hình hiểu văn bản đầu vào và sinh ra bản tóm tắt ngắn gọn, tự nhiên.
- Kiến trúc: Cả hai phiên bản của BARTpho đều sử dụng kiến trúc "lớn" ("large" architecture), với 12 lớp (layer) trong cả bộ mã hóa và bộ giải mã.
- Chức năng kích hoạt: BARTpho sử dụng hàm kích hoạt GeLU (Gaussian Error Linear Units) thay vì ReLU và khởi tạo tham số từ N(0,0.02).
- Chuẩn hóa: BARTpho cũng bổ sung một lớp chuẩn hóa lớp (layer-normalization) trên cả bộ mã hóa và bộ giải mã, theo mô hình mBART.

1. BARTpho

1.1 Các phiên bản BARTpho

Phiên bản	Kiểu đầu vào	Số lượng tham số	Tokenizer/Từ vựng	Tính hiệu quả
BARTphosyllable	Cấp độ âm tiết (syllable level)	Khoảng 396M	Sử dụng mô hình SentencePiece từ XLM-RoBERTa, với 40K loại phụ âm tiết thường xuyên nhất.	Hiệu quả hơn mBART.
BARTphoword	Cấp độ từ (word level), đã được tự động phân tách từ	Khoảng 420M	Sử dụng tokenizer của PhoBERT, với 64K loại subword và BPE.	Hiệu quả nhất, vượt trội so với BARTphosyllable và mBART.

1. BARTpho

1.2 Cơ sở lý thuyết

Cơ sở lý thuyết của mô hình BART (Bidirectional and Auto-Regressive Transformers) nằm ở việc nó là một bộ tự mã hóa khử nhiễu (denoising autoencoder) được huấn luyện trước (pre-trained) theo kiến trúc chuỗi-tới-chuỗi (Sequence-to-Sequence - Seq2Seq).

BART được thiết kế để cung cấp một phương pháp huấn luyện tự giám sát tổng quát và linh hoạt hơn so với các mô hình trước đây, cho phép nó xử lý hiệu quả cả các tác vụ tạo sinh ngôn ngữ (generation) và hiểu ngôn ngữ (comprehension)

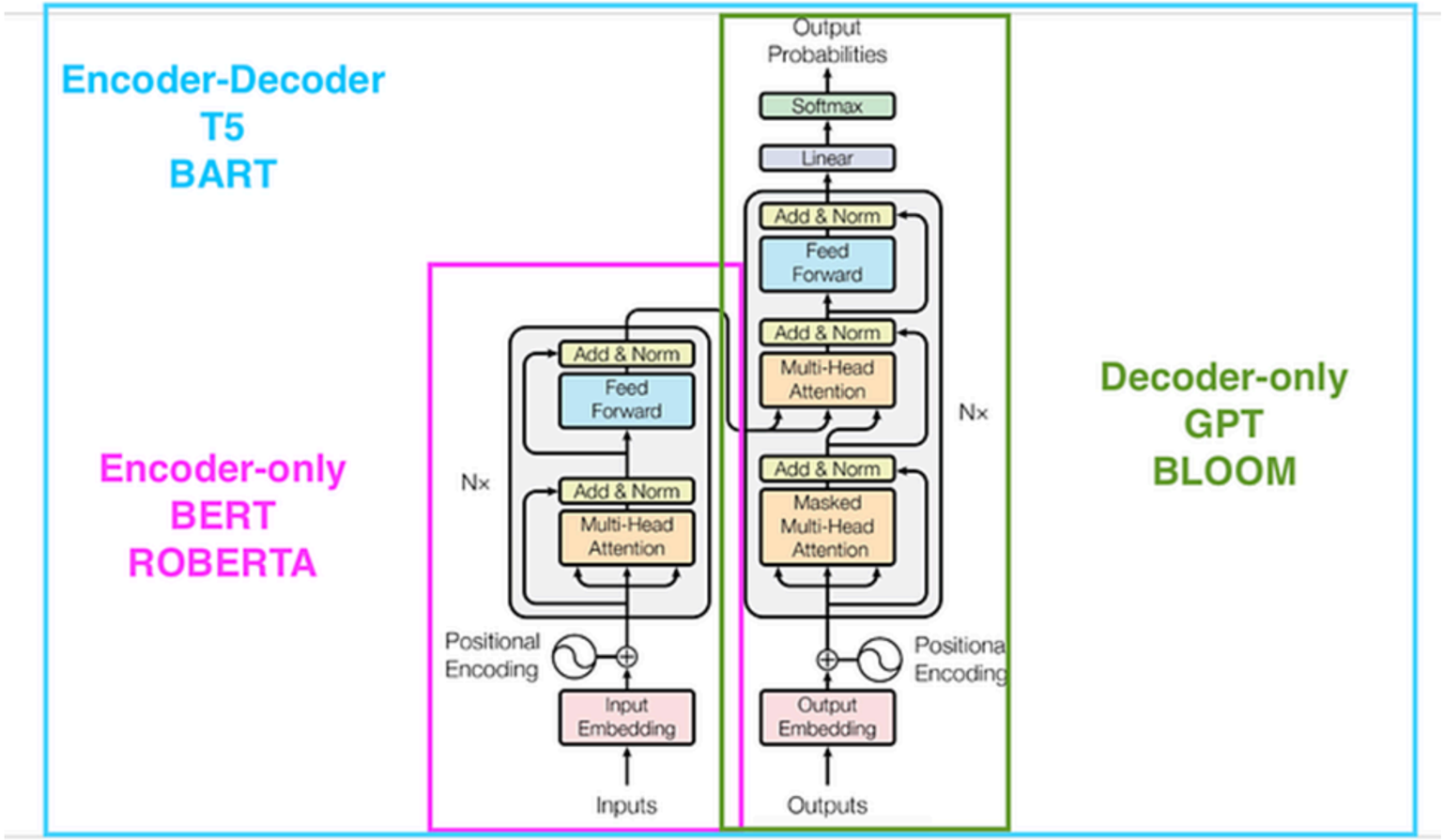
1.2 Cơ sở lý thuyết BARTpho

1.2.1 Kiến trúc Seq2Seq Khái quát hóa

- Bộ Mã hóa Hai chiều (Bidirectional Encoder):
  - Bộ mã hóa của BART tương đương với BERT (Bidirectional Encoder Representations from Transformers).
  - Nhiệm vụ của nó là mã hóa đầu vào bị làm nhiễu loạn bằng cách điều kiện hóa đồng thời trên ngữ cảnh cả bên trái và bên phải ở tất cả các lớp. Khả năng hai chiều này là cần thiết cho các tác vụ hiểu văn bản.
- Bộ Giải mã Tự hồi quy (Auto-Regressive Decoder):
  - Bộ giải mã của BART hoạt động theo cơ chế tự hồi quy từ trái sang phải (left-to-right autoregressive), tương tự như GPT.
  - Nhiệm vụ của nó là tái tạo lại tài liệu gốc từ đầu ra của bộ mã hóa.
  - Đáng chú ý, mỗi lớp của bộ giải mã thực hiện cơ chế chú ý chéo (cross-attention) trên lớp ẩn cuối cùng của bộ mã hóa.

1.2 Cơ sở lý thuyết BARTpho

1.2.1 Kiến trúc Seq2Seq Khái quát hóa



- Sự kết hợp này cho phép BART linh hoạt hơn: bộ mã hóa hiểu ngữ cảnh sâu rộng, còn bộ giải mã tạo điều kiện cho việc tinh chỉnh trực tiếp (direct fine-tuning) trên các tác vụ tạo sinh chuỗi.

1.2 Cơ sở lý thuyết BARTpho

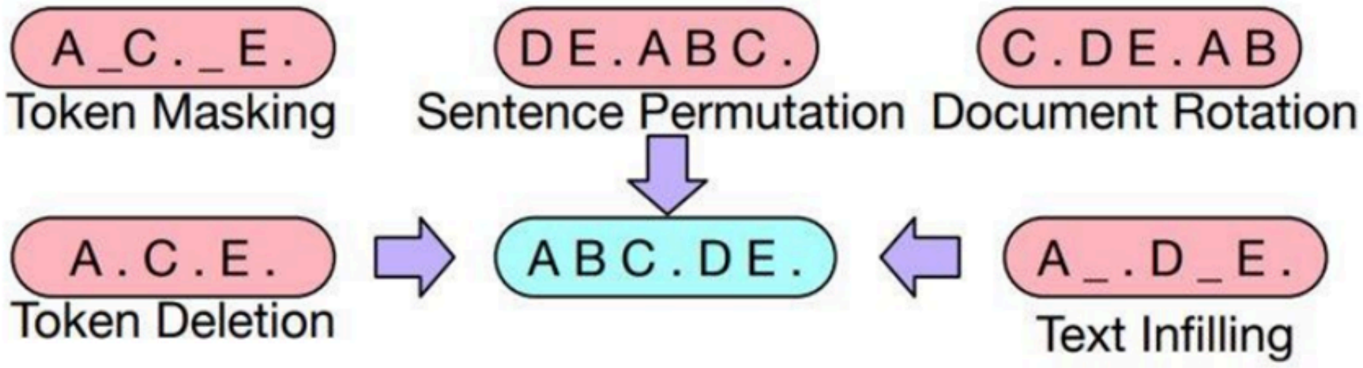
1.2.2 Phương pháp Huấn luyện trước Khử nhiễu

- Cơ sở lý thuyết cốt lõi của BART là một bộ tự mã hóa khử nhiễu được tối ưu hóa thông qua tổn thất tái tạo (reconstruction loss):
  - Làm nhiễu Văn bản (Text Corruption): Văn bản đầu vào được làm hỏng bằng cách áp dụng một hàm gây nhiễu tùy ý (arbitrary noising function). BART được thiết kế để linh hoạt với nhiều loại nhiễu khác nhau, ngay cả khi chúng thay đổi độ dài của chuỗi.
  - Tối ưu hóa Tổn thất Tái tạo (Reconstruction Loss): Mô hình Seq2Seq được huấn luyện để khôi phục lại văn bản gốc, bằng cách tối ưu hóa tổn thất entropy chéo (cross-entropy) giữa đầu ra của bộ giải mã và tài liệu gốc ban đầu.

1.2 Cơ sở lý thuyết BARTpho

1.2.2 Phương pháp Huấn luyện trước Khử nhiễu

- Các thuật toán làm nhiễu (noising functions) là chìa khóa để đạt được sự linh hoạt và khả năng sinh văn bản của BART:
  - Điền khuyết Văn bản (Text Infilling): Kỹ thuật này dạy mô hình phải dự đoán số lượng token bị thiếu trong một khoảng trống, giúp mô hình lý luận về độ dài chuỗi tổng thể.
  - Hoán vị Câu (Sentence Permutation): Tài liệu được chia thành các câu và các câu này bị xáo trộn theo thứ tự ngẫu nhiên. Kỹ thuật này buộc mô hình phải học cách sắp xếp các đơn vị ngôn ngữ lớn hơn (câu) một cách hợp lý.
  - Các kĩ thuật gây nhiễu khác: Che từ ngẫu nhiên (Token Masking), Xóa từ (Token Deletion), Xoay tài liệu (Document Rotation)



1.2 Cơ sở lý thuyết BARTpho

1.2.3 Các Thuật toán Cấu tạo (Attention and Activation)

Kiến trúc Transformer trong BART sử dụng các cơ chế sau:

- Cơ chế Tự chú ý (Self-Attention) và Chú ý Đa đầu (Multi-head Attention): Đây là thuật toán cơ bản của Transformer được sử dụng trong cả Encoder và Decoder để tính toán mối quan hệ ngữ cảnh giữa các token trong chuỗi.
- Cơ chế Chú ý Chéo (Cross-Attention): Bộ giải mã (Decoder) sử dụng cơ chế này để chú ý (attend) đến các biểu diễn trạng thái ẩn (hidden states) được tạo ra bởi Bộ mã hóa (Encoder), đảm bảo rằng đầu ra được tạo ra phù hợp với ngữ cảnh đầu vào.
- Hàm Kích hoạt (Activation Function): BART sử dụng hàm kích hoạt Gaussian Error Linear Unit (GeLU) thay vì hàm ReLU được sử dụng trong Transformer gốc.

1. BARTpho

1.3 Kết quả thực nghiệm

- ROUGE-1 = 0.6456: Mức trùng lặp từ đơn cao, chứng tỏ mô hình giữ lại được nhiều từ khóa quan trọng.
- ROUGE-2 = 0.4110: Khả năng bảo toàn các cụm từ ngắn (bigrams) ở mức tốt, giúp tóm tắt có tính mạch lạc.
- ROUGE-3 = 0.4224: Phản ánh mô hình duy trì được cấu trúc câu và trật tự thông tin tương đối tốt.
- BERTScore F1 = 0.7326 cho thấy mức độ tương đồng ngữ nghĩa cao giữa bản tóm tắt sinh và bản tóm tắt chuẩn. Điểm số >0.70 chứng tỏ mô hình không chỉ sao chép từ vựng mà còn hiểu được nội dung và sinh câu có ý nghĩa gần với bản gốc.
- Cosine Similarity = 0.6128: Mức độ tương đồng nội dung giữa văn bản gốc và tóm tắt được giữ ở mức trung bình – khá. Giá trị ≈0.6 cho thấy tóm tắt vẫn bao phủ phần lõi thông tin của văn bản đầu vào, nhưng không quá trùng lặp — điều này phù hợp với đặc trưng của tóm tắt abstractive.

1. BARTpho

1.4 Ưu điểm và hạn chế

Ưu điểm

- Tóm tắt sinh (abstractive) nên có thể viết lại câu, tổng hợp ý và tạo văn bản mượt mà, tự nhiên hơn các phương pháp trích rút.
- Hiểu ngữ nghĩa tốt
- Linh hoạt: có thể áp dụng cho nhiều loại văn bản (tin tức, học thuật, pháp lý, mô tả).
- Hỗ trợ fine-tune: dễ điều chỉnh cho các miền dữ liệu cụ thể để tăng hiệu năng.

Hạn chế:

- Phụ thuộc dữ liệu huấn luyện
- Yêu cầu tài nguyên tính toán cao
- Có thể sinh thông tin không có trong văn bản (hiện tượng "hallucination"), đặc biệt với văn bản dài hoặc phức tạp.
- Quá trình triển khai phức tạp: cần tokenizer, pipeline và tối ưu tham số kỹ lưỡng để đạt chất lượng tốt

Phần 5: Kết quả thực nghiệm và so sánh

5.1 Phương pháp đánh giá

5.1.1 Đánh giá dựa trên trùng khớp từ vựng – ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

- ROUGE-1: đo mức trùng lặp từ đơn giữa bản tóm tắt mô hình sinh và bản tóm tắt tham chiếu → Phản ánh khả năng giữ lại các từ khóa quan trọng.
- ROUGE-2: đo mức trùng lặp bigram (2 từ liên tiếp) → Phản ánh mức độ bảo toàn các cụm từ có ý nghĩa cục bộ.
- ROUGE-L: đo độ dài chuỗi con chung dài nhất → Đánh giá mức độ mạch lạc tổng thể của câu sinh ra.
- Ưu điểm: đơn giản, phổ biến, dễ so sánh.
- Hạn chế: không đo được ý nghĩa sâu, nhạy với sự thay đổi từ ngữ không quan trọng.

Phần 5: Kết quả thực nghiệm và so sánh

5.1 Phương pháp đánh giá

5.1.2 Đánh giá dựa trên ngữ nghĩa – BERTScore

- BERTScore sử dụng embedding ngữ nghĩa từ mô hình ngôn ngữ (như BERT, mBERT hoặc PhoBERT) để so sánh từng token trong hai câu dựa trên cosine similarity.
    - Đầu ra gồm Precision, Recall và F1, trong đó F1 được dùng để đánh giá tổng quát.
    - Dùng tốt cho tóm tắt vì hai câu có thể khác từ ngữ nhưng vẫn giữ nghĩa tương đồng.
- Ưu điểm: bắt được ngữ nghĩa sâu, đánh giá đúng các câu diễn đạt lại (paraphrasing).
- Hạn chế: tính toán nặng, phụ thuộc vào chất lượng mô hình embedding.

Phần 5: Kết quả thực nghiệm và so sánh

5.1 Phương pháp đánh giá

5.1.3 Đánh giá mức độ bao phủ nội dung – Cosine Similarity (TF-IDF)

- Phương pháp này chuyển văn bản gốc và tóm tắt mô hình sinh thành vector TF-IDF, sau đó tính độ tương đồng giữa hai vector:
  - Cho biết tóm tắt có giữ được phần nội dung nào trong văn bản gốc hay không.
  - Giá trị dao động từ 0 đến 1, càng cao càng giống nhau.
- Ưu điểm: đánh giá dựa trên "vùng nội dung", không cần bản tóm tắt tham chiếu.
- Hạn chế: chỉ phản ánh mức trùng lặp từ theo trọng số, không đại diện ngữ nghĩa sâu.

Phần 5: Kết quả thực nghiệm và so sánh

5.2 So sánh

PP đánh giá	LSA	KL-Sum	TextRank	BARTpho
ROUGE-1	0.4	0.5972	0.5717	0.6456
ROUGE-2	Nhỏ hơn 0.25	0.5268	0.5390	0.4110
ROUGE-L	0.334	0.5196	0.4528	0.4224
BERTScore	0.7	0.7565	0.7953	0.7326
Cosine Similarity	0.483	0.8896	0.7394	0.6128

Mô hình	Điểm mạnh	Điểm yếu	Khi nào nên dùng
LSA	Dễ triển khai, chạy nhanh	Chất lượng kém nhất, không giữ ngữ nghĩa	Chỉ dùng cho bài tập minh hoạ thuật toán
TextRank	BERTScore cao → giữ nghĩa tốt, ổn định	ROUGE-L không cao nhất	Tóm tắt tin tức, văn bản ngắn
KL-Sum	Cosine cao nhất, ROUGE rất tốt	Dễ trùng lặp, hơi "thô"	Tóm tắt trích rút **chính xác nội dung**, báo cáo kỹ thuật
BARTpho	ROUGE-1 cao nhất, văn phong tự nhiên	Phụ thuộc fine-tune, ROUGE-2 thấp	Tạo **tóm tắt sinh tự nhiên**, văn phong mượt, giống người viết

Phần 5: Kết quả thực nghiệm và so sánh

5.2 So sánh

- BARTpho (tóm tắt NGẮN – GIỮ NGHĨA – TỰ NHIÊN):
  - Diễn đạt lại câu → không copy y nguyên
  - Tóm tắt mượt, dễ đọc
  - Phù hợp làm ứng dụng Web
- KL-Sum (tóm tắt TRÍCH RÚT – CHÍNH XÁC – GIỐNG NGUYÊN BẢN):
  - Bám sát nội dung gốc nhất
  - Dùng cho báo cáo khoa học, văn bản kỹ thuật
- TextRank (trích rút đơn giản – giữ nghĩa tốt nhất):
  - Hiệu quả với tin tức, văn bản trung bình

Phần 6: Phương hướng phát triển

Mở rộng và nâng cao chất lượng dữ liệu huấn luyện

- Thu thập thêm tập dữ liệu văn bản tiếng Việt lớn và đa dạng hơn (tin tức, khoa học, báo cáo, pháp luật...).
- Làm sạch dữ liệu tốt hơn: loại nhiễu, chuẩn hóa dấu câu, tách câu.
- Tăng kích thước dữ liệu fine-tune cho BARTpho để mô hình tạo bản tóm tắt mượt và chính xác hơn.

Kết hợp nhiều mô hình (Hybrid Summarization)

- Phát triển mô hình kết hợp trích rút + sinh:
  - Dùng TextRank hoặc KL-Sum chọn câu quan trọng.
  - Dùng BARTpho để diễn đạt lại những câu đó thành bản tóm tắt tự nhiên hơn.
  - Giải pháp này vừa đảm bảo độ chính xác, vừa có tính mượt mà.

THE END