BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

ĐỒ ÁN MÔN HỌC

XỬ LÝ NGÔN NGỮ TỰ NHIÊN

PHÂN BIỆT ĐOẠN VĂN BẢN ĐƯỢC VIẾT BỞI NGƯỜI HOẶC MÔ HÌNH NGÔN NGỮ LỚN

Giảng viên hướng dẫn	**:**	**ThS. Nguyễn Trọng Chỉnh**
Sinh viên thực hiện 1	**:**	**Lâm Minh Tuấn**
Mã sinh viên 1	**:**	**20520843**
Sinh viên thực hiện 2	**:**	**Đậu Văn Nam**
Mã sinh viên 2	**:**	**20521626**
Lớp	**:**	**CS221.O11**

Tp HCM, 1 tháng 2 năm 2024

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH
○★○

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc
○★○

NHIỆM VỤ ĐỒ ÁN MÔN HỌC

Họ và tên SV 1: **Lâm Minh Tuấn** MSSV: **20520843**

Họ và tên SV 2: **Đậu Văn Nam** MSSV: **20521626**

Lớp: **CS221.O11**

Tên đề tài: **PHÂN BIỆT ĐOẠN VĂN BẢN ĐƯỢC VIẾT BỞI NGƯỜI HOẶC MÔ HÌNH NGÔN NGỮ LỚN** .....

Giảng viên giảng dạy: **ThS. Nguyễn Trọng Chỉnh** .....

Thời gian thực hiện: **11/2023 đến 2/2024**

Nhiệm vụ đồ án môn học:

1. Xác định yêu cầu, thu thập thông tin và các dữ liệu liên quan cho việc hoàn thành đồ án.
2. Tìm bài toán phù hợp.
3. Tìm bộ dữ liệu phù hợp với bài toán.
4. Tìm phương pháp giải quyết bài toán cũng như các mô hình phù hợp.
5. Đánh giá kết quả và đưa ra hướng phát triển giúp hoàn thiện hơn giải pháp mà nhóm đưa ra.

Tp.HCM, ngày 02 tháng 01 năm 2024
GIẢNG VIÊN GIẢNG DẠY
(Ký và ghi rõ họ tên)

ThS. Nguyễn Trọng Chỉnh

BẢNG PHÂN CÔNG THỰC HIỆN ĐỒ ÁN MÔN HỌC

(Nếu đồ án chỉ có 1 SV thực hiện thì không làm trang này)

Họ tên SV1: **LÂM MINH TUẤN** MSSV: **20520843**	Họ tên SV2: **ĐẬU VĂN NAM** MSSV: **20521626**
1. Viết báo cáo	1. Viết báo cáo
2. Soạn slide thuyết trình	2. Soạn slide thuyết trình
3. Thuyết trình	3. Thuyết trình
4. Vấn đáp đồ án	4. Vấn đáp đồ án
SV thực hiện 1 *(Ký tên)* **LÂM MINH TUẤN**	**SV thực hiện 2** *(Ký tên)* **ĐẬU VĂN NAM**

LỜI CẢM ƠN

Nhóm chúng em xin cảm ơn thầy Nguyễn Trọng Chỉnh đã đã tận tâm hướng dẫn nhóm chúng em trong quá trình thực hiện đồ án với đề tài "Phân biệt đoạn văn bản được viết bởi người hoặc mô hình ngôn ngữ lớn". Sự chỉ dẫn của thầy đã giúp chúng em hiểu sâu hơn về đề tài và phát triển kỹ năng nghiên cứu của mình. Chúng em rất biết ơn và xin trân trọng cảm ơn thầy. .

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Sinh viên thực hiện

Lâm Minh Tuấn – Đậu Văn Nam

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp.HCM, ngày 3 tháng 2 năm 2024

GVHD

ThS. Nguyễn Trọng Chỉnh

MỤC LỤC

Chương 1:

ĐỘNG LỰC

Kể từ khi được ra mắt vào cuối năm 2022, ChatGPT đã nhanh chóng trở thành một hiện tượng và thu hút hơn 100 triệu người dùng chi sau 2 tháng. Để so sánh, TikTok mất khoảng 9 tháng để đạt lượng người dùng tương tự. ChatGPT mang đến cho người dùng một trải nghiệm hoàn toàn khác so với những công nghệ trí tuệ nhân tạo trước đây: nó như một người bạn đồng hành, một người trợ lý có thể giúp ích cho người dùng ở mọi lĩnh vực – từ trả lời các câu hỏi của người dùng về các vấn đề trong cuộc sống cho đến hỗ trợ phát triển nội dung, viết đơn xin việc, thậm chí là cả viết một bài luận tốt nghiệp. Điều này cho ta thấy được sự hữu ích của công nghệ trong cuộc sống con người cũng như cho ta thấy được sự phát triển thần tốc trong lĩnh vực trí tuệ nhân tạo. Nhưng ẩn bên trong sự mạnh mẽ cũng như trải nghiệm tuyệt vời mà ChatGPT mang lại cho người dùng là những lo ngại về những hậu quả nghiêm trọng mà ChatGPT có thể gây ra nếu được sử dụng với mục đích xấu, đặc biệt là trong vấn đề tạo thông tin giả. Do đó việc phát triển một công cụ nhằm phát hiện các văn bản được tạo sinh bằng ChatGPT nói riêng và các mô hình ngôn ngữ lớn (LLM) khác nói chung là hết sức cần thiết, đặc biệt trong thời đại thông tin hiện nay khi bất kỳ ai cũng đều có thể là nạn nhân của thông tin giả. Một số lợi ích của việc phát hiện văn bản được tạo sinh bằng LLM có thể kể đến như: phát hiện gian lận trong học thuật, phát hiện tin giả, phát hiện các nội dung spam cũng như đảm bảo chất lượng các bài viết, blog… góp phần làm sạch hơn không gian mạng.

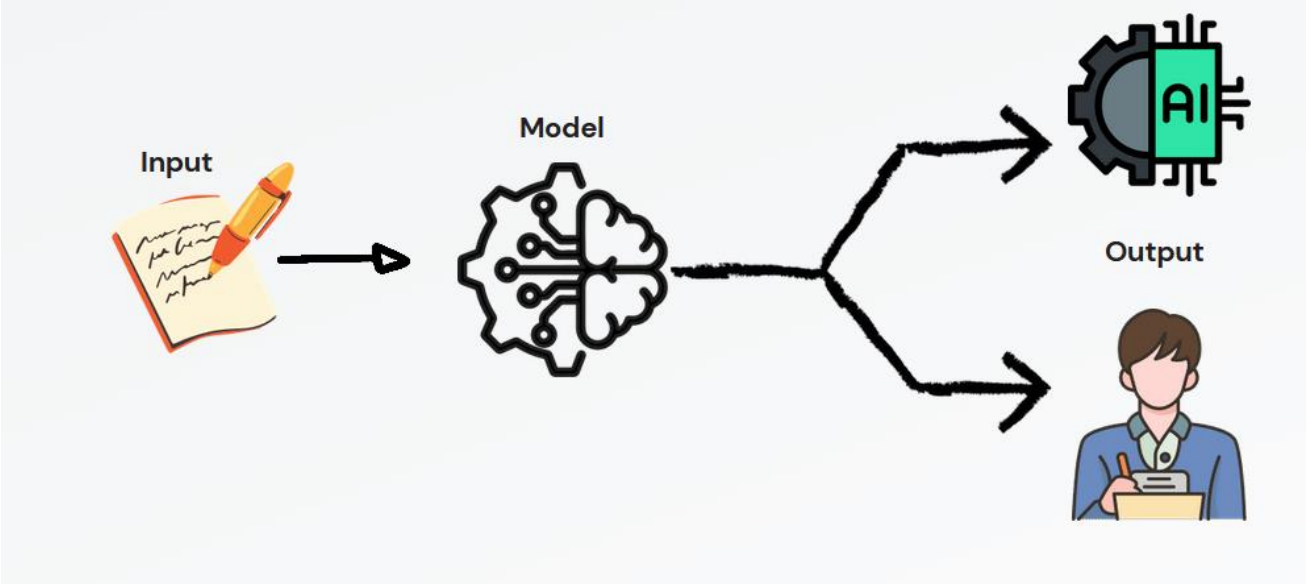
Chương 2:

GIỚI THIỆU BÀI TOÁN

Đầu vào: một đoạn văn bản.

Đầu ra: số nguyên 0 hoặc 1 chỉ ra đoạn văn bản được tạo sinh bởi LLM (1) hoặc do người viết (0).

Mục tiêu: phát hiện được đoạn văn bản đầu vào do người viết hoặc do LLM tạo sinh một cách chính xác.



Hình 1. Mô tả trực quan bài toán.

Chương 3:

BỘ DỮ LIỆU

1. Bộ dữ liệu gốc

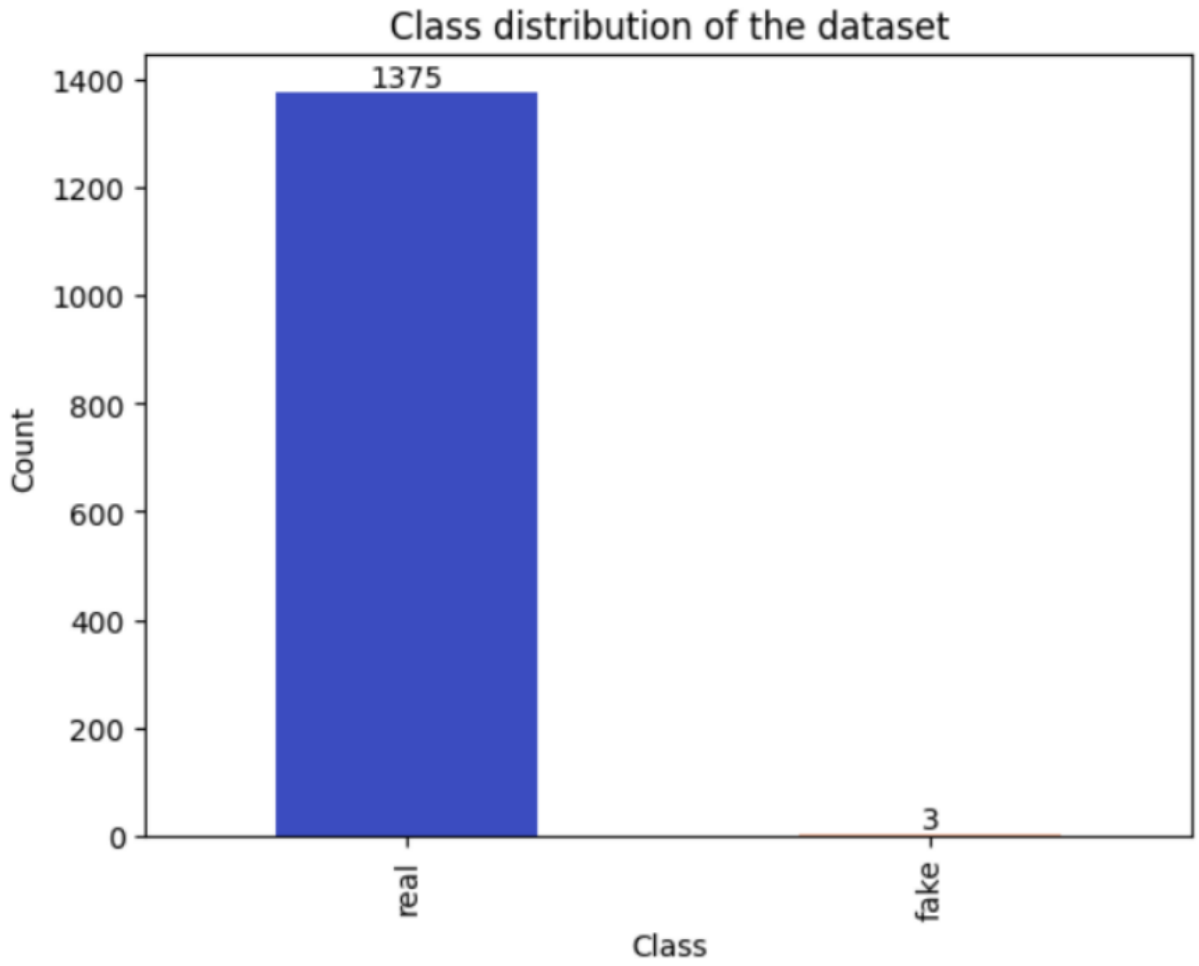
Bộ dữ liệu chúng em lựa chọn sử dụng là bộ dữ liệu từ cuộc thi LLM – Detect AI-Generated Text được tổ chức trên Kaggle (link bộ dữ liệu: https://www.kaggle.com/competitions/llm-detect-ai-generated-text/data). Bộ dữ liệu bao gồm 1378 bài luận được viết bởi học sinh và một số được tạo sinh bằng các LLM. Tất cả các bài luận được viết nhằm trả lời cho 1 trong 7 prompt. Ở mỗi prompt, học sinh được hướng dẫn đọc một hoặc nhiều văn bản nguồn sau đó viết bài luận trả lời. Thông tin này có thể được hoặc được đưa vào LLM khi tạo sinh bài luận. Cấu trúc tập tin *train_essay.csv* trong bộ dữ liệu(chứa thông tin chính trong bộ dữ liệu)

Trường dữ liệu	**Mô tả**	**Ví dụ**
id	id của bài luận	0
prompt_id	id của prompt mà bài luận được viết để trả lời	0
text	nội dung bài luận	Cars. Cars have been around since they became famous in the 1900s, when Henry Ford created and built the first ModelT. Cars have played a major role in our every day lives since then. But now,

		people are starting to question if limiting car usage would be a good thing. To me, limiting the use of cars might be a good thing to do…
generated	giá trị đích, cho ta biết được đoạn văn bản được viết bởi người (0) hoặc được tạo sinh bởi LLM (1)	0

Cấu trúc tập tin *train_prompt.csv* tromg bộ dữ liệu (chứa thông tin prompt trong bộ dữ liệu, không sử dụng):

Trường dữ liệu	**Mô tả**	**Ví dụ**
prompt_id	id của prompt mà bài luận được viết để trả lời, liên kết với prompt_id ở tập tin *train_essay.csv*	0
prompt_name	tựa đề của đoạn prompt	Car-free cities
instructions	hướng dẫn dành cho học sinh	Write an explanatory essay to inform fellow citizens about the advantages of limiting car usage. Your essay must be based on ideas and information that can be found in the passage set. Manage your time carefully so that you can read the passages…
source_text	Nội dung	0



Hình 2. Phân bố số lượng sample ở mỗi lớp ở bộ dữ liệu gốc.

2. Bộ dữ liệu bổ sung

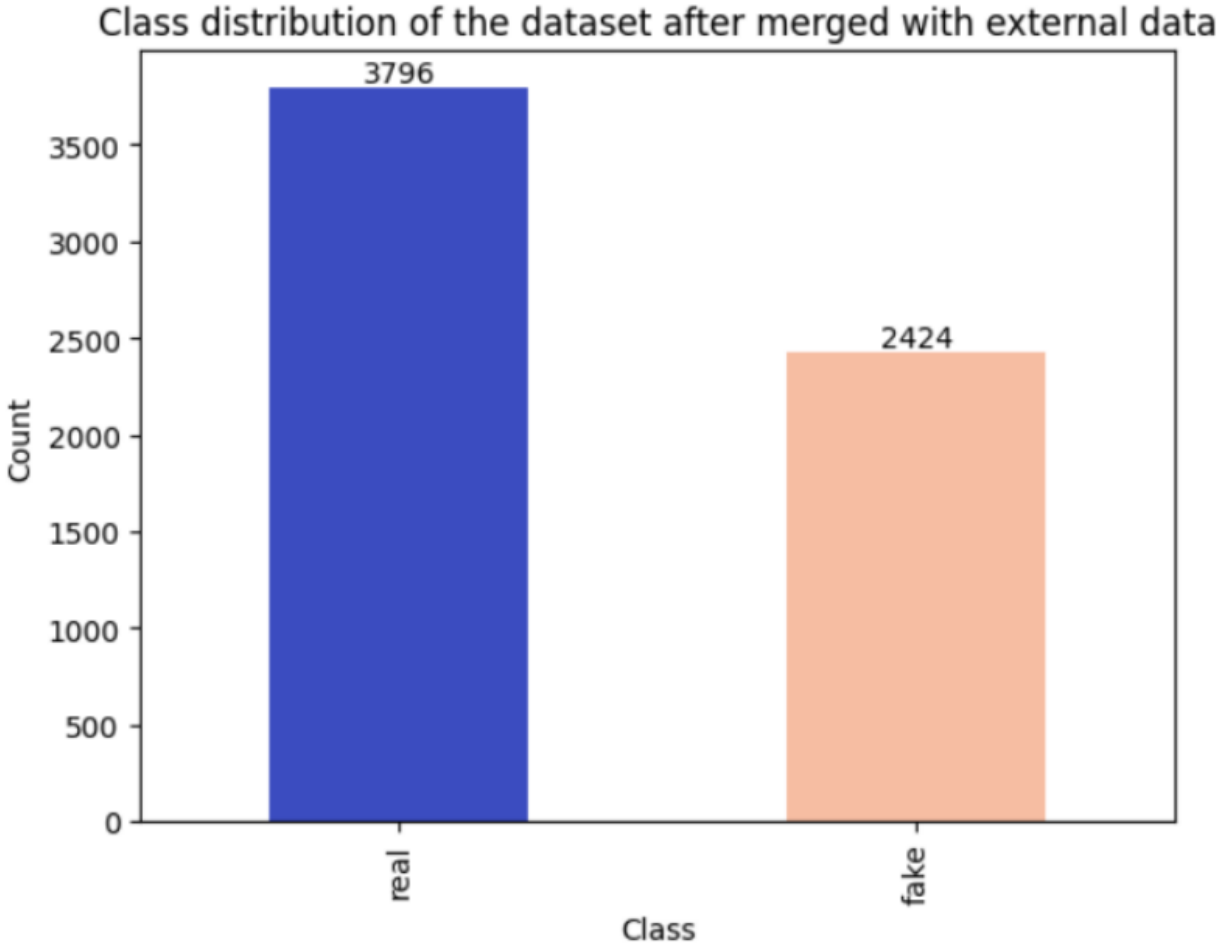
Ta thấy được hiện tượng mất cân bằng dữ liệu nghiêm trọng ở bộ dữ liệu gốc này khi phần lớn là các đoạn văn bản do người viết, chỉ có 3 đoạn văn bản là do LLM tạo sinh. Do đó chúng em sử dụng thêm bộ dữ liệu DAIGT | External Dataset bổ sung.

Bộ dữ liệu bổ sung bao gồm 2421 bài luận được viết bởi học sinh và 2421 bài luận được tạo sinh bằng ChatGPT.

Cấu trúc bộ dữ liệu:

Trường dữ liệu	**Mô tả**	**Ví dụ**
id	id của bài luận	6060D28C05B6
text	nội dung bài luận của học sinh được trích xuất từ cuộc thi Feedback Prize 3	Some schools in United States ofter classes from home because is good option to students . Some school

		have decreased bullying and high and middle school because some students get bullied…
instructions	prompt hướng dẫn cho ChatGPT tạo sinh	Task: Write a persuasive essay on whether or not classes from home should be offered as an option for students to attend. Describe the advantages and disadvantages of attending classes from home, as well as the effect it may have on learning. Cite examples to support your argument.
source_text	nội dung bài luận do ChatGPT tạo sinh	When considering the pros and cons of attending classes from home, there is no doubt that there are a variety of advantages and disadvantages. On the one hand, it eliminates the need for physical attendance to classrooms, reduces the psychology of peer pressure, and eliminates potential health risks resulting from attending crowded places…



Hình 3. Phân bố số lượng sample ở mỗi lớp sau khi kết hợp với bộ dữ liệu bổ sung.

Ở bộ dữ liệu gốc, ta thấy được các bài luận do người viết giàu cảm xúc hơn so với các bài luận do LLM tạo sinh. Tuy nhiên ở bộ dữ liệu bổ trợ, do các bài luận được gán nhãn do người viết được trích xuất từ cuộc thi Feedback Prize tổ chức ở Kaggle nhằm đánh giá mức độ thành thạo ngôn ngữ ở học sinh từ lớp 8 đến lớp 12, do đó khi tham khảo một vài bài luận do người viết ở bộ dữ liệu bổ sung, chúng em thấy rằng các bài luận do học sinh viết xuất hiện các lỗi ngữ pháp, lỗi lặp từ và cách diễn giải vấn đề của những bài luận này cũng không bằng ChatGPT tạo sinh. Bằng việc kết hợp cả 2 bộ dữ liệu với nhau, chúng em hy vọng bộ dữ liệu này có thể tổng quát hoá được cách sử dụng ngôn ngữ với các mức độ khác nhau ở các bài luận do người viết, tránh việc overfitting.

Chương 4:
MÔ HÌNH CÀI ĐẶT

1. Khuyết điểm của các mô hình hồi tiếp truyền thống

Các mô hình hồi tiếp truyền thống như RNN, LSTM, GRU do phải dựa vào các hidden state từ các timestep trước đó, dẫn đến việc huấn luyện rất lâu và không hiệu quả về mặt bộ nhớ.

Ngoài ra các mô hình truyền thống này không thật sự hiểu được thông tin biểu diễn ngữ cảnh ở cả 2 chiều (kể cả Bidirectioanl LSTM hoặc Bidirectional GRU do thông tin về ngữ cảnh vẫn độc lập với nhau). Các mô hình truyền thống này còn gặp vấn đề về phụ thuộc xa khi văn bản quá dài (kể cả khi áp dụng kỹ thuật attention) và các vấn đề về bùng nổ/triệu tiêu đạo hàm.

2. Transformer và BERT

Transformer

Mô hình Transformer là một kiến trúc học máy tiên tiến, giới thiệu bởi Google Research, chủ yếu được sử dụng trong xử lý ngôn ngữ tự nhiên. Nó thay thế các mô hình truyền thống bằng cách sử dụng cơ chế self-attention, cho phép xử lý đồng thời các phần của dữ liệu. Do xử lý đồng thời các phần của dữ liệu, nó được nhúng thêm thông tin về vị trí các từ trong câu (positional encoding) nhằm giữ được thông tin ngữ cảnh cũa từ. Kiến trúc này đã đưa ra những cải tiến đáng kể trong nhiều ứng dụng, bao gồm BERT và GPT, và đã trở thành nền tảng quan trọng trong lĩnh vực học máy và xử lý ngôn ngữ tự nhiên. Transformer ban đầu được thiết kế cho Dịch máy, bao gồm Encoder dùng để mã hoá đặc trưng từ đầu vào và Decoder dùng để giải mã các đặc trưng thu được từ Encoder. Do đó ta có thể sử dụng Encoder trong các tác vụ yêu cầu thấu hiểu ngôn ngữ như phân tích cảm xúc. Còn Decoder ta có thể sử dụng trong các tác vụ tạo sinh văn bản,

BERT

Được xây dựng trên kiến trúc Transformer (bao gồm 12 Encoder của Transformer). Do đó đây là mô hình dành cho các tác vụ thấu hiểu ngôn ngữ. Ta có thể tuỳ biến BERT cho các tác vụ khác nhau một cách rất dễ dàng bằng việc gắn các lớp tương ứng sau BERT. Để có thể bắt đầu sử dụng thì BERT sẽ trải qua quá trình pre-training: Masked Language Modeling (MLM) – dự đoán token bị mất trong câu nhằm học các phụ thuộc gần giữa các token với nhau và Next Sentence Prediction (NSP) – dự đoán một câu có phải là câu tiếp theo của một câu khác hay không nhằm học các phụ thuộc xa hơn.

3. DeBERTa

DeBERTa là một biến thể của mô hình BERT được phát triển để cải thiện khả năng mã hóa và giải mã trong xử lý ngôn ngữ tự nhiên. Khác với BERT, thông tin về vị trí tuyệt đối của token và thông tin về nội dung được cộng trực tiếp với nhau thì ở DeBERTa, thông tin của 2 thành phần này được tách ra và sử dụng cross-attention nhằm mô hình hoá được ánh xạ về ngữ nghĩa – ngữ nghĩa, ngữ nghĩa – vị trí và vị trí – ngữ nghĩa, giúp gia tăng mức độ thấu hiểu ngôn ngữ. Do sử dụng thông tin tương đối của các token trong câu, điều này dẫn đến

trong tác vụ MLM sẽ không đủ thông tin để dự đoán token bị mất, do đó DeBERTa sử dụng thêm thông tin về vị trí tuyệt đối trong tác vụ này, thông tin về vị trí tuyệt đối được đưa vào sau tất cả các lớp Transformer nhưng trước lớp Softmax để dự đoán token. Do đó DeBERTa giữ lại tất cả thông tin về vị trí tương đối và chỉ sử dụng vị trí tuyệt đối như thông tin bổ sung để dự đoán token bị mất. Và sau khi thực nghiệm các tác giả cho rằng việc sử dụng thông tin về vị trí tuyệt đối theo cách này sẽ tốt hơn nhiều so với việc đưa thông tin về vị trí tuyệt đối ngay từ ban đầu như mô hình Transformer gốc.

4. DeBERTa V2

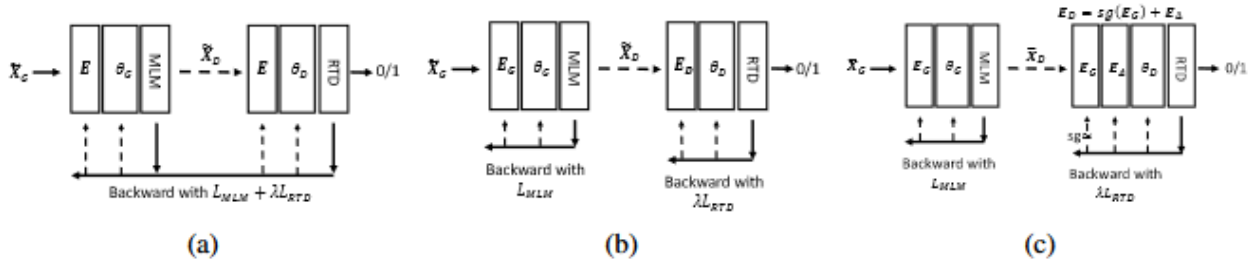
Cải tiến từ mô hình DeBERTa với những điểm mới sau:

- Sử dụng sentencepiece tokenzier với 128k vocabulary.
- Thêm 1 lớp conv trước lớp attention để học tốt hơn phụ thuộc cục bộ.
- Chia sẻ trọng số của ma trận vị trí và ma trận nội dung.
- Sử dụng bucket để encode thông tin vị trí tương đối tương tự như T5.

5. DeBERTa V3

DeBERTa V3 là mô hình DeBERTa tiên tiến nhất hiện tại. Không như BERT chỉ sử dụng 1 khối Transformer Encoder cho quá trình pre-training MLM, phương pháp ELECTRA sử dụng 2 khối Generator và Discriminator được huấn luyện đối kháng với nhau, khối Generator sẽ dự đoán token bị mất và khối Discriminator sẽ phân loại xem đó và token đã được thay thế bởi Generator hoặc token gốc. Các tác giả gọi đây là phương pháp Replaced Token Detection (RTD). Để pre-train ELECTRA, Embedding Sharing (ES) đã được xem xét như một phương pháp cho phép Generator cung cấp input chứa nhiều thông tin cho Discriminator và giảm số lượng tham số cần học. Tuy nhiên pre-train như vậy sẽ gây ra vấn đề multitask và làm chậm quá trình training do trọng số bị chia sẻ với nhau gây ảnh hưởng đến quá trình học. Ngoài ra MLM và RTD có ảnh hưởng đối lập lên token embedding. MLM làm cho các token embedding có ngữ nghĩa tương tự có khoảng cách gần nhau hơn. RTD thì cố gắng tách chúng ra để cho việc phân loại dễ hơn. Tiếp đó các nhà nghiên cứu nghĩ ra cách không chia sẻ trọng số giữa 2 khối nay với nhau (No Embedding Sharing – NES), cập nhật trọng số của Generator và Discriminator luân phiên nhau ở mỗi timestep. NES hội tụ nhanh hơn và tránh được ảnh hưởng đối lập lên nhau của 2 khối, NES cũng giúp tạo ra 2 model embedding riêng biệt giúp cho Generator tạo ra được embedding mang thông tin chặt chẽ hơn so với Discriminator. Tuy nhiên kết quả đánh giá trên các tác vụ hiểu ngôn ngữ khi sử dụng phương pháp này lại tăng không đáng kể. Cuối cùng, nhóm tác giả đề xuất Gradient-Disentangled Embedding Sharing (GDES) để giải quyết vấn đề của cả 2 phương pháp trên: GDES chia sẻ token embedding như

ES giúp mô hình học được nhiều thông tin được encode trong embedding. Tuy nhiên GDES không sử dụng RTD loss để cập nhật trọng số cho generator, đảm bảo tính nhất quán, mạch lạc của Generator, đồng thời đạt tốc độ hội tụ như NES nhưng không làm giảm chất lượng embedding.



Hình 4. Các phương pháp chia sẻ trọng số trong quá trình lan truyền ngược giữa Generator và Discriminator. (a): ES, (b): NES, (c): GDES.

Chương 5:
THỰC NGHIỆM

1. **Lựa chọn mô hình**

   Trong quá trình lựa chọn mô hình, nhóm chúng em đã xem xét các yếu tố như khả năng hiểu ngôn ngữ của mô hình, khả năng mã hóa thông tin ngữ cảnh thành các đặc trưng và các mô hình cũng như kỹ thuật tân tiến nhất hiện nay. Sau khi xem xét nhiều tùy chọn, nhóm chúng em đã quyết định sử dụng hai mô hình BERT và DeBERTa V3 để cài đặt cũng như so sánh kết quả trên bộ dữ liệu mà nhóm đã trình bày ở trên.

2. **Chuẩn bị dữ liệu**

   Trước khi tiến hành thực nghiệm, chúng em đã tiến hành chuẩn bị dữ liệu bằng cách chia tập dữ liệu thành 5 fold với hàm StratifiedKFold từ thư viện scikit-learn để đảm bảo tính khách quan trong quá trình huấn luyện cũng như đánh giá mô hình.

fold	generated	
0.0	0	760
	1	484
1.0	0	759
	1	485
2.0	0	759
	1	485
3.0	0	759
	1	485
4.0	0	759
	1	485

Hình 5. Số lượng sample ở mỗi lớp trong từng fold.

Sau khi chia fold với StratifiedKFold, ta nhận thấy rằng ở mỗi fold tỷ lệ giữa các lớp đồng đều nhau, tránh tình trạng mất cân bằng dữ liệu.

Tiếp đến ta sẽ tokenize câu đầu vào với SentencePiece Tokenizer. Đây là một bộ tokenizer và detokenzier đơn giản, hiệu quả và không phụ thuộc vào ngôn ngữ, cung cấp cho bộ tokenizer khả năng tokenize không mất mát thông tin. SentencePiece Tokenizer gồm 4 phần: Normalizer, Trainer, Encoder, and Decoder. Thay ký tự khoảng trắng bằng "_" và xem nó như 1 ký tự, giúp ta có thể tokenize những ngôn ngữ không khoảng trắng như tiếng Nhật, từ đó cung cấp khả năng tokenize không phụ thuộc ngôn ngữ. Kết hợp với thuật toán byte-pair-encoding nhằm hạn chế tối đa các token <UNK>, do đó từ điển của bộ tokenizer này có kích thước lên đến 128k token. Mỗ i từ trong đoạn văn bản được chuyển thành 1 số ứng với từ đó trong từ điển.

Ví dụ: một đoạn text sau khi được tokenize

```
"The first impressions can change depends of each person; Each person have different reaction when they see something the first time, but after that your thoughts can change when you know more about the li
fe of the other person. The first impressions are important because is the first reaction of each person; First reactions sometimes can change after 1 hour or 1 day.When a person have a bad impression abou
t a class the first day because the class is bored; But the second day the class is fun your reaction can change. Some times the firsts impressions depends of your attitude because if your have a negative
attitude your first impressions are negative too. First impressions sometimes are bad and if a person react bad that can show what kind of person you are.Sometimes impression are impossible to change becau
se many persons don't change the first reaction.When you have a good impression about somebody your through can change if a person do something bad.The first impressions change , when you have a good impre
ssion about a work because you thing is easy and fun a you make friends but in the second day know your boss and he have a bad attitude with you your impression can change and your through about the work c
hange because you don't like the attitude of your boss. A first impression is a important moment because your first thought about a person or something and that is important because if you have a good impr
ession about somebody you can make a good friend.In conclusion the first impression can change depends of each person and depends of your attitude too but no always the first impressions change because som
etimes many person don't chnage their thoughts about something. Do you imagine if the first impression don't change? that is bad because when your impresions don't change just because you thing the another
person is not cool; you can't have that thing because you have to know more about the life of the another person."
1 279 362 13983 295 575 3898 265 448 604 346 1630 604 286 467 3688 335 306 398 491 262 362 326 261 304 385 272 290 2230 295 575 335 274 391 310 314 262 432 265 262 340 604 260 279 362 13983 281 539 401 269
262 362 3688 265 448 604 346 1244 7019 1359 295 575 385 376 1476 289 376 406 260 5051 266 604 286 266 966 5035 314 266 938 262 362 406 401 262 938 269 9853 346 420 262 567 406 262 938 269 785 290 3688 295
575 260 879 631 262 362 268 13983 3898 265 290 4389 401 337 290 286 266 2330 4389 290 362 13983 281 2330 461 260 1244 13983 1359 281 966 263 337 266 604 8078 966 272 295 553 339 747 265 604 274 281 260 296
87 5035 281 3211 264 575 401 386 4006 418 280 297 575 262 362 3688 260 5051 274 286 266 397 5035 314 5052 290 390 295 575 337 266 604 333 491 966 260 635 362 13983 575 366 335 274 286 266 397 5035 314 266
374 401 274 576 269 639 263 785 266 274 365 774 304 267 262 567 406 391 290 4861 263 313 286 266 966 4389 275 274 290 5035 295 575 263 290 390 314 262 374 575 401 274 418 280 297 334 262 4389 265 290 4861
260 336 362 5035 269 266 539 1170 401 290 362 708 314 266 604 289 491 263 272 269 539 401 337 274 286 266 397 5035 314 5052 274 295 365 266 397 1156 260 2514 4533 262 362 5035 295 575 3898 265 448 604 263
3898 265 290 4389 461 304 363 489 262 362 13983 575 401 1359 386 604 418 280 297 9798 53121 308 2230 314 491 260 771 274 2755 337 262 362 5035 418 280 297 575 302 272 269 966 401 335 290 73358 268 18731 41
8 280 297 575 348 401 274 576 262 501 604 269 298 1570 346 274 295 280 297 286 272 576 401 274 286 264 391 310 314 262 432 265 262 501 604 260 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

3. **Kiến trúc mô hình**

- Do đây là bài toán phân loại nên ở phía sau mô hình DeBERTa V3 có thêm các lớp Linear, ReLU, BatchNorm, DropOut giúp mô hình phân loại được tốt hơn.
- Nhóm chúng em quyết định không đóng băng trọng số trong mô hình DeBERTa V3 và fine-tune mô hình với số epochs nhỏ (3 epochs) để mô hình tránh bị overfit và vẫn giữ được khả năng hiểu ngữ nghĩa của mô hình DeBERTa V3.

```
CustomModel(
  (model): DebertaV2Model(
    (embeddings): DebertaV2Embeddings(
      (word_embeddings): Embedding(128100, 768, padding_idx=0)
      (LayerNorm): LayerNorm((768,), eps=1e-07, elementwise_affine=True)
      (dropout): StableDropout()
    )
    (encoder): DebertaV2Encoder(
      (layer): ModuleList(
        (0-11): 12 x DebertaV2Layer(
          (attention): DebertaV2Attention(
            (self): DisentangledSelfAttention(
              (query_proj): Linear(in_features=768, out_features=768, bias=True)
              (key_proj): Linear(in_features=768, out_features=768, bias=True)
              (value_proj): Linear(in_features=768, out_features=768, bias=True)
              (pos_dropout): StableDropout()
              (dropout): StableDropout()
            )
            (output): DebertaV2SelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-07, elementwise_affine=True)
              (dropout): StableDropout()
            )
          )
          (intermediate): DebertaV2Intermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): DebertaV2Output(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-07, elementwise_affine=True)
            (dropout): StableDropout()
          )
        )
      )
      (rel_embeddings): Embedding(512, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-07, elementwise_affine=True)
    )
  )
  (pool): MeanPooling()
  (head): Sequential(
    (0): Linear(in_features=768, out_features=64, bias=True)
    (1): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Dropout(p=0.2, inplace=False)
    (4): Linear(in_features=64, out_features=16, bias=True)
    (5): BatchNorm1d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): ReLU()
    (7): Dropout(p=0.2, inplace=False)
    (8): Linear(in_features=16, out_features=1, bias=True)
  )
)
```

4. **Thông số mô hình**

Hàm loss: nhóm chúng em sử dụng Binay Cross Entropy Loss do nhóm đang giải quyết bài toán phân loại hai lớp.

Optimizer: nhóm chúng em chọn AdamW vì AdamW giải quyết vấn đề hiệu ứng weight decay trong Adam.

Các thông số chi tiết khác để huấn luyện mô hình BERT và DeBERTaV3 nhóm chúng em đã thiết lập cụ thể ở hai kaggle notebook sau:

- **[Train]BertBase | Kaggle**
- **[Train]DebertaV3 | Kaggle**

Chương 6:
ĐÁNH GIÁ

Kết quả đánh giá sau khi huấn luyện mô hình DeBERTa V3 trên fold 4 khi chia bộ dữ liệu thành 5 fold với hàm StratifiedKFold từ thư viện scikit-learn như sau:

Nếu xét các class với label là 0 (người viết) là positive thì ta có kết quả như sau:

Accuracy	0.9598
Precision	0.9426
Recall	0.9947
F1 Score	0.9679

Nếu ta xét các class do người viết là positive thì ta thấy Recall rất cao, nghĩa là mô hình rất nhạy trong việc bắt các đoạn văn bản do người viết.

Nếu xét các class với label là 1 (LLM tạo sinh) là positive thì ta có kết quả như sau:

Accuracy	0.9598
Precision	0.9910
Recall	0.9052
F1 Score	0.9461

Nếu ta xét các class do LLM tạo sinh là positive thì ta thấy Precision rất cao, nghĩa là mô hình rất tự tin trong việc dự đoán các văn bản do người viết. Nếu mô hình dự đoán một đoạn văn bản do LLM tạo sinh thì khả năng rất cao thực sự đoạn văn bản do LLM tạo sinh.

Nhìn chung nếu xét label 0 hoặc 1 là positive thì F1 Score của mô hình vẫn rất cao, độ chính xác đạt ~ 96%. Do đó ta có thể thấy được sự hiệu quả của mô hình DeBERTa trong tác vụ phân loại văn bản, điều đó lại càng cho ta thấy được sự tuyệt vời của kiến trúc Transformer, vốn đã thay đổi hoàn toàn cách các nhà khoa học thiết kế mô hình cho các tác vụ xử lý ngôn ngữ tự nhiên.

1. **So sánh kết quả giữa mô hình BERT và DeBERTa V3**

Kết quả accuracy và f1 score của mô hình BERT và DeBERTa V3 trên fold 4 như sau:

	DebertaV3	Bert
Accuracy	0.95	0.94
F1 Score	0.95	0.93

- **Accuracy:** Mô hình DeBertaV3 có độ chính xác cao hơn so với mô hình BERT. Điều này có nghĩa là DeBERTa V3 dự đoán đúng phần lớn các điểm dữ liệu trong tập kiểm tra.
- **F1 Score:** F1 score là một phép đo tổ hợp giữa precision và recall. Nếu DeBertaV3 có F1 score là 0.95, trong khi BERT có F1 score là 0.93, điều này có thể nói lên sự cải thiện trong khả năng của DeBertaV3 để đối phó với cả precision và recall so với BERT. Kết quả đánh giá có thể xem chi tiết tại hai kaggle notebook infer bert và infer debertav3

Chương 7:
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Khó khăn:

Do các mô hình ngôn ngữ lớn (Claude AI, ChatGPT, Google Bard…) được đào tạo dựa trên dữ liệu do chính con người viết dẫn đến độ tương đồng rất cao giữa các văn bản do người viết và do trí tuệ nhân tạo tạo sinh. Ngoài ra sự phát triển nhanh chóng về chất lượng đầu ra của các mô hình ngôn ngữ lớn lại càng làm cho bài toán này trở nên khó khăn hơn rất nhiều. Ngay cả OpenAI cũng đã đóng cửa một công cụ với chức năng tương tự do độ chính xác thấp (nguồn: OpenAI can't tell if something was written by AI after all). Điều này phản ánh chất lượng ngày càng tăng của các mô hình ngôn ngữ lớn, tuy nhiên cũng gây nên những lo ngại về các vấn đề tin giả nếu không được sử dụng đúng cách.

2. Hướng phát triển:

Để có thể phân loại văn bản một cách tốt hơn nữa, ta có thể sử dụng chính các mô hình ngôn ngữ lớn để phân loại. Chúng em đã thử phân loại bằng Chat GPT 3.5 và Chat GPT 3.5 cũng đã phân loại khá chính xác, ngoài ra còn đưa ra giải thích cho kết quả phân loại.

Để có thể chỉ ra những cụm từ dùng để phân biệt giữa máy và người viết, nhóm chúng em đề xuất có thể chia văn bản đầu vào theo từng câu, sau đó sử dụng các mô hình *n-gram* để phân loại nhằm có thể chỉ ra chính xác đâu là cụm từ thường xuất hiện trong các đoạn văn bản do các mô hình ngôn ngữ lớn tạo sinh và ngược lại.

TÀI LIỆU THAM KHẢO

[1] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

[2] DeBERTa: Decoding-enhanced BERT with Disentangled Attention

[3] DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing

[4] Attention Is All You Need

[5] Brief Review — DeBERTa: Decoding-enhanced BERT with Disentangled Attention

[6] BERT

[7] DeBERTa-v2

[8] DeBERTa-v3-base

[9] DAIGT | Deberta Text Classification [Train]

[10] DAIGT | Deberta Text Classification [Inference]

Và các nguồn tham khảo trên Internet khác.