

TITLE GENERATION

SCIENTIFIC

(SPRINGER)

SV thực hiện

**KHƯU THÀNH THIỆN
HUỲNH THIÊN THUẬN
HUỲNH NHẬT NAM**

GV hướng dẫn

LÊ THANH TÙNG



MỤC LỤC

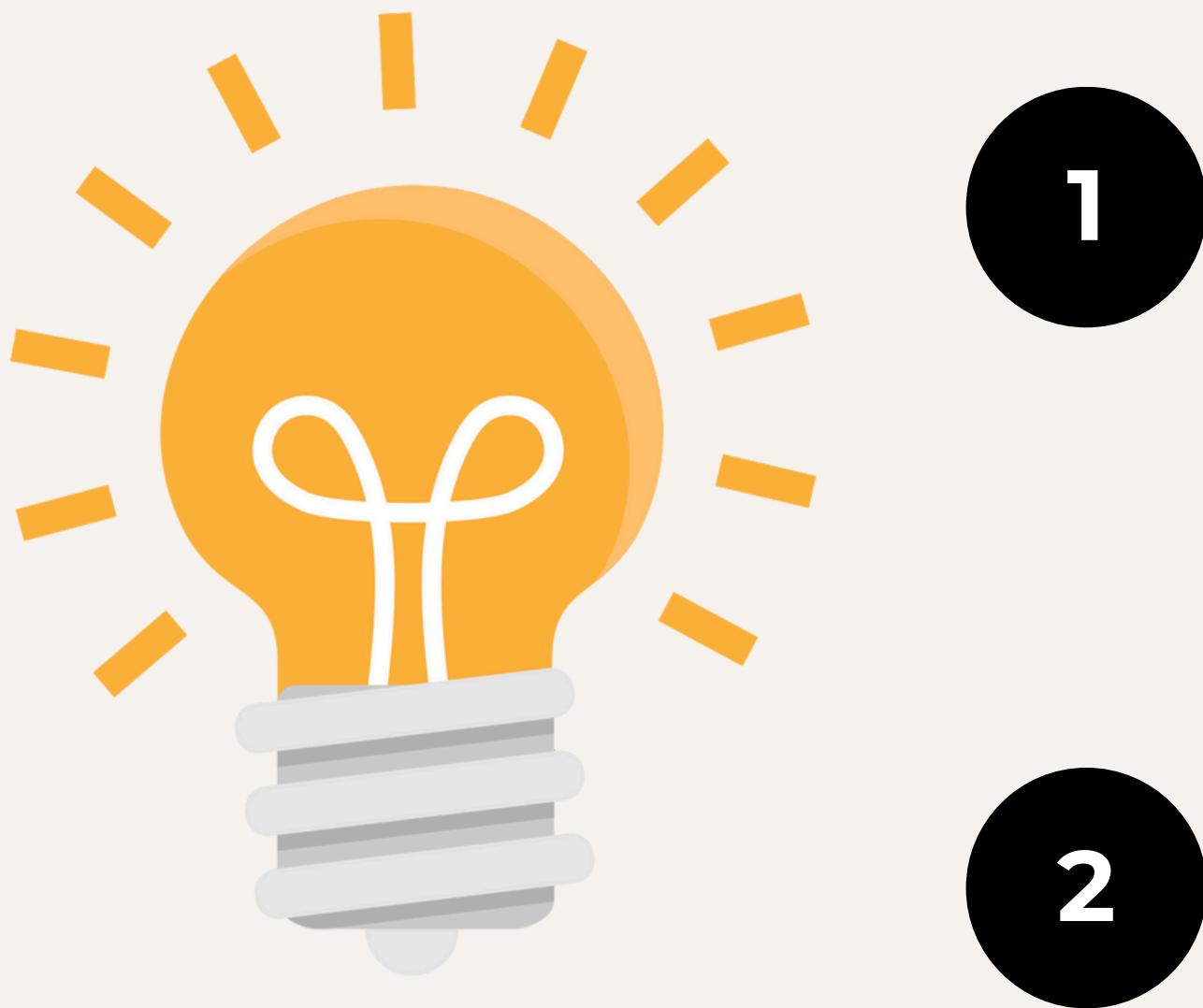
- 01 **Bài toán**
- 02 **Input/Output**
- 03 **Data source**
- 04 **Phương pháp đánh giá**
- 05 **Mô hình**
- 06 **Kết quả model**
- 07 **LLM Fine-tune**
- 08 **Cải tiến model**
- 09 **Đánh giá model**
- 10 **Deploy**

01

BÀI TOÁN



Ý TƯỞNG BÀI TOÁN

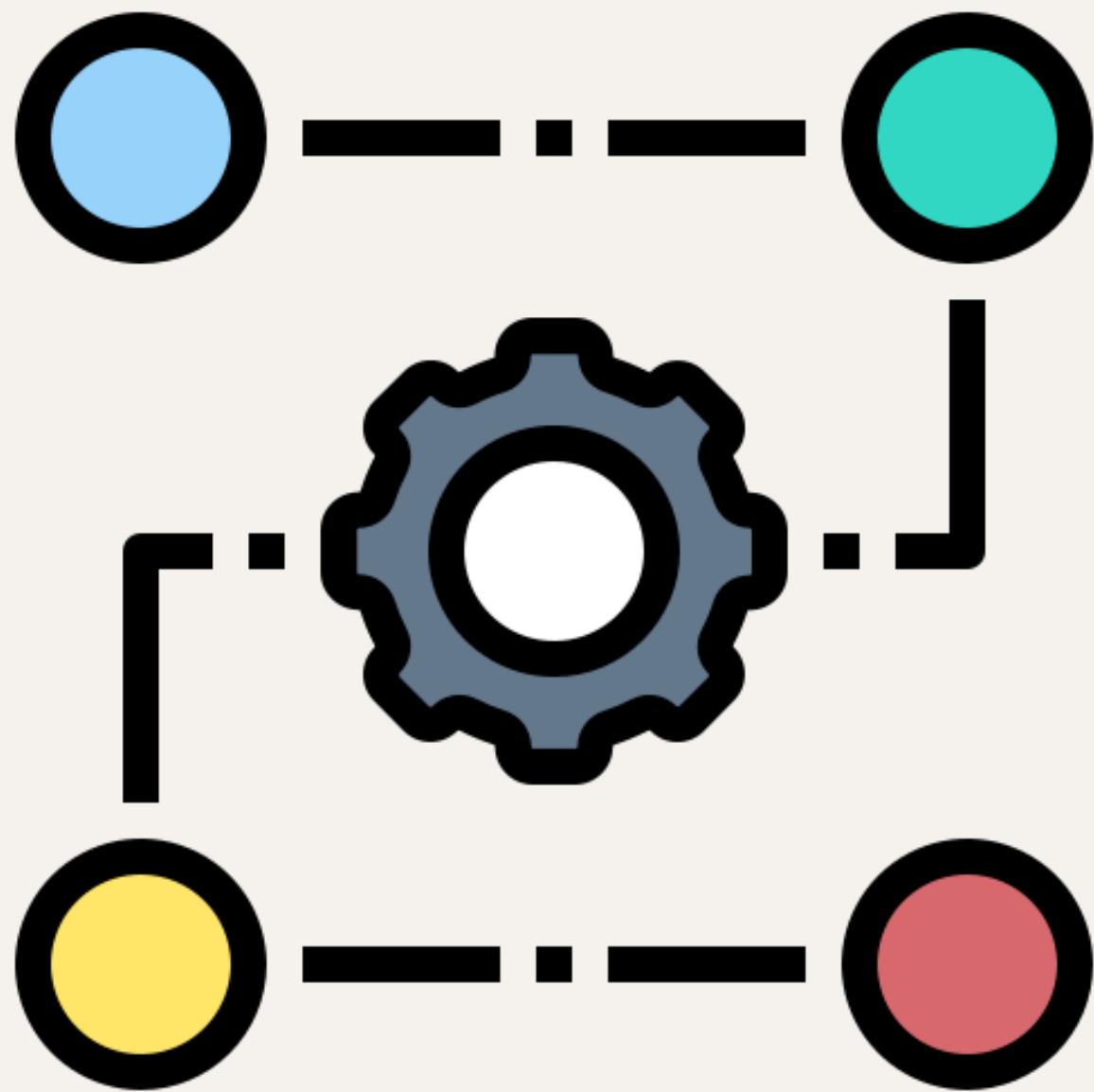


Một hệ thống/mô hình nhận dữ liệu đầu vào (input) là đoạn văn bản tóm tắt của đề tài và đề xuất (output) tiêu đề phù hợp cho đề tài.

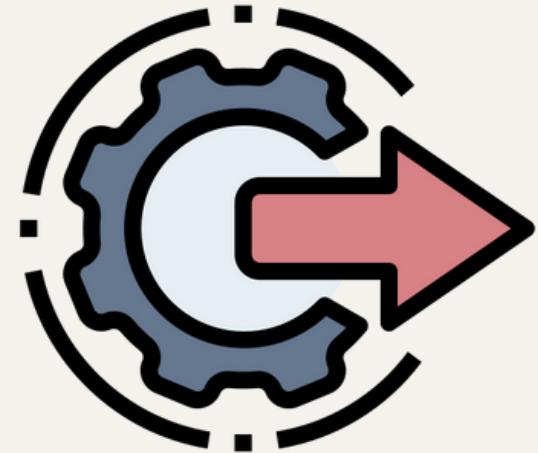
Đánh giá xem output có chính xác không bằng cách so sánh với một kết quả chuẩn (gold output)

02

INPUT VÀ OUTPUT

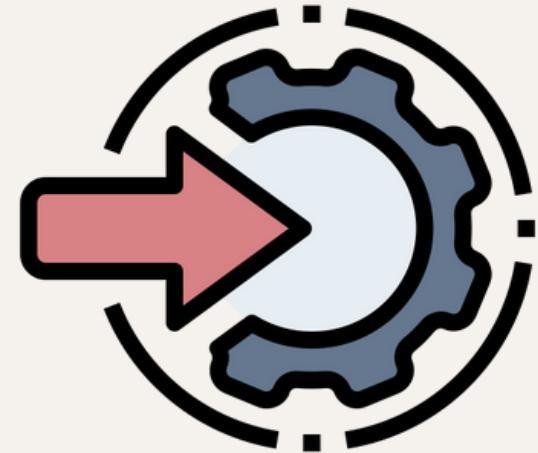


INPUT



Dữ liệu văn bản (Text Data): Ví dụ: một đoạn tóm tắt văn bản hoặc một đoạn giới thiệu về đề tài nghiên cứu khoa học.

OUTPUT



Đề xuất tiêu đề: dựa trên những dữ liệu đã thu thập, đưa ra tiêu đề về đề tài khoa học.

03

DATA SOURCE



Nguồn thu thập data

Dữ liệu được thu thập từ trang **Springer** - một nhà xuất bản sách và tạp chí khoa học, kỹ thuật và y học.

Với lĩnh vực hướng đến cho việc thu thập dữ liệu là **Machine Learning**.



Springer

Quy trình thu thập dữ liệu

STT	Bước thực hiện	Mô tả
01	Tải danh sách URL	Dùng <u>requests</u> và <u>BeautifulSoup</u> để cào lầy URL sau đó dựa vào journal để cào đường dẫn đến bài báo trong kết quả tìm kiếm.
02	Cào lầy dữ liệu những đường dẫn đã thu thập	Cào lầy tiêu đề và phần tóm tắt của từng bài báo.
03	Xử lý dữ liệu	Loại bỏ những dữ liệu bị trùng hoặc thiếu.
04	Lọc bài báo khoa học liên quan	Lọc kết quả theo chủ đề (Machine Learning).

Kết quả thu thập được

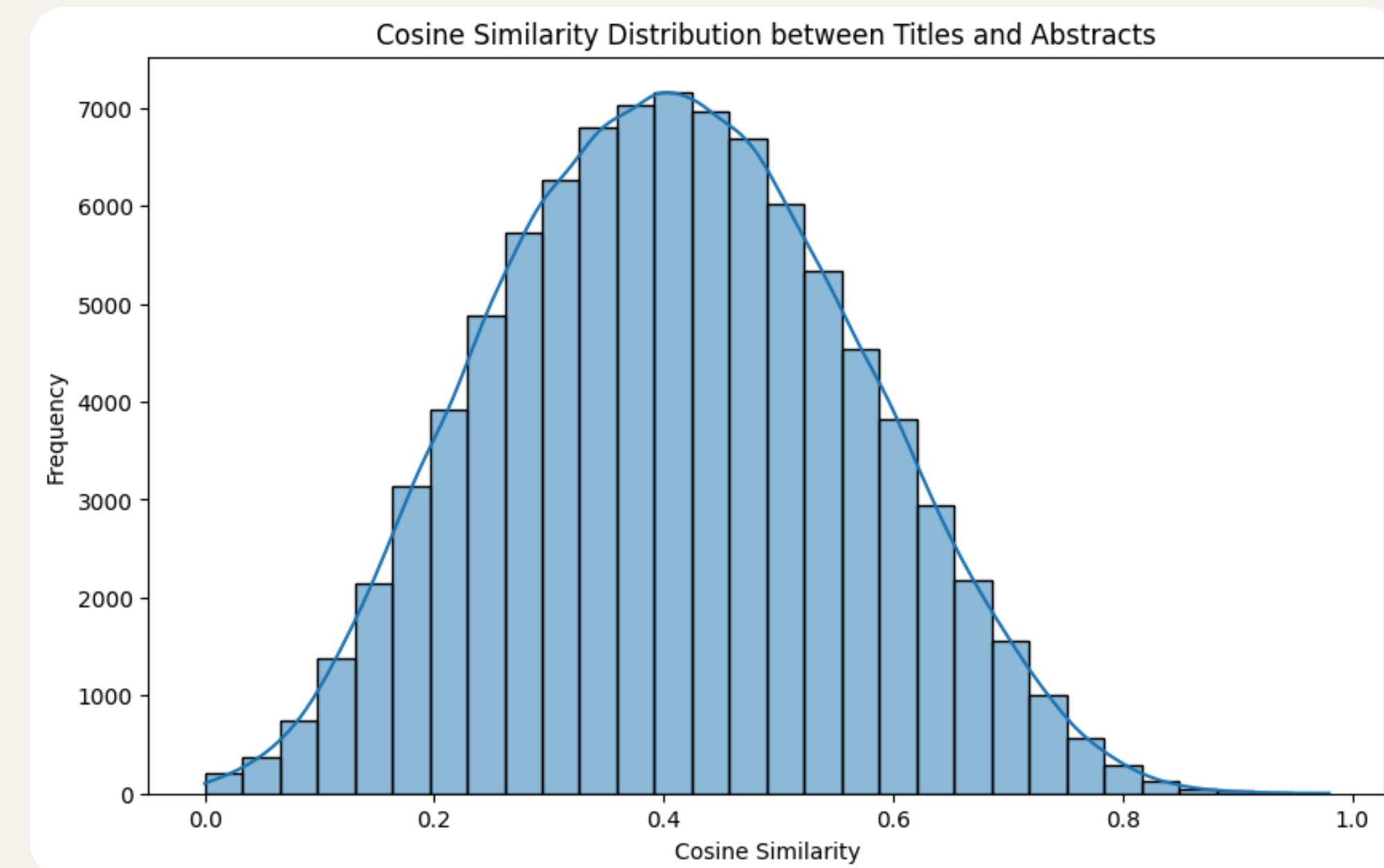
	Số journal	Tổng số dữ liệu lấy được	Tổng số dữ liệu sau khi xử lý
Nam	960	1.986.465	1.620.652
Thuận	1000	2.197.457	1.668.187
Thiện	1000	2.300.761	1.843.756
Tổng	2960	6.484.683	5.132.595
Sau khi lọc theo các từ khóa liên quan đến Machine Learning			91.788

EDA và lọc nâng cao

- Một số bài báo có abstract dài quá **512** tokens, vượt quá giới hạn xử lý của hầu hết các mô hình transformer.
- Do đó, dữ liệu tiếp tục được lọc:
 - **Abstract** có độ dài từ 128 - 512 tokens.
 - **Tiêu đề** (title) có độ dài từ 8 - 32 tokens.

EDA và lọc nâng cao

- Ngoài ra còn vấn đề về **độ tương đồng Cosine** khi giữa abstract và title có độ tương đồng **quá thấp** dẫn đến hiệu suất đầu ra của mô hình bị kém.
- **Giải pháp:** Loại bỏ những dòng dữ liệu có **độ tương đồng Cosine** < 0.3 .



Tiền xử lý từ vựng và vấn đề phân chia dữ liệu

Sau khi xử lý từ vựng (thống kê số lượng từ và tần suất), nhóm nhận thấy:

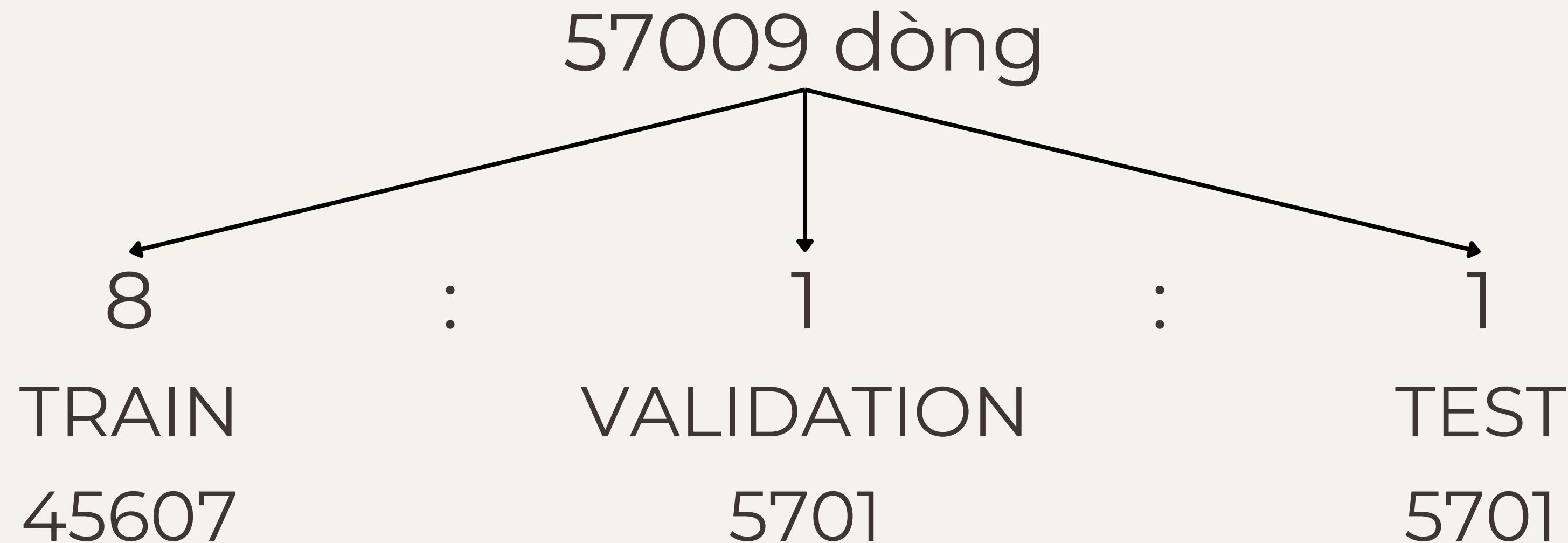
- Cách chia tập dữ liệu bằng **train_test_split()** của *scikit-learn* không đảm bảo sự phân bố đồng đều giữa các tập.
- Nhiều từ trong tập test và tập validation không xuất hiện trong tập train, gây ảnh hưởng tiêu cực đến hiệu quả mô hình.

Giải pháp

- Sử dụng **phương pháp chọn mẫu phân tổ (stratified sampling)** để đảm bảo sự phân bố từ khóa đồng đều giữa các tập **train, validation** và **test**.
- Ý tưởng:
 - Chia dữ liệu thành nhóm nhỏ dựa trên từ khóa chính (vd: cnn, rnn, deep learning,...).
 - Sau đó chia mẫu từ mỗi nhóm để tạo ra các tập dữ liệu **train, validation** và **test**.
 - Đồng thời chú ý đến các từ khóa xuất hiện quá ít trong tập dữ liệu có thể gây nhiễu model sau này.

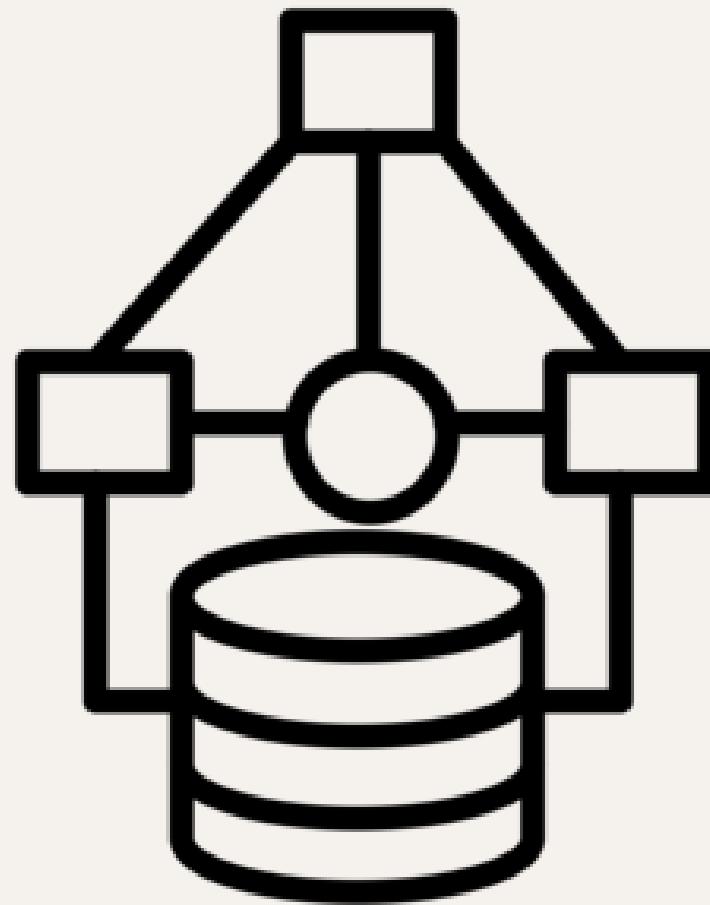
Cách chọn mẫu này được gọi là **token frequency-based and vocabulary-aware stratified sampling**.

Tập dữ liệu sau khi xử lý



04

DÁNH GIÁ



Cách xác thực kết quả

BERTScore

Bidirectional
Encoder
Representations
from **T**ransformers

Đánh giá chất lượng văn bản sinh bằng cách so sánh **embedding** của từng từ trong câu gốc và câu sinh dựa trên **BERT**. Sử dụng **cosine similarity** để tính **Precision, Recall, F1-score**, giúp đo lường mức độ tương đồng về ngữ nghĩa.

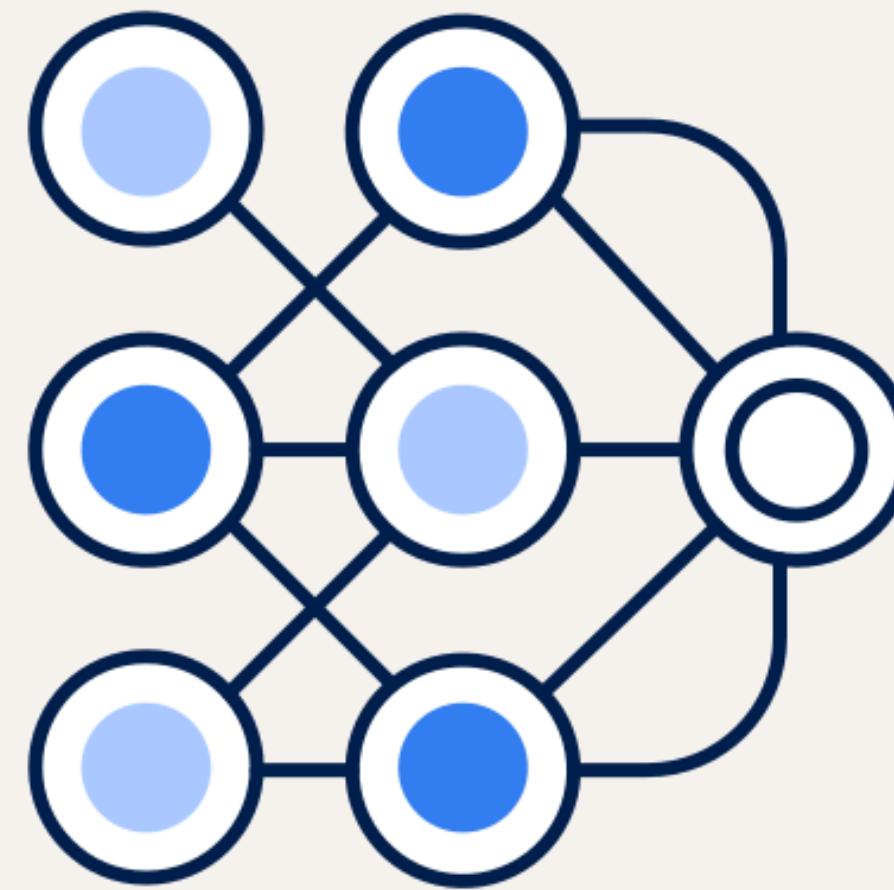
ROUGE Score

Recall-**O**riented
Understudy
for **G**isting
Evaluation

So sánh văn bản sinh với văn bản gốc dựa trên **n-gram** (ROUGE-N), chuỗi con dài nhất (**Longest Common Subsequence - LCS**) (ROUGE-L), hoặc **skip-bigram** (ROUGE-S).

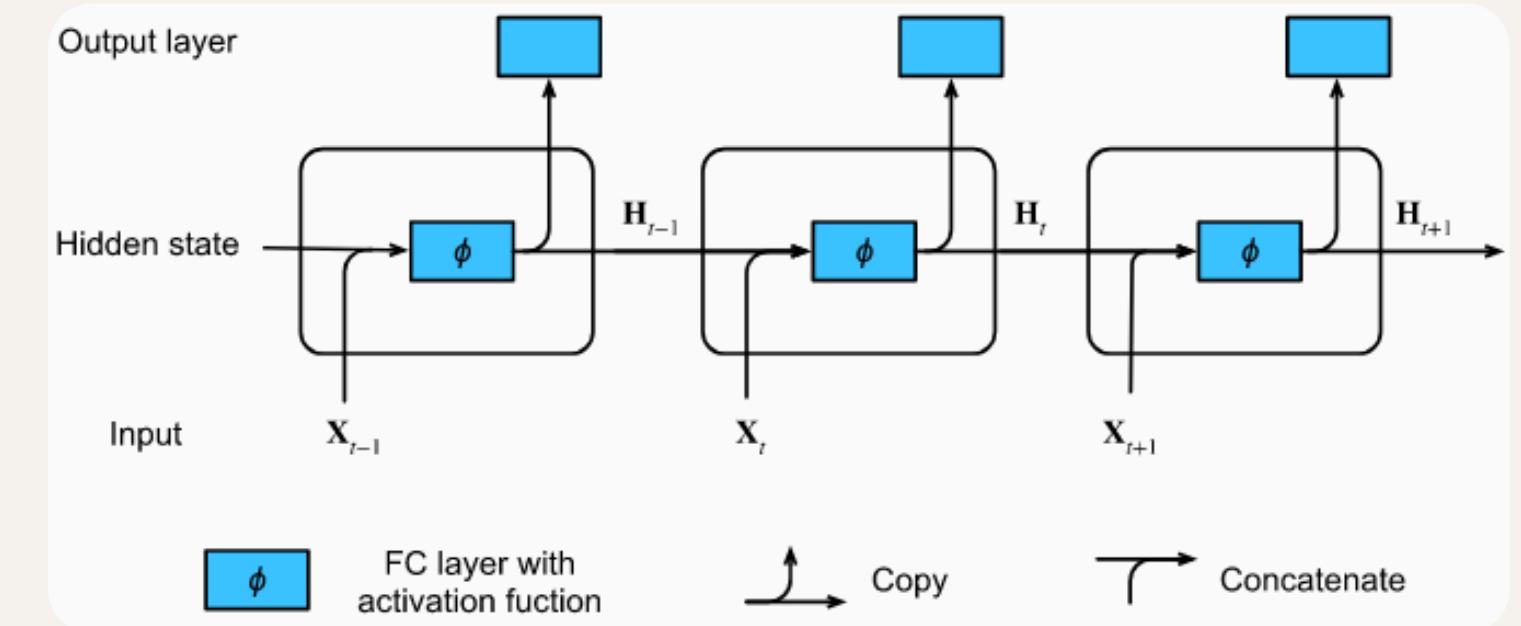
05

MÔ HÌNH



GRU (GATED RECURRENT UNIT)

GRU là một biến thể của mạng nơ-ron hồi tiếp (RNN), giúp giảm vấn đề tiêu biến đạo hàm bằng cách sử dụng cổng cập nhật và cổng xóa.



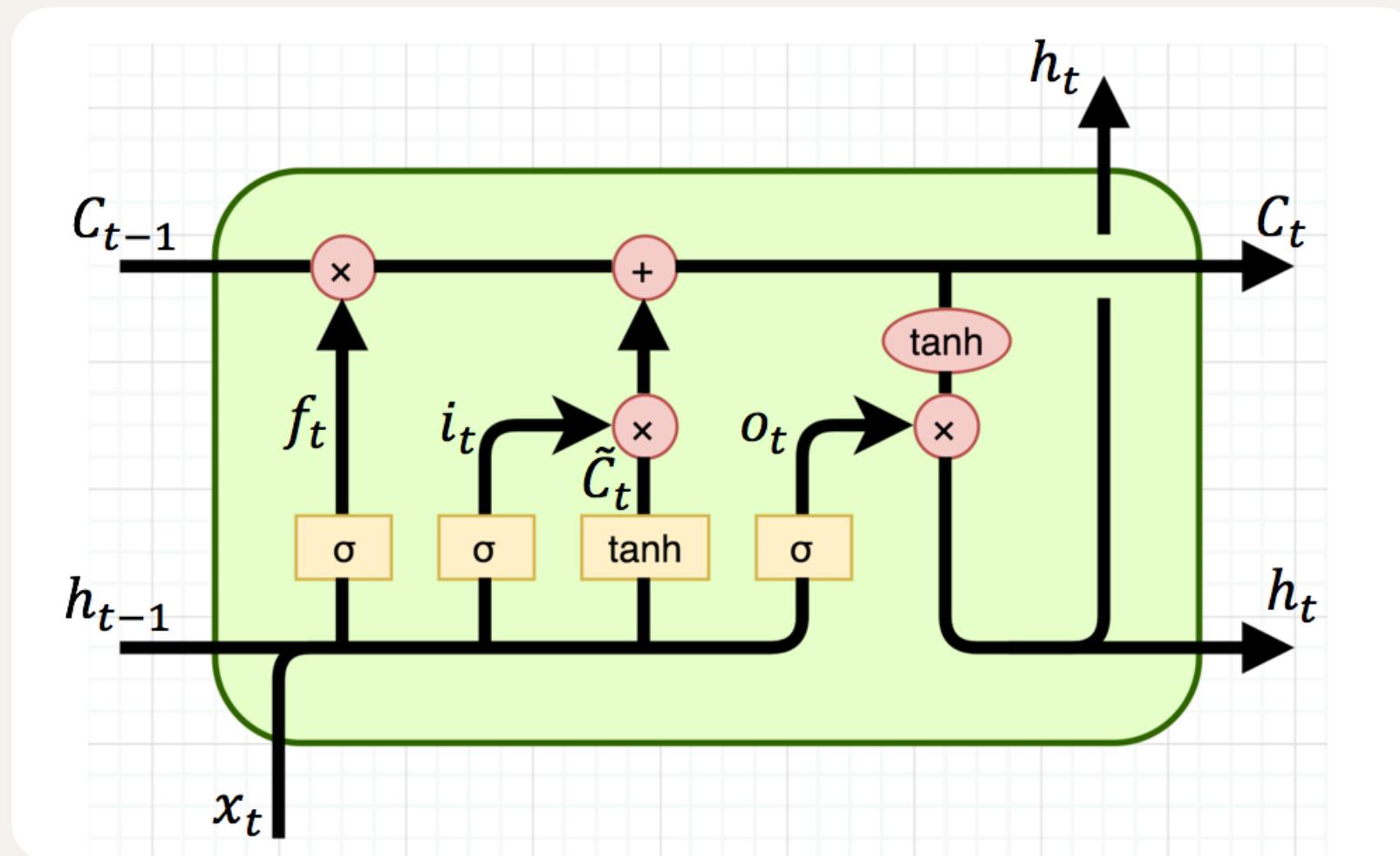
GRU (GATED RECURRENT UNIT)

Quá trình huấn luyện GRU được thực hiện với hai loại embedding:

- Tự tạo bộ embedding: Sử dụng phương pháp texts_to_sequences để chuyển đổi văn bản thành chuỗi số, phục vụ cho quá trình embedding.
- GloVe: Sử dụng embedding có sẵn từ GloVe để cải thiện chất lượng biểu diễn từ.

LSTM (LONG SHORT-TERM MEMORY)

LSTM là một biến thể cải tiến của RNN, sử dụng các cổng đầu vào, đầu ra và cổng quên để duy trì thông tin dài hạn.



LSTM (LONG SHORT-TERM MEMORY)

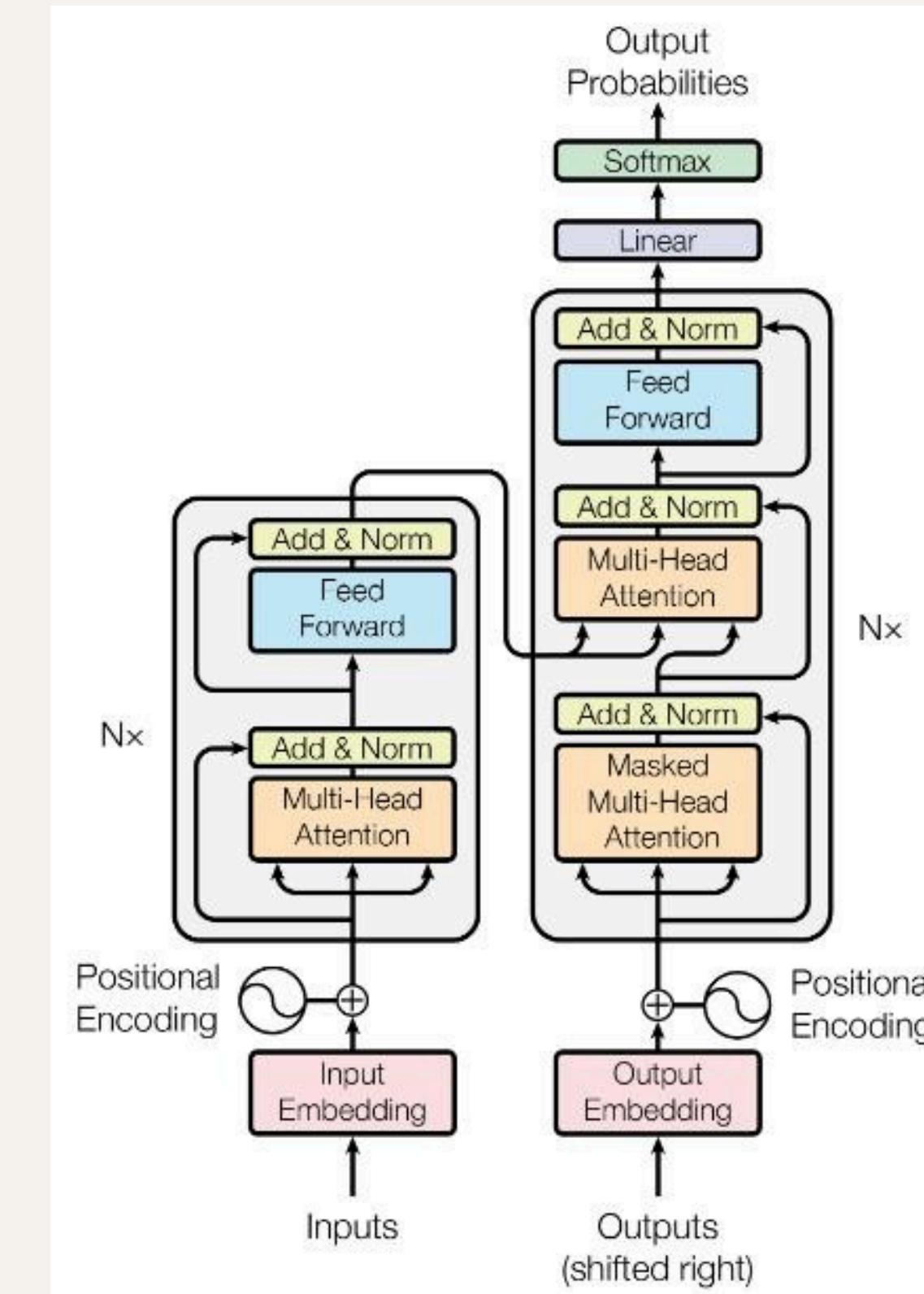
Tương tự GRU, LSTM được huấn luyện với hai phương pháp embedding:

- Tự làm embedding: Sử dụng phương pháp texts_to_sequences để chuyển đổi văn bản thành chuỗi số, phục vụ cho quá trình embedding.

GloVe: Sử dụng embedding có sẵn từ GloVe để cải thiện chất lượng biểu diễn từ.

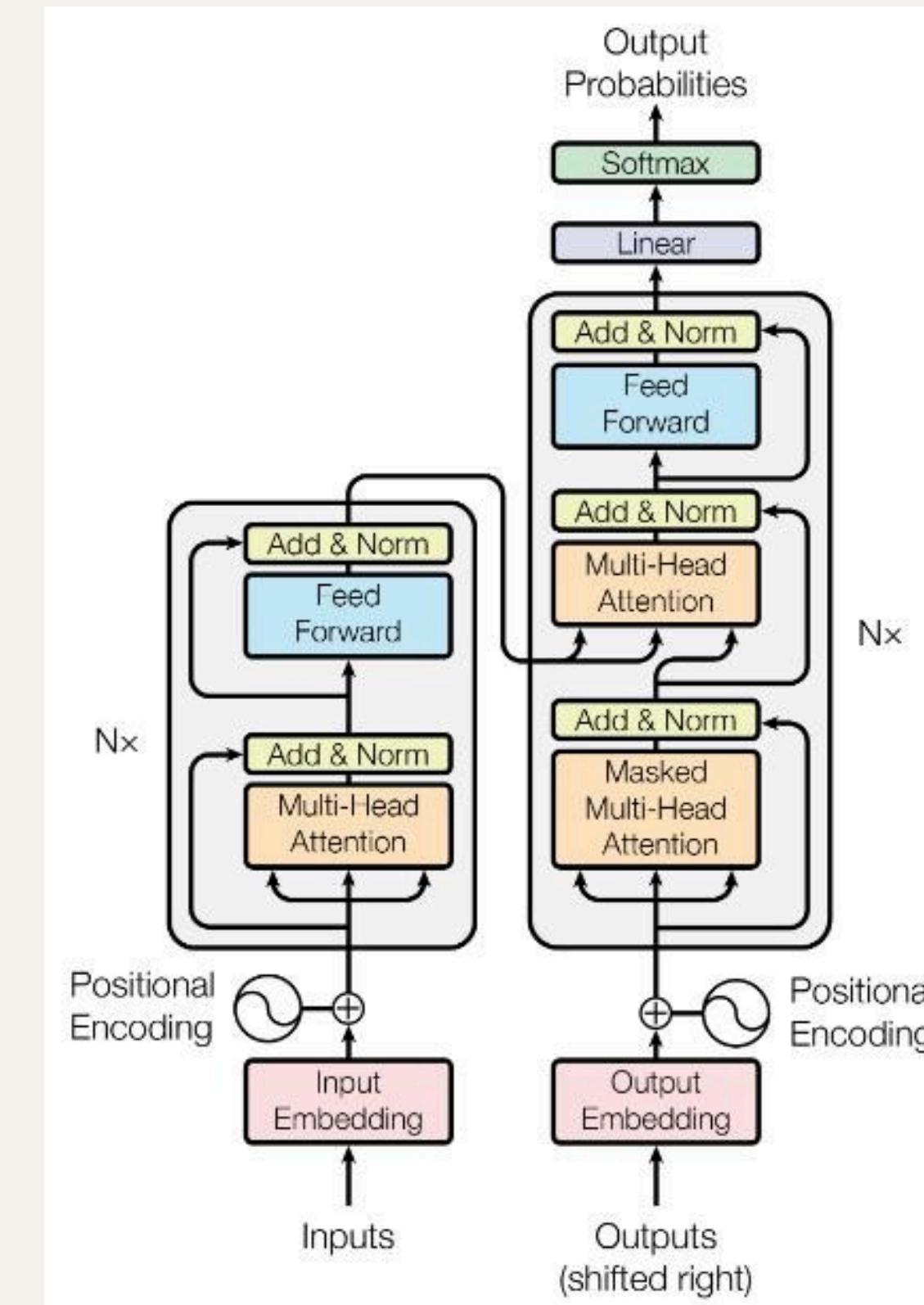
BART

BART được huấn luyện bằng cách phá hỏng văn bản gốc (ví dụ: xóa, tráo đổi, che từ) rồi yêu cầu mô hình khôi phục lại nó. Điều này giúp BART học được cả ngữ cảnh hai chiều (như BERT) và khả năng sinh văn bản (như GPT).



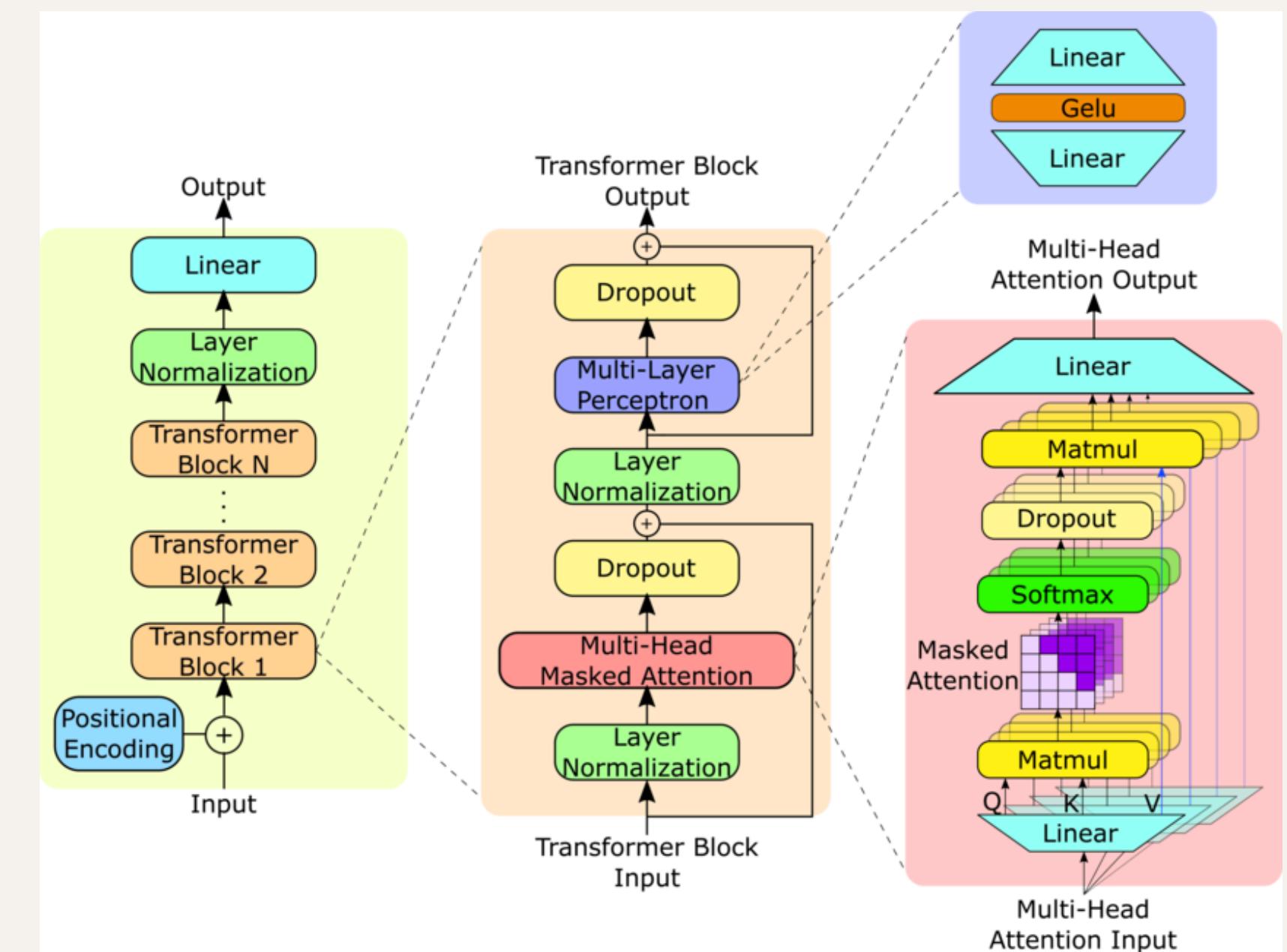
FLAN-T5

Flan-T5: Là phiên bản cải tiến của T5 (Text-to-Text Transfer Transformer) do Google nghiên cứu. Mô hình này được huấn luyện với nhiều nhiệm vụ NLP khác nhau, bao gồm tóm tắt và tạo tiêu đề.



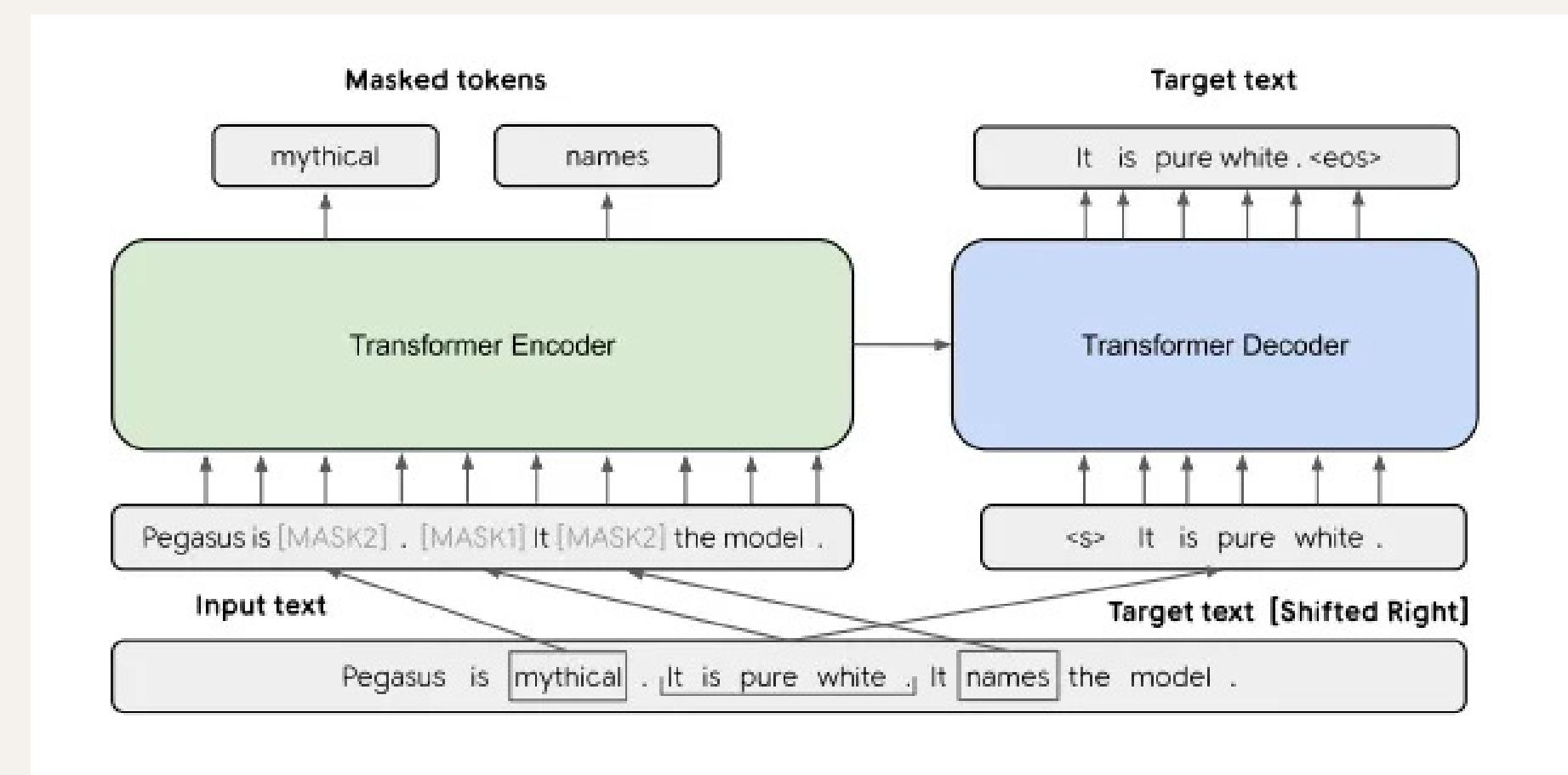
GPT-2

GPT-2: Là phiên bản trước của GPT-3, được phát triển bởi OpenAI. Mô hình này có khả năng sinh văn bản tự nhiên nhưng bị giới hạn về độ dài ngữ cảnh và độ chính xác so với các mô hình tiên tiến hơn.



PEGASUS-XSUM

PEGASUS-XSum là mô hình tóm tắt văn bản dựa trên kiến trúc Transformer, được huấn luyện với chiến lược *masked language modeling* đặc biệt gọi là **Gap Sentence Generation** (GSG). Phiên bản huấn luyện trên tập XSum giúp mô hình tạo ra bản tóm tắt ngắn, súc tích và giàu thông tin cho từng đoạn văn.



06

KẾT QUẢ MÔ HÌNH



Kết quả GRU-LSTM (10 epoch)

Mô hình	ROUGE				BERTScore		
	rouge1	rouge2	rougeL	rougeLsum	Precision	Recall	F1
GRU (Custom Embedding)	0.0410	0.00276	0.0350	0.0350	0.6885	0.8012	0.7402
LSTM (Custom Embedding)	0.0371	0	0.0371	0.0371	0.8378	0.7814	0.8085
GRU (GloVe)	0.0370	0	0.0371	0.0371	0.8378	0.7815	0.8085
LSTM (GloVe)	0.0371	0	0.0370	0.0371	0.8378	0.7815	0.8085

Kết quả transformer-based model (3 epoch)

Mô hình	ROUGE				BERTScore		
	rouge 1	rouge 2	rouge L	rouge Lsum	Precision	Recall	F1
BART-base	0.5200	0.3153	0.4476	0.4477	0.9108	0.9013	0.9059
Flan-T5-base	0.5037	0.3000	0.4322	0.4320	0.9117	0.8985	0.9050
GPT-2	0.095	0.0491	0.0756	0.0756	0.7977	0.8814	0.8374
Pegasus-XSum	0.4829	0.2822	0.4156	0.4156	0.9074	0.891	0.8989

Kết quả transformer-based model (5 epoch)

Mô hình	ROUGE				BERTScore		
	rouge1	rouge2	rougeL	rougeL sum	Precision	Recall	F1
BART-base	0.5195	0.3146	0.4470	0.4472	0.9100	0.9010	0.9053
Flan-T5- base	0.5074	0.3035	0.4354	0.4352	0.9120	0.8993	0.9054

SO SÁNH FINE-TUNED MODEL (3 EPOCH) VÀ ZERO-SHOT LLMS TRÊN 500 DATA

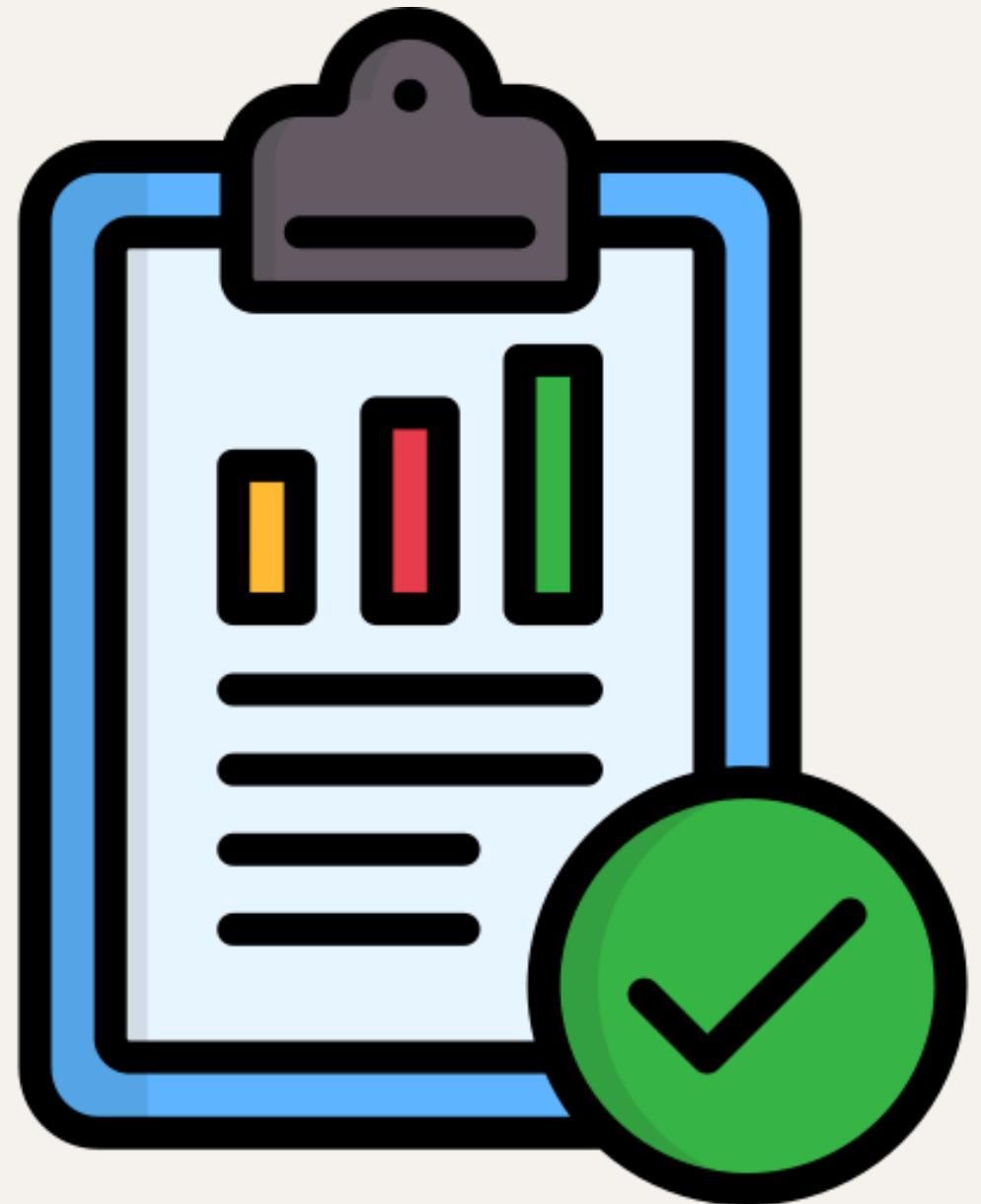
Mô hình	ROUGE				BERTScore		
	rouge 1	rouge 2	rouge L	rouge Lsum	Precision	Recall	F1
BART-base	0.5473	0.5321	0.4681	0.4681	0.9149	0.9036	0.9090
Grok3	0.5435	0.3011	0.4467	0.4467	0.8955	0.9	0.8976
GPT-4o	0.5321	0.3335	0.4354	0.4354	0.8953	0.8995	0.8973

SO SÁNH FINE-TUNED MODEL (5 EPOCH) VÀ ZERO-SHOT LLMS TRÊN TOÀN BỘ TẬP TEST

Mô hình	ROUGE				BERTScore		
	rouge1	rouge2	rougeL	rougeLsum	Precision	Recall	F1
BART-base	0.5195	0.3146	0.4470	0.4472	0.9100	0.9010	0.9053
Flan-T5-base	0.5074	0.3035	0.4354	0.4352	0.9120	0.8993	0.9054
Gemini 2.0 Flash	0.1073	0.0516	0.0831	0.0831	0.8717	0.7983	0.8333
GPT-4.1 mini	0.1328	0.0682	0.1001	0.1001	0.8784	0.8041	0.8396

07

LLM FINE-TUNE



LLAMA (META AI):

- **Llama (Meta AI)**: Gồm nhiều phiên bản từ 1B đến 8B tham số, sử dụng kiến trúc transformer tự hồi quy.
- Các bản Instruct được tinh chỉnh bằng SFT và RLHF để nâng cao độ chính xác, an toàn và khả năng làm theo chỉ dẫn.



MISTRAL

- **Mixtral (Mistral AI)**: Mô hình 7B tham số với kiến trúc Mixture-of-Experts, tối ưu cho tác vụ theo chỉ dẫn.
- Được tinh chỉnh bằng dữ liệu khoa học nhằm cải thiện chất lượng sinh tiêu đề từ tóm tắt.



DEEPSEEK

DeepSeek-R1-Distill-Llama-8B:

Mô hình chưng cất từ Llama, tập trung vào hiệu quả và chất lượng sinh văn bản trong các nhiệm vụ nghiên cứu.

DeepSeek-R1-Distill-Qwen-7B:

Mô hình chưng cất từ Qwen2.5, được tinh chỉnh với 800k mẫu dữ liệu lý luận từ DeepSeek-R1, tập trung vào hiệu quả tính toán và hiệu suất vượt trội trong các nhiệm vụ toán học, lập trình và lý luận.



PHI-4

Phi-4: Mô hình ngôn ngữ 14B tham số của Microsoft, sử dụng dữ liệu tổng hợp chất lượng cao và dữ liệu web được lọc, nổi bật với khả năng lập luận, toán học và lập trình. Hỗ trợ ngữ cảnh dài 16K token, tối ưu cho môi trường tính toán hạn chế.



KẾT QUẢ MÔ HÌNH BASELINE

Mô hình	ROUGE				BERTScore		
	rouge1	rouge2	rougeL	rouge Lsum	Precision	Recall	F1
BART-base	0.5200	0.3153	0.4476	0.4477	0.9108	0.9013	0.9059
Flan-T5-base	0.5037	0.3000	0.4322	0.4320	0.9117	0.8985	0.9050

KẾT QUẢ SO SÁNH

Mô hình	ROUGE				BERTScore		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Precision	Recall	F1
Llama 3.2 1B Instruct (0.7 cosine, Unslot, Quantized, LoRA)	0.1172	0.0751	0.1009	0.1021	0.7515	0.8758	0.8081
Llama 3.2 1B Instruct (0.5 cosine, Unslot, Quantized, LoRA)	0.1490	0.0848	0.1252	0.1259	0.7719	0.8773	0.8199
Llama 3.2 1B Instruct (0.7 cosine, Unslot, Quantized, no LoRA)	0.1089	0.0708	0.0957	0.0971	0.7859	0.8820	0.8311
Llama 3.2 1B Instruct (0.7 cosine, LoRA, Quantized)	0.1228	0.0821	0.1093	0.1111	0.8017	0.8838	0.8406
Llama 3.2 3B (0.7 cosine, Unslot, Quantized, LoRA)	0.0969	0.0615	0.0856	0.0870	0.7678	0.8717	0.8161

KẾT QUẢ SO SÁNH

Mô hình	ROUGE				BERTScore		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Precision	Recall	F1
Llama 3.2 3B Instruct (0.7 cosine, Unsloth, Quantized, LoRA)	0.0902	0.0598	0.0788	0.0797	0.7649	0.8776	0.8171
Llama 3.1 8B Instruct (0.7 cosine, Unsloth, Quantized, LoRA)	0.1639	0.1030	0.1424	0.1432	0.7474	0.8750	0.8039
Mixtral-7B-Instruct-v0.3 (0.7 cosine, Unsloth, Quantized, LoRA)	0.1192	0.0805	0.1069	0.1074	0.7930	0.8785	0.8332
DeepSeek-R1-Distill-Llama-8B (0.7 cosine, Unsloth, Quantized, LoRA)	0.1175	0.0709	0.1021	0.1031	0.7977	0.8743	0.8337
DeepSeek-R1-Distill-Qwen-7B (0.7 cosine, Unsloth, Quantized, LoRA)	0.1104	0.0724	0.0993	0.1005	0.7945	0.8807	0.8352
Phi-4 (0.7 cosine, Unsloth, Quantized, LoRA)	0.1979	0.1366	0.1722	0.1739	0.8121	0.8897	0.8485

08

CẤI TIẾN

MODEL



ĐỀ XUẤT CÁC PHƯƠNG PHÁP CẢI THIỆN

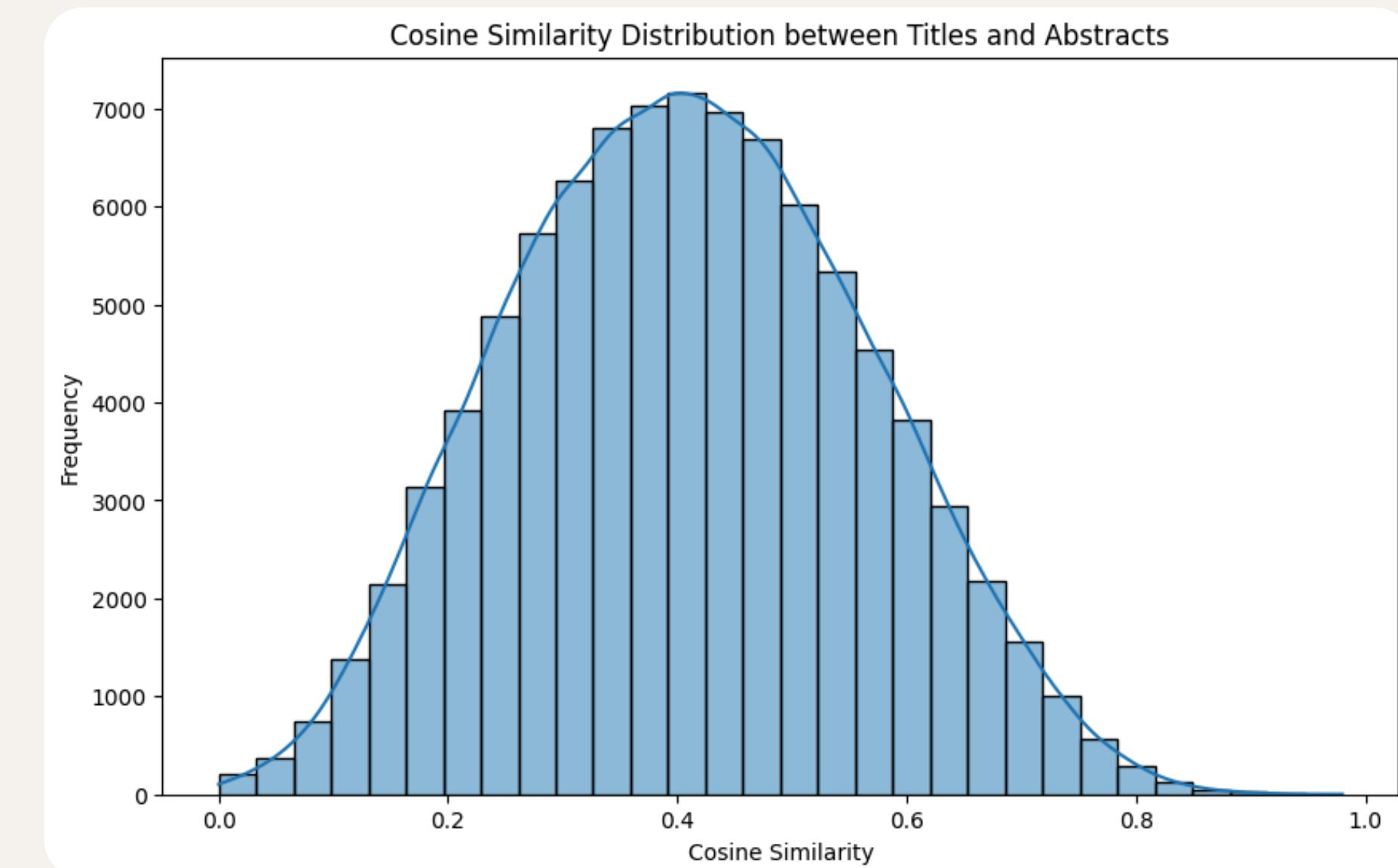
- Cosine similarity-based data trimming
- Keyword-aware instruction fine-tune
- Sentence Order Aware
- Data Augmentation

COSINE SIMILARITY-BASED DATA TRIMMING

Cắt data theo **cosine similarities** tăng dần và train đến khi đạt mức bão hòa (tại 1 mức cosine similarity nào đó model không có cải thiện gì thêm hoặc thậm chí giảm).

Mục đích:

- Lọc ra được những data "chất lượng cao" tăng hiệu suất model.
- Giảm thời gian training vì số lượng dữ liệu đã được giảm bớt.



COSINE SIMILARITY-BASED DATA TRIMMING

Đồng thời loại bỏ mẫu có abstract với lượng token > **450** thay vì **512** như lúc trước để chứa lượng input token dư còn lại cho:

- Các câu lệnh prompt cho instruction fine-tune.
- Danh sách các keyword được thêm vào input.

Trước:

512 token abstract

Sau xử lý:

450 token abstract

62 token prompt + keywords

COSINE SIMILARITY-BASED DATA TRIMMING

Model	Cosine Similarity	Train data Size	ROUGE 1	ROUGE 2	ROUGE L	ROUGE Lsum	BERTScore F1	Training time
Flan-T5-base	= 0.3	45.6K	0.5074	0.3035	0.4354	0.4352	0.9054	~9h
	= 0.35	37.9K	0.5756	0.3583	0.4861	0.4861	0.9094	~8h

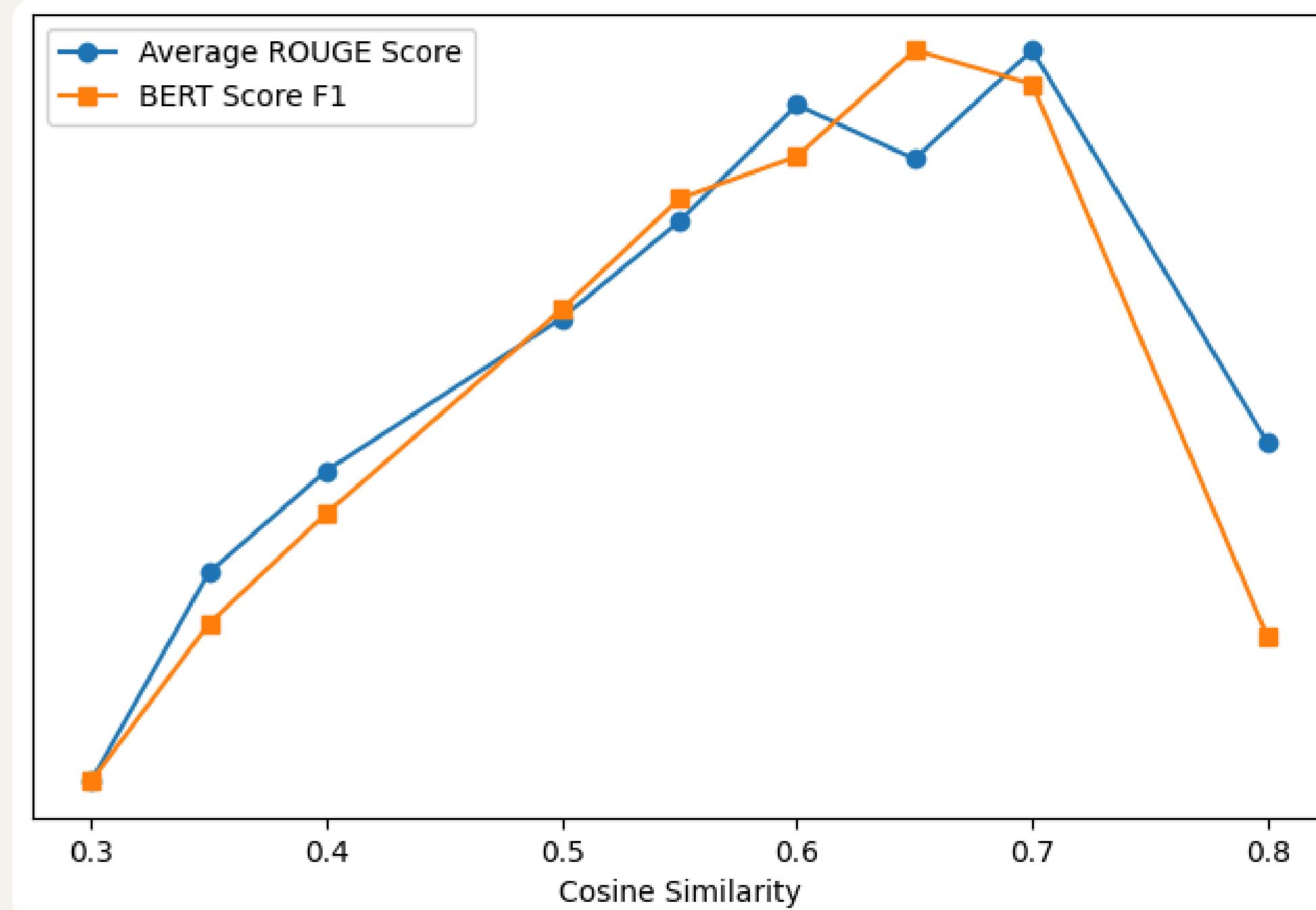
COSINE SIMILARITY-BASED DATA TRIMMING

Train model trên các tập dữ liệu với cosine similarity khác nhau cho thấy:

- Model cho điểm số tốt nhất ở khoảng cosine từ **0.7**
- Sau đó giảm mạnh từ 0.8 trở đi.

Dự đoán nguyên nhân:

- Ở mức 0.8 cosine similarity, kích thước data train < **1000** mẫu.



KẾT QUẢ SO SÁNH TRÊN MÔ HÌNH BART-BASE

cosine	ROUGE				BERTScore		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Precision	Recall	F1
0.35	0.5249	0.3204	0.4509	0.4513	0.9109	0.9022	0.9064
0.4	0.5416	0.3359	0.466	0.4659	0.9134	0.9046	0.9088
0.5	0.5667	0.3669	0.4933	0.4934	0.9166	0.9084	0.9123
0.55	0.579	0.3827	0.5035	0.5038	0.9188	0.9106	0.9145
0.6	0.6079	0.4123	0.5351	0.5357	0.9236	0.9148	0.919

KẾT QUẢ SO SÁNH TRÊN MÔ HÌNH BART-BASE

cosine	ROUGE				BERTScore		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Precision	Recall	F1
0.65	0.6068	0.4179	0.5359	0.536	0.9242	0.9151	0.9195
0.7	0.6201	0.4311	0.5413	0.5414	0.9234	0.9145	0.9188
0.75	0.6071	0.4124	0.5442	0.546	0.9241	0.9142	0.919
0.8	0.5338	0.3508	0.4715	0.4743	0.9112	0.8981	0.9045

KEYWORD-AWARE INSTRUCTION FINE-TUNE

- **Prompt Engineering:** Thêm một prompt cụ thể vào đầu vào của mô hình nhằm hướng dẫn model thực hiện một tác vụ cụ thể: sinh ra một tiêu đề ngắn gọn và tổng quát dựa trên abstract và từ khóa.
- **Keyword Extraction:** Sử dụng KeyBERT để trích xuất từ khóa từ title. Train model để học cách trích xuất từ khóa từ abstract tương tự như từ khóa đã trích xuất từ title. Từ khóa được thêm vào đầu vào của mô hình, giúp cung cấp thêm thông tin.
- **Custom Loss Function:** Hàm mất mát gốc được thêm một thành phần phạt (penalty) dựa trên từ khóa, phạt các tiêu đề sinh ra nếu chúng không chứa từ khóa.
- **Multilearning:** Train model thực hiện vừa học cách trích xuất từ khóa vừa học cách sinh tiêu đề.

KẾT QUẢ TỐT NHẤT TRƯỚC CÀI TIỀN

Mô hình	ROUGE				BERTScore		
	rouge1	rouge2	rougeL	rouge Lsum	Precision	Recall	F1
BART-base	0.5200	0.3153	0.4476	0.4477	0.9108	0.9013	0.9059
Flan-T5-base	0.5037	0.3000	0.4322	0.4320	0.9117	0.8985	0.9050

TRAIN MODEL EXTRACT KEYWORD TRÊN DATA LỚN, TRAIN MODEL MULTILEARNING TRÊN DATA LỚN, DÙNG KEYBERT

Mô hình	ROUGE				BERTScore
	rouge1	rouge2	rougeL	rougeLsum	F1
Bert-base-uncased, Bart-base	0.5493	0.3330	0.4636	0.4636	0.9055
SciBERT-scivocab- uncased, Bart-base	0.5447	0.3348	0.4643	0.4643	0.9064

TRAIN MODEL EXTRACT KEYWORD TRÊN DATA NHỎ, TRAIN MODEL MULTILEARNING TRÊN DATA NHỎ, DÙNG KEYBERT

Mô hình	ROUGE				BERTScore
	rouge1	rouge2	rougeL	rougeLsum	F1
Flan-t5-base, Bart-base	0.6532	0.4471	0.5706	0.5706	0.9242
Flan-t5-base, Flan-t5-base	0.6465	0.4557	0.5541	0.5541	0.9199
Roberta-base, Bart-base	0.6498	0.4490	0.5835	0.5835	0.9236
SciBERT-scivocab-uncased, Bart-base	0.6502	0.4568	0.5697	0.5697	0.9246
SciBERT-scivocab-uncased, Flan-t5-base	0.6618	0.4629	0.5638	0.5638	0.9214
Roberta-base, Flan-t5-base	0.6567	0.4590	0.5613	0.5613	0.9220

TRAIN MODEL EXTRACT KEYWORD TRÊN DATA LỚN, TRAIN MODEL MULTILEARNING TRÊN DATA NHỎ, DÙNG KEYBERT

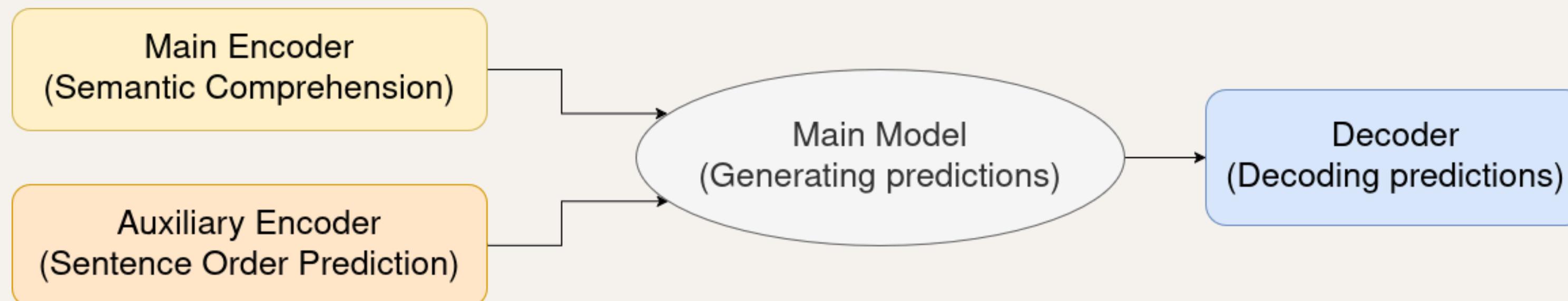
Mô hình	ROUGE				BERTScore
	rouge1	rouge2	rougeL	rougeLsum	F1
Flan-t5-base, Bart-base	0.6333	0.4335	0.5518	0.5518	0.9215
Flan-t5-base, Flan-t5-base	0.6357	0.4435	0.5430	0.5430	0.9193
Roberta-base, Bart-base	0.6544	0.4506	0.5773	0.5773	0.9244
SciBERT-scivocab-uncased, Bart-base	0.6494	0.4516	0.5596	0.5596	0.9224
SciBERT-scivocab-uncased, Flan-t5-base	0.6616	0.4737	0.5659	0.5659	0.9228
Roberta-base, Flan-t5-base	0.6493	0.4575	0.5591	0.5591	0.9207

Mô hình	ROUGE				BERTScore
	rouge1	rouge2	rougeL	rougeLsum	F1
Train model extract keyword trên data nhỏ, train model multilearning trên data nhỏ, dùng keyt5-large					
SciBERT-scivocab-uncased, Bart-base	0.6562	0.4620	0.5776	0.5776	0.9239
Train model extract keyword trên data lớn, train model multilearning trên data nhỏ, dùng keyt5-large					
SciBERT-scivocab-uncased, Bart-base	0.6575	0.4535	0.5657	0.5657	0.9226
Train model extract keyword trên data lớn, train model multilearning trên data nhỏ, dùng keyBart					
SciBERT-scivocab-uncased, Bart-base	0.6512	0.4484	0.5597	0.5597	0.9236
Train model extract keyword trên data nhỏ, train model multilearning trên data nhỏ, dùng keyBart					
SciBERT-scivocab-uncased, Bart-base	0.6514	0.6514	0.5691	0.5691	0.9242

SENTENCE ORDER AWARE

Tạo ra một **fusion-based joint model** với 2 encoder, mỗi encoder được train cho một tác vụ khác nhau:

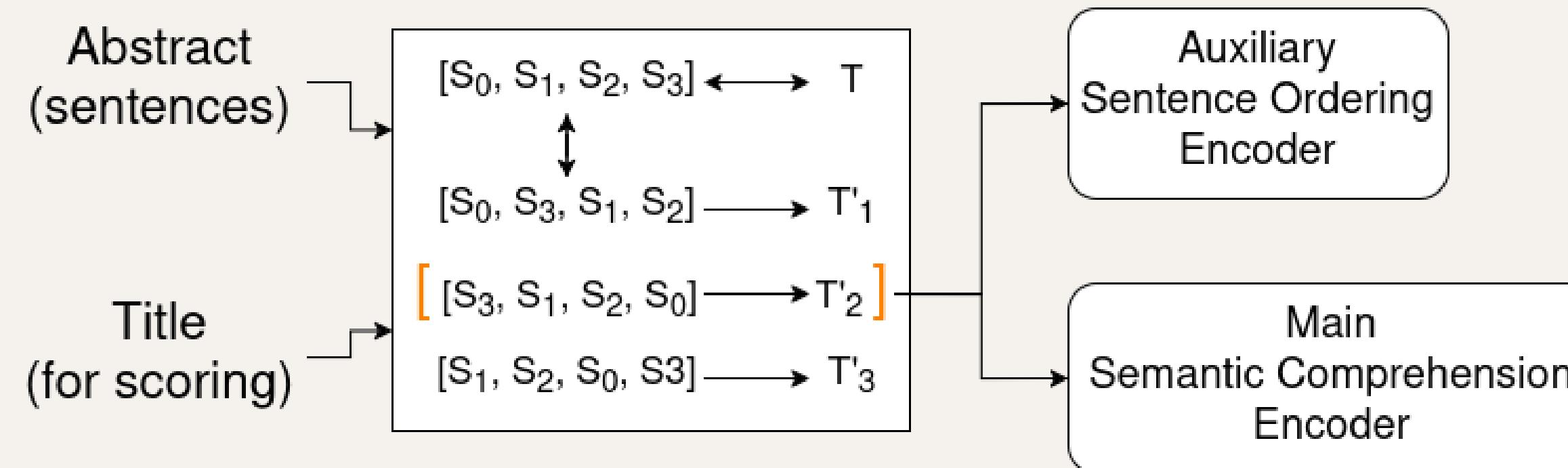
- Encoder chính (**main task**): đảm nhiệm **semantic comprehension**, nói cách khác là được train để hiểu ngữ nghĩa của input.
- Encoder phụ (**auxiliary task**): đảm nhiệm **sentence order prediction**, dự đoán thứ tự tối ưu nhất của các câu trong abstract.



SENTENCE ORDER AWARE

Các bước thực hiện:

1. Sinh ra tất cả hoán vị của các câu trong abstract.
2. Tạo ra tiêu đề cho từng hoán vị.
3. Chọn ra tiêu đề có ROUGE-S tốt nhất làm input cho **forward pass**.
4. Encoder chính sẽ học cách "hiểu" input, còn encoder phụ sẽ học thứ tự câu tối ưu nhất (hoán vị nào cho ra kết quả tốt nhất).



SENTENCE ORDER AWARE

Vấn đề phát sinh

Một abstract có thể lên đến **20 câu**, tức là:

$$20! = 2,432,902,008,176,640,000$$

cách sắp xếp câu, **quá lớn** so với tài nguyên hiện có cho việc training!

Giải pháp đề ra

- Áp dụng **top-k** số câu có điểm **sentence importance** cao nhất để thực hiện sinh hoán vị.
- Kết hợp **beam search** trong quá trình training để hòng tăng hiệu suất model.

SENTENCE ORDER AWARE

Tính chỉ số **Sentence importance** dựa trên độ **overlap từ khóa** của một câu so với **title**:

$$\text{Importance}(S_i) = \lambda \cdot \text{overlap}(S_i, T)$$

Trong đó:

S_i Là câu thứ i trong **abstract**

T Là tiêu đề bài báo (**title**)

$\text{overlap}(S_i, T)$ Số lượng từ khóa chung của câu thứ i và tiêu đề

λ Hệ số điều chỉnh để kiểm soát mức độ ưu tiên

SENTENCE ORDER AWARE

Ngoài ra cũng cần tính sentence importance của các câu liền kề so với câu đang xét để đảm bảo đầy đủ ngữ nghĩa nhất. Tức là:

$$\text{Importance}(S_{i\pm 1})$$

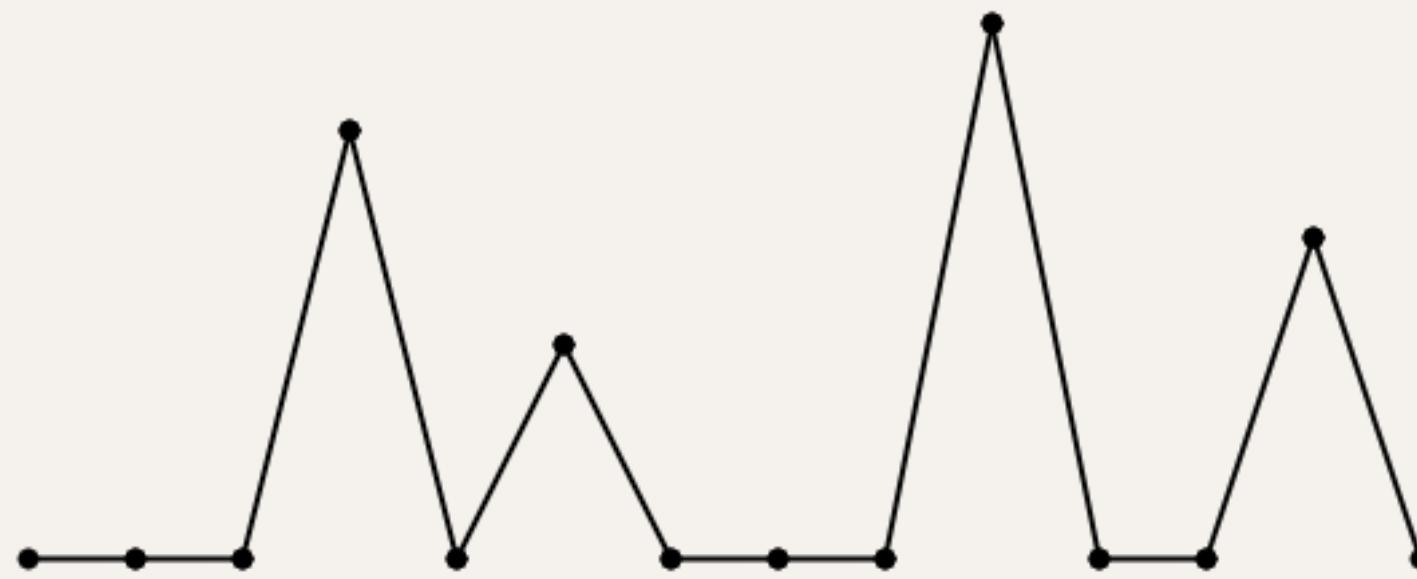
Trong đó, **overlap** sẽ được tính giữa câu liền kề ($i+1, i-1$) và câu đang xét (i).
Thay vì overlap của **câu thứ i** so với **tiêu đề**.

$$\text{overlap}(S_{i\pm 1}, S_i) \quad \text{thay vì} \quad \text{overlap}(S_i, T)$$

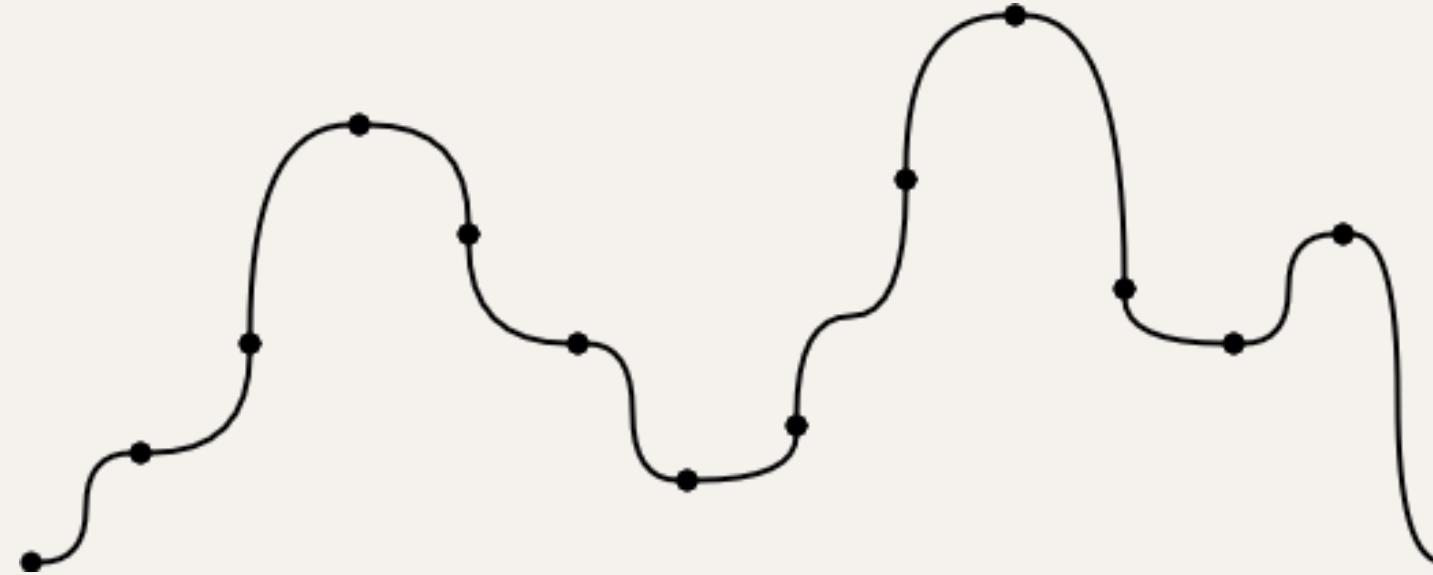
Phương pháp trên tạm gọi là **Sentence Importance Smoothing**

SENTENCE ORDER AWARE

Sentence Importance **mong muốn** khi:



Không có smoothing



Có smoothing

SENTENCE ORDER AWARE

Ví dụ minh họa ý tưởng:

1. Reinforcement learning agents often require large amounts of data to achieve high performance.
2. We propose a contrastive learning approach to improve sample efficiency in deep RL.
3. Our method encodes observations into a compact latent space using a contrastive loss.
4. We then use this representation to guide policy learning.
5. Experiments on Atari and DMControl show improved performance with fewer samples.



“Contrastive Learning Enhances Sample Efficiency in Deep Reinforcement Learning on Atari and DMControl”

SENTENCE ORDER AWARE

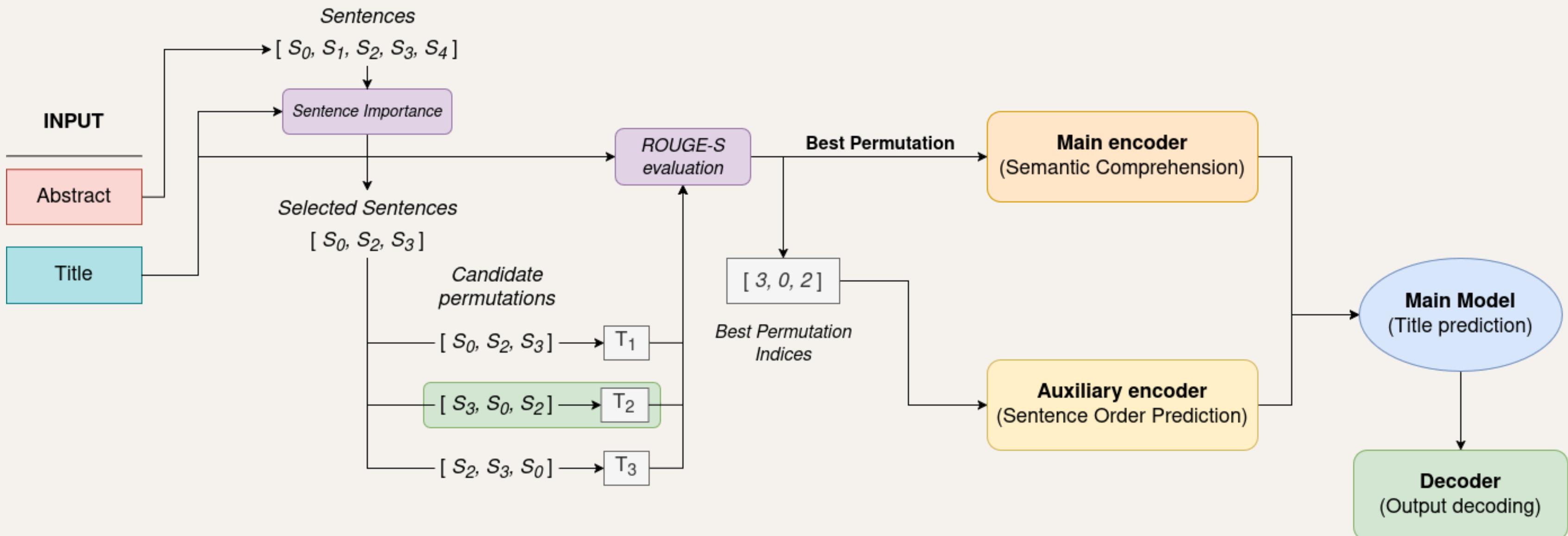
Ví dụ minh họa ý tưởng:

1. We propose a contrastive learning approach to improve sample efficiency
in deep RL.2
2. Our method encodes observations into a compact latent space using a
contrastive loss.3
3. We then use this representation to guide policy learning.4
4. Reinforcement learning agents often require large amounts of data to
achieve high performance.1



**“Contrastive Representation Learning for Sample-Efficient
Deep Reinforcement Learning”**

SENTENCE ORDER AWARE



09

DÁNH GIÁ MODEL



RNN-BASED MODELS (GRU VÀ LSTM):

- Hiệu suất yếu: Các mô hình RNN (GRU, LSTM) cho thấy khả năng sinh tiêu đề kém, đặc biệt với điểm ROUGE-2 gần bằng 0, phản ánh độ chính xác n-gram rất thấp.
- Ngữ nghĩa chấp nhận được: BERTScore đạt mức khá (~0.8085), cho thấy đầu ra có ý nghĩa ngữ nghĩa nhưng không bám sát nội dung gốc, thiếu sự liên quan chi tiết.
- So sánh GRU và LSTM: LSTM với GloVe nhỉnh hơn GRU về hiệu suất, nhờ tận dụng biểu diễn từ chất lượng cao, nhưng vẫn không thể cạnh tranh với Transformer.
- Hạn chế tổng quát: RNN gặp khó khăn trong việc nắm bắt các phụ thuộc dài hạn, dẫn đến hiệu quả thấp so với các mô hình hiện đại hơn

TRANSFORMER-BASED MODELS:

- BART-base dẫn đầu: BART-base vượt trội trong việc sinh tiêu đề, tạo ra kết quả chính xác, sát nội dung và phong cách của các bài báo khoa học.
- Flan-T5 hiệu quả: Flan-T5-base gần với BART về độ chính xác, phù hợp cho các tác vụ yêu cầu đầu ra ngắn gọn và đúng trọng tâm.
- Fine-tuned BART vượt zero-shot: Sau fine-tuning, BART-base vượt qua Grok3 và GPT-4o ở độ chính xác dài hạn (ROUGE-L), dù GPT-4o mạnh hơn về n-gram ngắn (ROUGE-2).

TRANSFORMER-BASED MODELS:

- Pegasus-XSum trung bình: Pegasus-XSum tạo tiêu đề ngắn gọn nhưng kém sáng tạo, không đạt hiệu quả cao như BART hay Flan-T5.
- GPT-2 yếu kém: GPT-2 cho thấy hiệu suất thấp nhất trong nhóm, thiếu độ chính xác và khả năng bám sát ngữ cảnh khoa học.

FINE-TUNED LLM (LLAMA, MIXTRAL, DEEPSEEK, PHI-4):

- Llama 3.2 1B hiệu quả nhẹ: Llama 3.2 1B (0.5 cosine, Unslot, LoRA) phù hợp cho môi trường tài nguyên hạn chế, với hiệu suất tốt trong các tác vụ đơn giản.
- Llama 3.1 8B mạnh n-gram: Llama 3.1 8B Instruct nổi bật về độ chính xác n-gram, thích hợp cho các tác vụ yêu cầu tiêu đề chi tiết.
- Llama 3.2 3B hạn chế: Llama 3.2 3B Instruct kém nhất trong nhóm, thiếu độ chính xác n-gram dù vẫn duy trì ngữ nghĩa ở mức chấp nhận được.

FINE-TUNED LLM (LLAMA, MIXTRAL, DEEPSEEK, PHI-4):

- DeepSeek nổi bật: DeepSeek-R1-Distill-Llama-8B dẫn đầu nhờ kiến trúc tối ưu, cân bằng tốt giữa độ chính xác n-gram và ngữ nghĩa, phù hợp cho các tác vụ phức tạp.
- Phi-4 mạnh ngữ nghĩa: Phi-4 tạo tiêu đề có ngữ nghĩa sâu, lý tưởng cho các ứng dụng yêu cầu hiểu biết chi tiết, nhưng ROUGE thấp hơn DeepSeek.
- Mixtral: Mixtral-7B-Instruct cho kết quả tốt về ngữ nghĩa, là lựa chọn thay thế hiệu quả cho DeepSeek.

COMPARISON OF FINE-TUNING STRATEGIES:

- Cosine Similarity-based Data Trimming:
 - Hiệu quả tối ưu tại 0.70: Ngưỡng 0.70 giúp loại bỏ dữ liệu dư thừa, giữ mẫu đa dạng, cải thiện chất lượng học mà không làm mất thông tin quan trọng.
 - Lọc quá mức gây hại: Vượt ngưỡng 0.70 (như 0.80) làm mất mẫu giá trị, giảm khả năng tổng quát hóa của mô hình.
 - Cân bằng dữ liệu: Chiến lược này đảm bảo tập dữ liệu gọn gàng, tăng hiệu suất huấn luyện mà không cần tăng quy mô dữ liệu.

COMPARISON OF FINE-TUNING STRATEGIES:

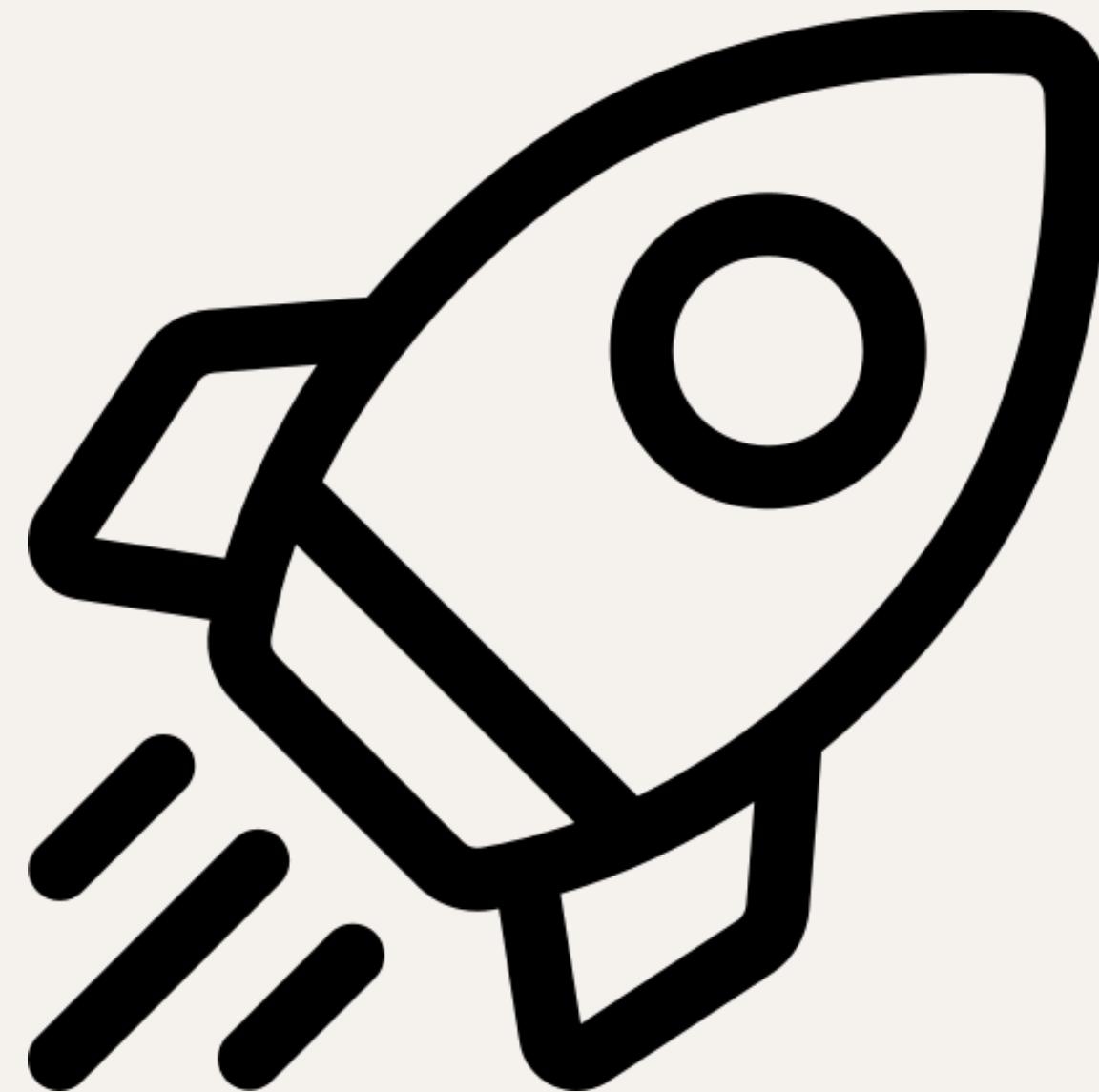
- Keyword-aware Instruction Fine-Tuning:
 - Tăng độ liên quan: Sử dụng KeyBERT để trích xuất từ khóa giúp mô hình tập trung vào khái niệm chính, cải thiện độ chính xác và liên quan của tiêu đề.
 - SciBERT + BART hiệu quả: Kết hợp SciBERT-scivocab-uncased và BART-base đạt kết quả vượt trội, đặc biệt về ROUGE-L và BERTScore.
 - Bền vững với dữ liệu nhỏ: Huấn luyện từ khóa trên tập lớn và đa nhiệm trên tập nhỏ vẫn duy trì hiệu suất cao, phù hợp cho môi trường dữ liệu hạn chế.
 - Tăng tính sáng tạo: Phương pháp này giúp tiêu đề sát nội dung hơn, đồng thời giữ được sự ngắn gọn và phong cách khoa học.

KẾT LUẬN:

- BART-base và DeepSeek dẫn đầu về hiệu suất tổng thể, phù hợp cho các ứng dụng thực tế.
- Cosine Similarity (0.70) và Keyword-aware (SciBERT + BART) tối ưu hóa huấn luyện, cải thiện độ chính xác và liên quan.
- Llama 3.2 1B phù hợp cho ứng dụng nhẹ, trong khi Phi-4 và Llama 3.1 8B lý tưởng cho các tác vụ yêu cầu ngữ nghĩa sâu.

10

DEPLOY





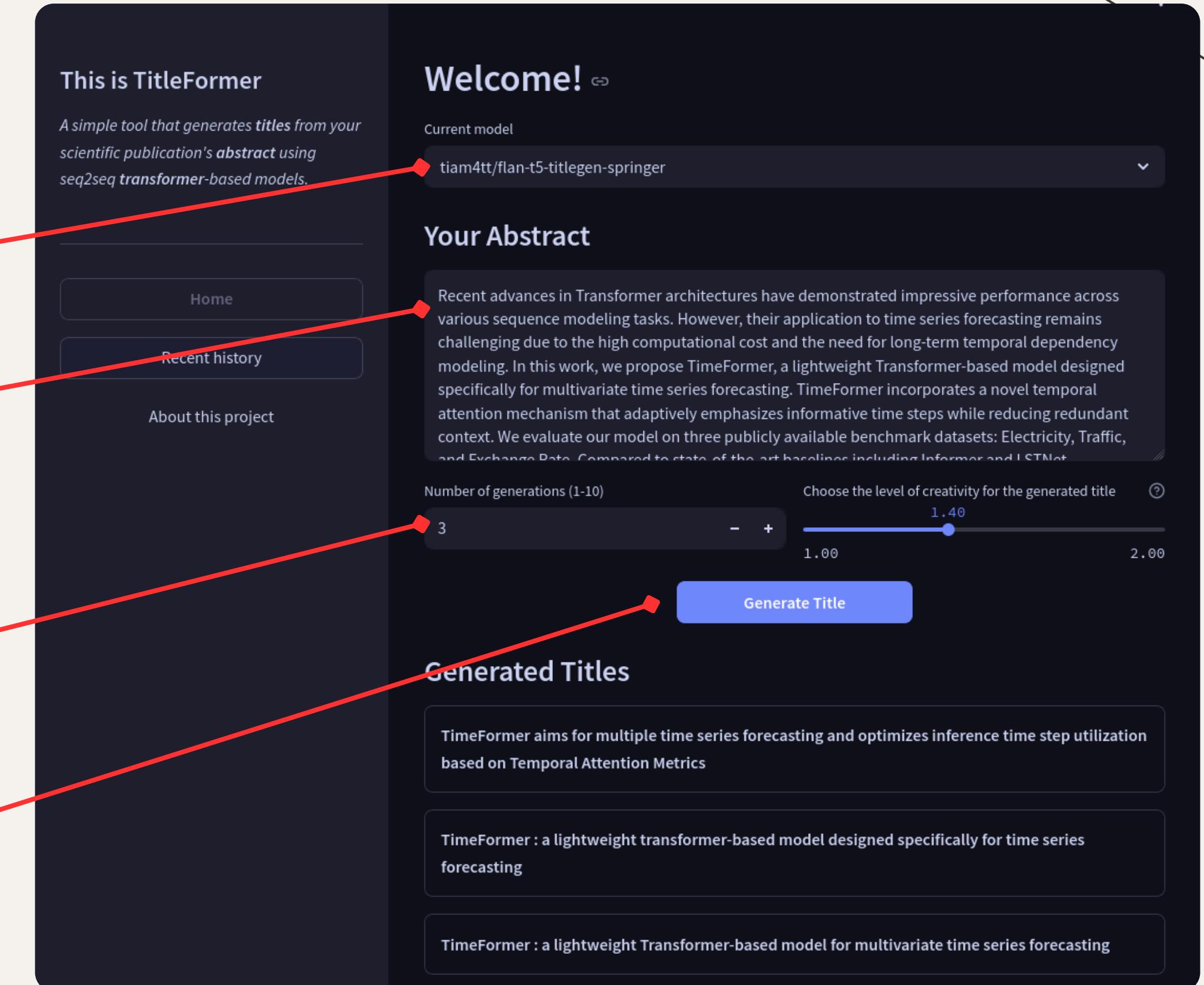
TÍNH NĂNG

1. Chọn mô hình

2. Nhập vào tóm tắt

3. Chọn số tiêu đề sẽ
được tạo ra và mức độ
sáng tạo của tiêu đề.

4. Tạo ra tiêu đề.



THANKS

FOR

WATCHING