

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



NHÓM DEEPSTUDY
ĐỒ ÁN CUỐI KÌ

Deep Learning

Hồ Chí Minh, Ngày 05 tháng 01 năm 2025

THÔNG TIN ĐỒ ÁN

GIẢNG VIÊN HƯỚNG DẪN: ThS. Nguyễn Trần Duy Minh

THÔNG TIN NHÓM

MSSV	Họ và tên
21120036	Nguyễn Hoài An
21120103	Phan Thảo Nguyên
21120179	Nguyễn Đăng Đăng Khoa
21120275	Huỳnh Cao Khôi
21120308	Phạm Lê Tú Nhi

THỜI HẠN ĐỒ ÁN: 10/11/2024 - 20/01/2025

LỜI CẢM ƠN:

Chúng em xin gửi lời cảm ơn tới TS. Nguyễn Tiến Huy vì đã hướng dẫn tận tình và đưa lại kiến thức cần thiết cũng như làm nguồn cảm hứng để học hỏi thêm về Deep Learning riêng và về Khoa học máy tính/Khoa học dữ liệu nói chung. Chúng em cũng xin gửi lời cảm ơn ThS. Nguyễn Trần Duy Minh vì những hướng dẫn kỹ lưỡng và những lời khuyên của thầy trong quá trình học.

Mục lục

1	Giới thiệu nhóm	5
1.1	Giới thiệu thành viên	5
1.2	Phân công công việc	5
2	Tổng quan đồ án	6
2.1	Kaggle Competition: Eedi - Mining Misconception in Maths	6
2.2	Hướng tiếp cận	6
3	Khám phá dữ liệu	7
3.1	Tổng quan về dữ liệu	7
3.1.1	Dữ liệu trong <code>train.csv</code>	7
3.1.2	Dữ liệu trong <code>test.csv</code>	8
3.1.3	Dữ liệu trong <code>misconception.csv</code>	8
3.1.4	Dữ liệu trong <code>sample_submission.csv</code>	9
3.2	Khám phá dữ liệu dựa trên quan sát và thống kê	10
3.2.1	Dữ liệu trong <code>train.csv</code>	10
3.2.1.1	Kiểu dữ liệu và dữ liệu bị thiếu	10
3.2.1.2	Khám phá <code>CorrectAnswer</code>	12
3.2.1.3	Khám phá <code>Misconception[A/B/C/D] Id</code>	13
3.2.1.4	Khám phá <code>Subject</code>	16
3.2.1.5	Khám phá <code>Construct</code>	20
3.2.1.6	Khám phá mối quan hệ giữa <code>Subject</code> , <code>Construct</code> và <code>Misconception</code>	25
3.2.1.7	Khám phá <code>QuestionText</code>	27
3.2.1.8	Khám phá <code>Answer[A/B/C/D]Text</code>	32
3.2.2	Dữ liệu trong <code>misconception_mapping.csv</code>	36
3.3	Khám phá ngữ nghĩa	39
3.3.1	Phương pháp thực hiện	39
3.3.2	Phương pháp xử lý dữ liệu	40
3.3.3	Khám phá dữ liệu	40
3.3.3.1	Khám phá <code>Subject</code>	40

3.3.3.2	Khám phá Construct	48
3.3.3.3	Khám phá Misconception	53
3.3.3.4	Tổng hợp insights	57
4	Xây dựng mô hình	59
4.1	Bài toán: Information Retrieval	59
4.2	Stage 1: Retrieval	61
4.2.1	Phân tích bài toán	61
4.2.2	Quá trình xây dựng mô hình	61
4.2.2.1	Chuẩn bị dữ liệu	61
4.2.2.2	Biểu diễn dữ liệu bằng embedding	63
4.2.2.3	Tính toán độ tương đồng	63
4.3	Stage 2: Re-ranking	64
4.3.1	Mô hình Qwen 2.5 32B	64
4.3.2	Bộ xử lý MultipleChoiceLogits	65
5	Thử nghiệm	67
5.1	Các cách tiếp cận trước của nhóm	67
5.2	Kết hợp nhiều embedding khác nhau	67
5.3	Kết hợp quá trình re-rank nhiều lần	68
5.4	Fine tune mô hình ngôn ngữ	69
5.5	Data Augmented	70
6	Tổng kết	77
6.1	Kết quả trên Kaggle	77
6.2	Tự nhận xét	77
6.3	Hướng đi trong tương lai	77
	Tài liệu tham khảo	79

1 Giới thiệu nhóm

1.1 Giới thiệu thành viên

Nhóm có 5 thành viên thuộc chuyên ngành Khoa học dữ liệu. Trong quá trình thực hiện đồ án, tất cả các thành viên thực hiện công việc đầy đủ và cộng tác trong các tác vụ liên quan.

1.2 Phân công công việc

Phân công công việc cho đồ án được trình bày trong bảng sau:

Công việc	Phân công
Tìm hiểu Data Discovery	Nguyễn Đặng Đăng Khoa
Tìm hiểu Data Discovery	Phạm Lê Tú Nhi
Tìm hiểu về Model Training	Nguyễn Hoài An
Tìm hiểu về Model Training	Phan Thảo Nguyên
Tìm hiểu về Model Training	Huỳnh Cao Khôi
Tổng hợp notebook Model Training	Nguyễn Hoài An
Tổng hợp notebook EDA	Nguyễn Đặng Đăng Khoa
Tổng hợp notebook EDA	Phạm Lê Tú Nhi
Chuẩn bị Slides	Phan Thảo Nguyên
Chuẩn bị Slides	Huỳnh Cao Khôi
Chuẩn bị Slides	Phạm Lê Tú Nhi
Thuyết trình Slides	Phạm Lê Tú Nhi
Viết report về nội dung đã tìm hiểu	Cả nhóm

2 Tổng quan đề án

2.1 Kaggle Competition: Eedi - Mining Misconception in Maths

Bối cảnh: Eedi là một nền tảng giáo dục mà người học trả lời các câu hỏi Diagnostic Question (Câu hỏi chuẩn đoán). Mỗi câu hỏi thử thách người học về một kỹ năng nhất định. Có 4 lựa chọn cho 1 câu hỏi, trong đó 1 lựa chọn là đáp án đúng và 3 lựa chọn sai. Mỗi lựa chọn sai tương ứng với một Misconception.

Yêu cầu cuộc thi: Phát triển mô hình NLP để dự đoán Misconception và các câu trả lời sai trong các câu hỏi trắc nghiệm có chủ đề toán học.

Ý nghĩa: Việc hỗ trợ gợi ý các Misconception sẽ giúp các giáo viên chuyên môn dễ dàng label các Misconception cho câu hỏi mới, từ đó giúp cải thiện quá trình quản lý Misconception, nâng cao trải nghiệm giáo dục của người dạy và người học.

2.2 Hướng tiếp cận

Nhóm tiếp cận giải quyết vấn đề của bài toán bằng cách thực hiện 2 công việc một cách song song:

- **Khám phá dữ liệu:** Các thành viên trong nhóm thực hiện khám phá dữ liệu, sử dụng các phương pháp hiện thống kê, khác... và trực quan để hiểu thêm về tính chất của dữ liệu.
- **Xây dựng mô hình:** Các thành viên trong nhóm thực hiện thử nghiệm xây dựng mô hình học máy và submit vào cuộc thi để đánh giá chất lượng. Việc tìm kiếm mô hình thử nghiệm có thể dựa vào kinh nghiệm của thành viên, hoặc bằng cách khảo sát các mô hình được dùng phổ biến giữa các thí sinh tham gia cuộc thi.

Trong quá trình thực hiện đề án, nhóm tổ chức họp và kiểm tra tiến độ của các thành viên mỗi tuần 1 lần. Trong các tác vụ liên quan, các thành viên liên lạc trực tiếp với nhau để cộng tác giải quyết vấn đề.

3 Khám phá dữ liệu

3.1 Tổng quan về dữ liệu

3.1.1 Dữ liệu trong `train.csv`

- **Kích thước dữ liệu:**
 - Số lượng dòng: **1,869** mẫu dữ liệu;
 - Số lượng cột: **15** đặc trưng.
- **Ý nghĩa từng mẫu dữ liệu:** Mỗi dòng là dữ liệu ghi chép nội dung và các thông tin liên quan về Câu hỏi dự đoán (**Diagnostic Questions** hay **DQ**), các câu trả lời và giải thích quan niệm sai lầm cho các phương án lựa chọn sai - nội dung của phương án sai sẽ có yếu tố gây nhiễu.
- **Ý nghĩa các đặc trưng:** Ý nghĩa của từng đặc trưng như sau:
 - **QuestionId (int):** Mã định danh duy nhất cho từng câu hỏi DQ;
 - **ConstructId (int):** Mã định danh duy nhất cho hướng dẫn cấu trúc thực hiện để trả lời câu hỏi DQ;
 - **ConstructName (str):** Nội dung hướng dẫn cấu trúc thực hiện để trả lời câu hỏi DQ;
 - **SubjectId (int):** Mã định danh duy nhất cho môn học liên quan đến câu hỏi;
 - **SubjectName (str):** Tên môn học liên quan đến câu hỏi;
 - **CorrectAnswer (str):** Câu trả lời đúng (Chỉ chọn 1 trong: **A, B, C** hoặc **D**);
 - **QuestionText (str):** Nội dung câu hỏi;
 - **Answer[A/B/C/D]Text (str):** Nội dung câu trả lời của phương án **A, B, C** và **D**;
 - **Misconception[A/B/C/D]Id (int):** Mã định danh duy nhất cho quan niệm sai lầm. Nhân quan niệm sai lầm được gắn sẵn trong `train.csv`; nhiệm vụ của người tham gia là dự đoán các nhân này cho `test.csv`.

3.1.2 Dữ liệu trong test.csv

- **Kích thước dữ liệu:**
 - Số lượng dòng: **3** mẫu dữ liệu;
 - Số lượng cột: **11** đặc trưng.
- **Ý nghĩa từng mẫu dữ liệu:** Mỗi dòng là dữ liệu ghi chép nội dung và các thông tin liên quan về DQ, nội dung của các phương án trả lời để dự đoán nhận quan niệm sai lầm **misconception** cho các phương án lựa chọn sai;
- **Ý nghĩa các đặc trưng:** Ý nghĩa của từng đặc trưng như sau:
 - **QuestionId (int):** Mã định danh duy nhất cho từng câu hỏi DQ;
 - **ConstructId (int):** Mã định danh duy nhất cho hướng dẫn cấu trúc thực hiện để trả lời câu hỏi DQ;
 - **ConstructName (str):** Nội dung hướng dẫn cấu trúc thực hiện để trả lời câu hỏi DQ;
 - **SubjectId (int):** Mã định danh duy nhất cho môn học liên quan đến câu hỏi;
 - **SubjectName (str):** Tên môn học liên quan đến câu hỏi;
 - **CorrectAnswer (str):** Câu trả lời đúng (Chỉ chọn 1 trong: A, B, C hoặc D);
 - **QuestionText (str):** Nội dung câu hỏi;
 - **Answer[A/B/C/D]Text (str):** Nội dung câu trả lời của phương án A, B, C và D;

3.1.3 Dữ liệu trong misconception.csv

- **Kích thước dữ liệu:**
 - Số lượng dòng: **2,587** mẫu dữ liệu;
 - Số lượng cột: **2** đặc trưng.
- **Ý nghĩa từng mẫu dữ liệu:** Mỗi dòng là dữ liệu ghi chép nội dung các loại quan niệm sai lầm **misconception**.
- **Ý nghĩa các đặc trưng:** Ý nghĩa của từng đặc trưng như sau:

- `MisconceptionId` (int): Mã định danh duy nhất cho từng loại quan niệm sai lầm `misconception`;
- `MisconceptionName` (str): Nội dung quan niệm sai lầm `misconception`; dùng để diễn giải chi tiết nội dung quan niệm sai lầm của `Misconception[A/B/C/D]` Id trong tập dữ liệu `train.csv`.

3.1.4 Dữ liệu trong `sample_submission.csv`

Đây là mẫu ví dụ biểu diễn tập dữ liệu đầu ra sao cho đúng định dạng, cụ thể ý nghĩa mỗi cột:

- `QuestionId_Answer`: Mỗi câu hỏi có ba câu trả lời không chính xác mà bạn cần dự đoán, có format là `[QuestionId]_[A/B/C/D]`;
- `MisconceptionId`: Có thể dự đoán tối đa **25** giá trị, phân cách bằng dấu cách.

3.2 Khám phá dữ liệu dựa trên quan sát và thống kê

3.2.1 Dữ liệu trong `train.csv`

Trước hết, quan sát những thống kê số liệu cơ bản theo từng cột từ dữ liệu:

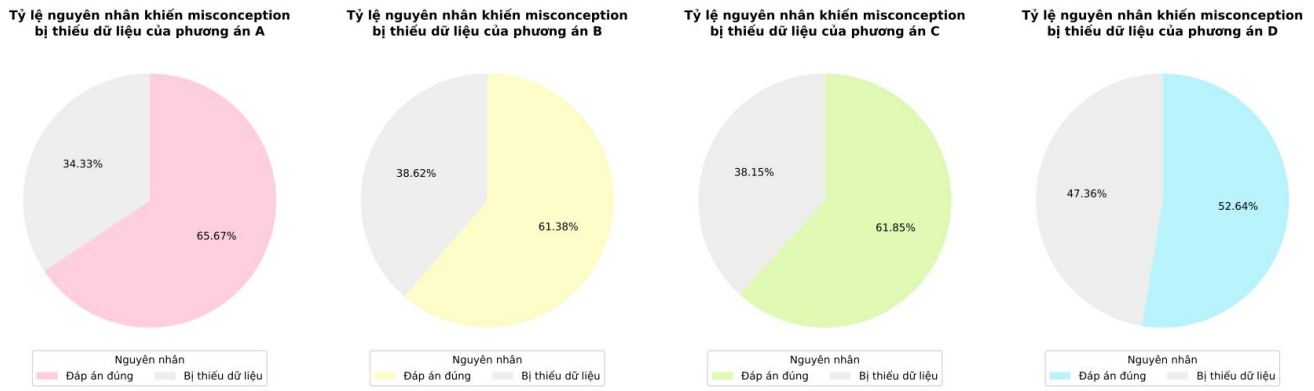
	Count	Unique	Top	Freq
ConstructName	1869	757	Calculate the square of a number	14
SubjectName	1869	163	Linear Equations	53
CorrectAnswer	1869	4	C	488
QuestionText	1869	1857	Which of the following pairs of function...	4
AnswerAText	1869	1219	Only Tom	93
AnswerBText	1869	1230	Only Katie	109
AnswerCText	1869	1222	Both Tom and Katie	158
AnswerDText	1869	1184	Neither is correct	187

	Count	Type	Mean	Std	Min	Median	Max
QuestionId	1869	int	934.00	539.68	0	934.00	1868
ConstructId	1869	int	1613.26	1060.59	4	1470.00	3526
SubjectId	1869	int	225.37	238.54	33	203.00	1984
MisconceptionAId	1135	float	1308.60	744.52	1	1336.00	2585
MisconceptionBId	1118	float	1308.03	766.49	1	1379.00	2586
MisconceptionCId	1080	float	1285.30	742.21	2	1294.50	2585
MisconceptionDId	1037	float	1264.57	759.82	0	1282.00	2583

3.2.1.1 Kiểu dữ liệu và dữ liệu bị thiếu

Trong bảng thống kê số liệu cơ bản cho các cột, ta thấy rằng `Misconception[A/B/C/D]Id` có số lượng mẫu dữ liệu không bằng với số lượng mẫu dữ liệu của các cột khác. Điều này cho thấy rằng có khả năng dữ liệu bị thiếu ở các cột này, khiến chúng có kiểu dữ liệu là `float` thay vì kiểu dữ liệu là `int`.

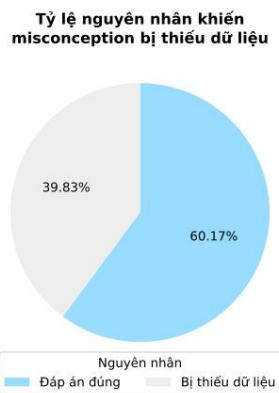
Hầu hết các phương án trả lời bị thiếu dữ liệu bởi:



Hình 1: Biểu đồ tròn biểu diễn tỷ lệ nguyên nhân misconceptions bị thiếu theo mỗi câu trả lời

- Từ 50% đến 70% dữ liệu là phương án trả lời thuộc đáp án đúng của câu hỏi, do đó không thể có misconception;
- Từ 30% đến 50% dữ liệu thực sự bị khuyết dự đoán misconception, đây là một tỷ lệ đáng quan ngại ảnh hưởng đến hiệu suất dự đoán của mô hình sau này.

Quan sát trên tổng thể toàn bộ phương án trả lời, ta thấy:

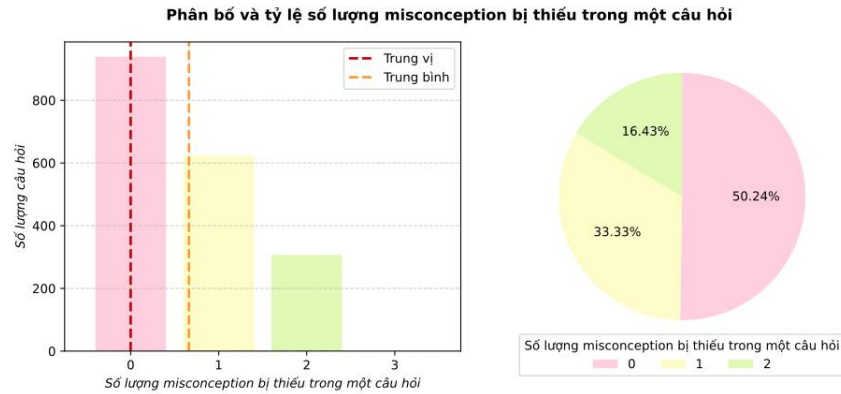


Hình 2: Biểu đồ tròn biểu diễn tỷ lệ nguyên nhân misconceptions bị thiếu

- Đa số misconception bị thiếu là do phương án đúng, không cần giải thích về quan niệm sai lầm, chiếm khoảng gần $\frac{2}{3}$ trong tổng số misconception bị thiếu;
- Tuy nhiên, còn tồn tại hơn $\frac{1}{3}$ trong tổng số misconception thực sự do bị khuyết dữ liệu, nó ảnh hưởng đến hiệu suất huấn luyện mô hình. Nếu không có phương pháp đề xuất làm giàu dữ liệu phù hợp, điểm đánh giá mô hình khó để vượt qua 0.6.

Và từ phần này trở đi, ta coi các dữ liệu bị thiếu của `Misconception[A/B/C/D]` Id không bao gồm lý do phương án trả lời đó là đáp án đúng.

Quan sát theo từng câu hỏi:



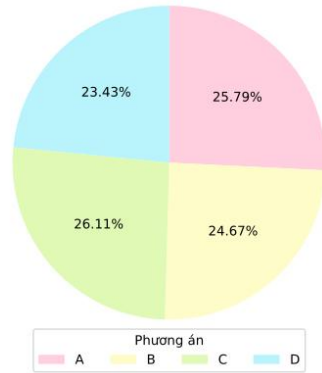
Hình 3: Các biểu đồ thống kê misconceptions bị thiếu theo câu hỏi

- Số lượng đáp án sai bị thiếu `misconception` trong một câu hỏi có giá trị tối đa là 2, nghĩa là không tồn tại bất kỳ câu hỏi nào bị thiếu toàn bộ `misconception`;
- Tỷ lệ câu hỏi có đầy đủ `misconception` chiếm khoảng 50% số lượng câu hỏi, tỷ lệ còn lại lần lượt mô tả cho số lượng câu hỏi mà trong đó có ít nhất 1 `misconception`. Đồng thời, giá trị trung vị cũng có giá trị là 0, cho thấy đa số câu hỏi mà phương án trả lời sai của nó vẫn được giải thích quan niệm sai lầm;
- Việc xem xét xóa bỏ giá trị thiếu theo đơn vị câu hỏi gây ra hao hụt dữ liệu cao hơn việc loại bỏ giá trị thiếu theo đơn vị `MisconceptionId`, bởi vì nếu tồn tại câu hỏi chứa nhiều nhất 2 phương án trả lời sai không được gán `misconception` sẽ vô tình loại bỏ thông tin đã được gán nhãn cho phương án sai còn lại.

3.2.1.2 Khám phá `CorrectAnswer`

Quan sát theo từng câu hỏi:

Tỷ lệ các loại phương án là câu trả lời đúng

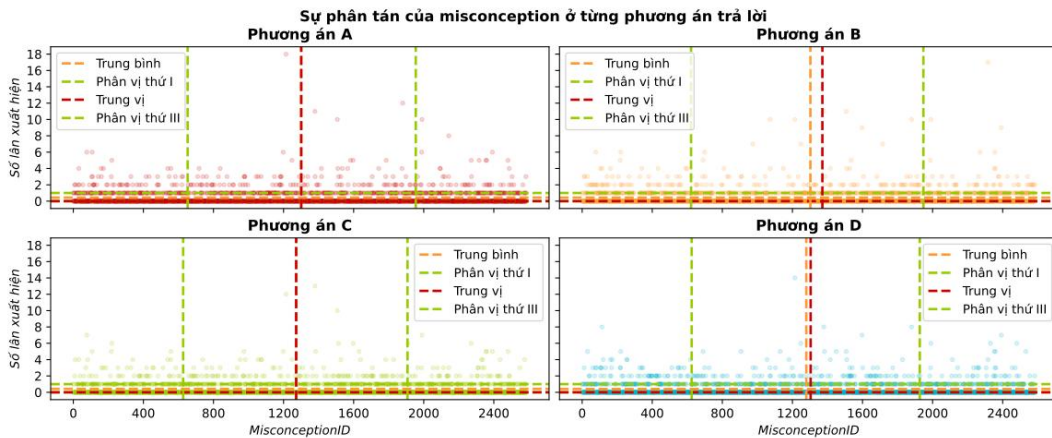


Hình 4: Biểu đồ tròn biểu diễn tỷ lệ phương án lựa chọn là câu trả lời đúng

- Câu trả lời đúng của mỗi câu hỏi phân bố khá đồng đều cho các phương án A, B, C và D;
- Tỷ lệ mỗi câu hỏi có phương án lựa chọn A, B, C và D là đáp án đúng xấp xỉ $\frac{1}{4}$;
- Giá trị tỷ lệ của mỗi phương án lựa chọn nằm trong $[23, 27] \%$.

3.2.1.3 Khám phá Misconception[A/B/C/D] Id

Quan sát theo từng câu hỏi:



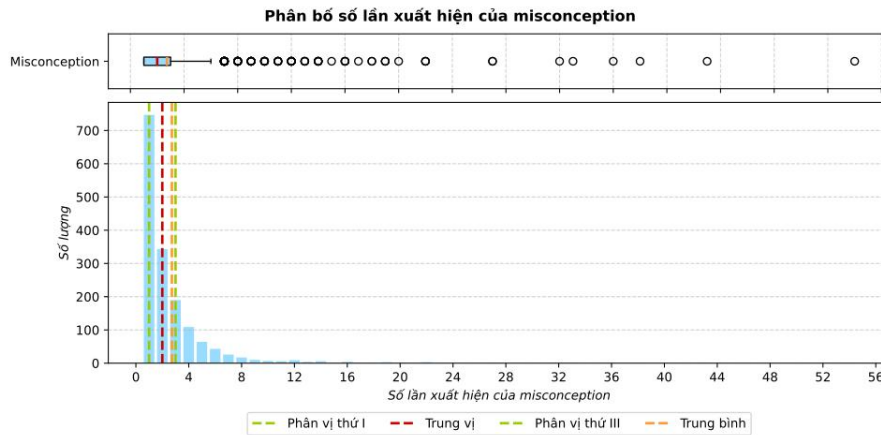
Hình 5: Biểu đồ phân tán biểu diễn sự phân tán của các loại misconception ở các phương án trả lời sai

- Giữa các phương án trả lời, phân bố misconception ở hai phương diện loại misconception và số lần xuất hiện của misconception hầu như giống nhau;
- Ở mỗi phương án, hầu hết số lần xuất hiện của misconception là 0 hoặc 1 lần, lý do gây ra điều này là bởi số lượng loại misconception được sử dụng ít hơn so với số lượng mẫu dữ liệu,

cộng hưởng với việc tồn tại những loại **misconception** có số lần xuất hiện lớn hơn nhiều so với trung vị;

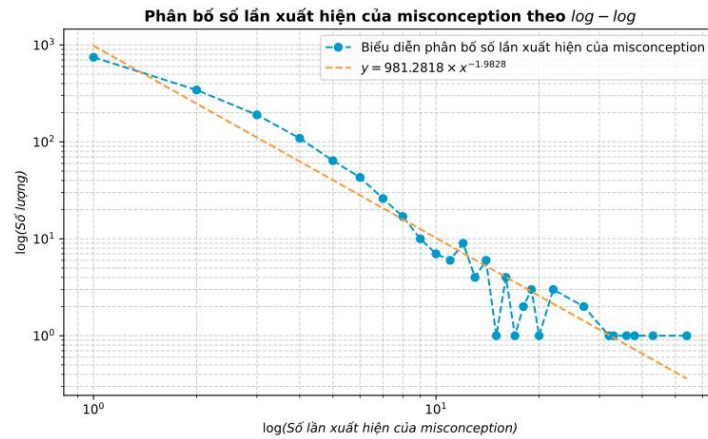
- Vì vậy, có thể tổng hợp **Misconception[A/B/C/D]Id** thành **MisconceptionId** duy nhất sẽ giúp cải thiện việc xử lý và trực quan đồ vật vĩa mà không ảnh hưởng mất mát thông tin.

Sau khi tổng hợp **Misconception[A/B/C/D]Id** thành **MisconceptionId**, ta có thể thấy:



Hình 6: Các biểu đồ biểu diễn sự phân bố số lần xuất hiện của **misconception**

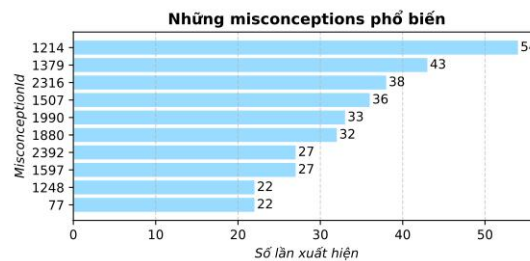
- Chỉ có 1,604 loại **misconception** trong tổng số 2,587 loại là xuất hiện trong dữ liệu huấn luyện, chiếm tỷ lệ khoảng 62%. Cho thấy rằng các loại **misconception** trong dữ liệu huấn luyện chưa thực sự phong phú để mô hình xây dựng đạt được hiệu suất cao;
- Phân bố số lần xuất hiện của **misconception** hầu hết có giá trị trong $[1, 3]$, tồn tại lượng khá nhỏ **misconception** có số lần xuất hiện lớn hơn 7 có tần suất thấp hơn đáng kể khiến không thể quan sát rõ bằng mắt thường trong biểu đồ histogram;
- Phân bố có thiên hướng lệch dương mạnh;
- Phân bố số lần xuất hiện của **misconception** và mối quan hệ giữa chúng có xu hướng tuân theo phân phối bậc tuân theo quy luật lũy thừa (power-law).



Hình 7: Biểu đồ biểu diễn sự phân bố số lần xuất hiện của misconception theo $\log - \log$

Kết quả kiểm định: (trực quan tại hình 7)

- **Lũy thừa của luật mũ:** 1.9828338233337477;
- **Hệ số góc:** 6.888859651005454;
- **R-squared:** 0.9187549215720946;
- **P-value:** 1.0831253499517404e-15;
- Kết quả cho thấy phân bố tuân theo phân phối bậc tuân theo quy luật lũy thừa của luật mũ xấp xỉ 1.98 với mức độ phù hợp 91.88% có độ tin cậy 99.99%.
- Kết quả này cho ta thấy một cách giải thích khoa học cho việc hầu hết số lần xuất hiện của misconception là 2 lần nhưng lại tồn tại một số misconception có số lần xuất hiện lớn hơn rất nhiều.
- Những loại misconception phổ biến nhất:

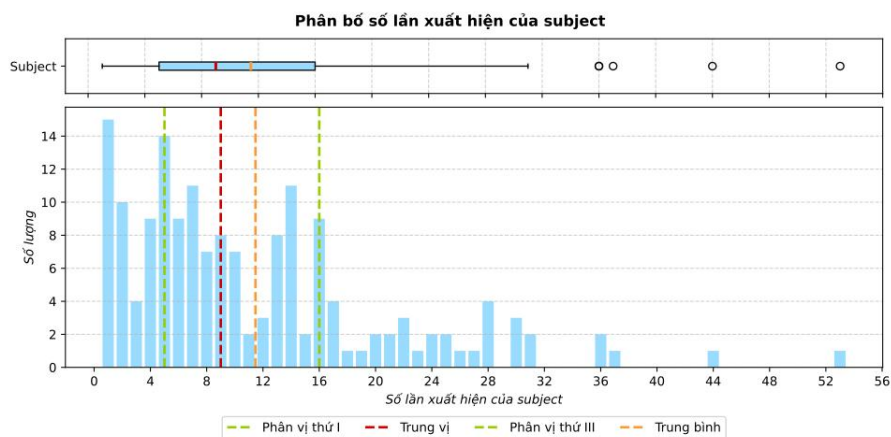


Hình 8: Biểu đồ thống kê misconception phổ biến

1. When solving an equation, uses the same operation rather than the inverse:
Khi giải phương trình, sử dụng cùng một phép toán thay vì phép toán nghịch đảo;
2. Rounds down instead of up: Làm tròn xuống thay vì làm tròn lên;
3. Mixes up squaring and multiplying by 2 or doubling: Nhầm lẫn giữa bình phương và nhân với 2 hoặc gấp đôi;
4. Carries out operations from left to right regardless of priority order: Thực hiện phép toán từ trái sang phải bất kể thứ tự ưu tiên;
5. Fails to reflect across mirror line: Không phản chiếu qua trục đối xứng;
6. Mixes up greater than and less than symbols: Nhầm lẫn giữa ký hiệu lớn hơn và nhỏ hơn;
7. Rounds to the wrong degree of accuracy (rounds too much): Làm tròn sai mức độ chính xác (làm tròn quá nhiều);
8. Believes multiplying two negatives gives a negative answer: Tin rằng nhân hai số âm sẽ cho kết quả âm;
9. Rounds to the wrong degree of accuracy (rounds too little): Làm tròn sai mức độ chính xác (làm tròn quá ít);
10. Does not follow the arrows through a function machine, changes the order of the operations asked: Không tuân theo hướng dẫn trong sơ đồ hàm, thay đổi thứ tự các phép toán yêu cầu.

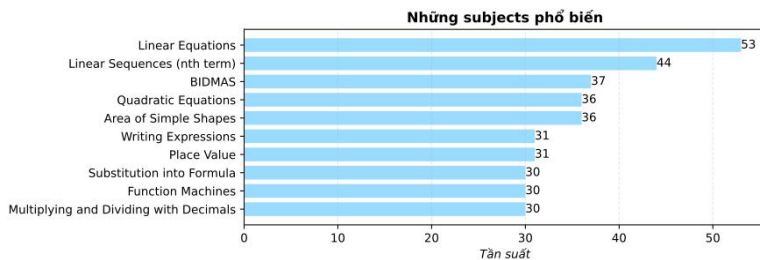
3.2.1.4 Khám phá Subject

Quan sát phân bố số lần xuất hiện của **subject**:



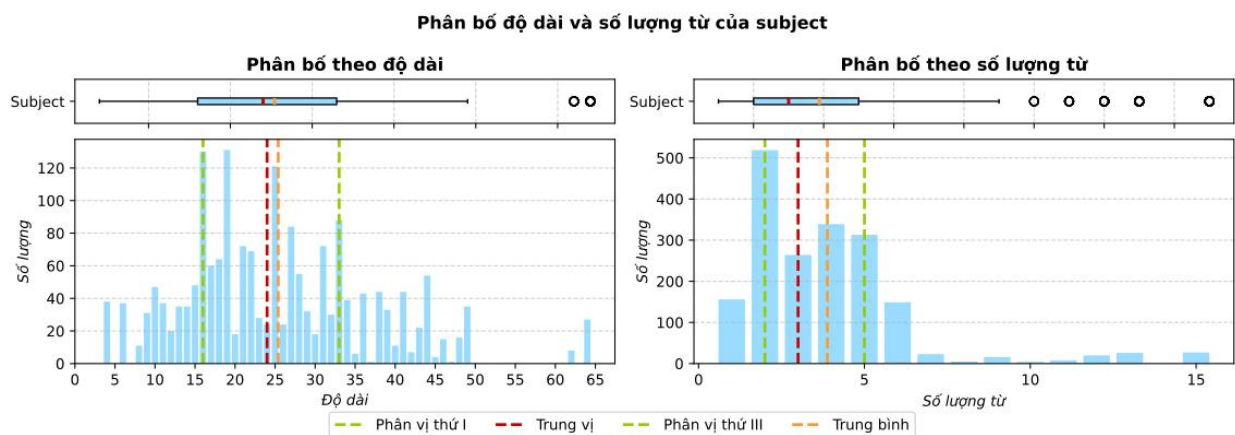
Hình 9: Các biểu đồ biểu diễn sự phân bố số lần xuất hiện của **subject**

- Số lần xuất hiện **subject** hầu hết tập trung trong $[1, 31]$, với giá trị trung vị là 9 và trung bình khoảng 11.5;
- Phân bố hầu như không có quy luật cụ thể, mang yếu tố ngẫu nhiên, và có thiên hướng lệch dương;
- Những **subject** phổ biến nhất:



Hình 10: Biểu đồ thống kê **subject** phổ biến

Quan sát phân bố kích thước của **subject**:



Hình 11: Các biểu đồ biểu diễn sự phân bố kích thước của **subject** khi chưa xử lý nội dung văn bản

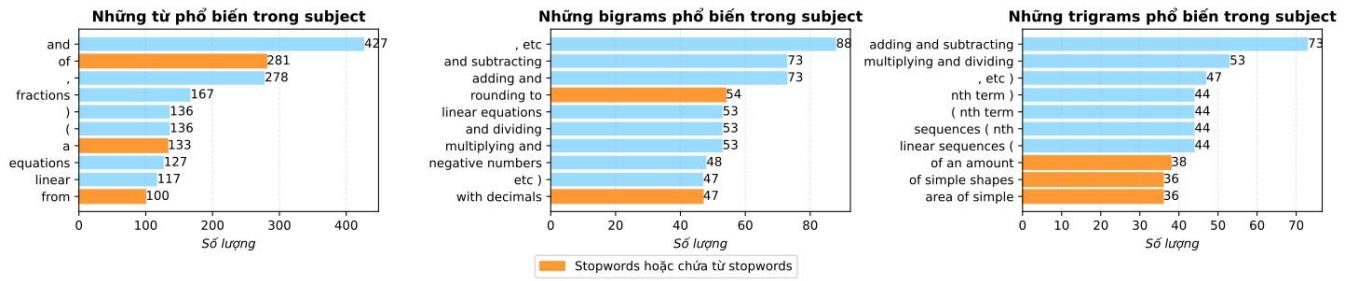
– Phân bố độ dài:

- * Độ dài của **subject** chủ yếu tập trung trong khoảng từ 15 đến 33 ký tự, với trung vị 24;
- * Vẫn có một số **subject** có độ dài xuất hiện ở các giá trị lớn hơn 50 ký tự, như được minh họa qua các điểm ngoài trong biểu đồ boxplot.

– Phân bố số lượng từ: Nhóm sử dụng hàm `word_tokenize` của thư viện `nltk` để đếm số lượng từ trong **subject**:

- * Số lượng từ **subject** tập trung nhiều ở khoảng từ 2 đến 5, với với trung vị 3;
- * Vẫn có một số **subject** có số lượng từ xuất hiện ở các giá trị lớn hơn 10, như được minh họa qua các điểm ngoài trong biểu đồ boxplot;
- * Giá trị trung bình cao hơn trung vị, cho thấy dữ liệu có thể bị kéo dài ở phần đuôi phải, làm cho đồ thị có phần đuôi phải dày.

– Những từ và n-grams phổ biến: Nhóm sử dụng hàm `word_tokenize` của thư viện `nltk` để đếm số lượng từ trong **subject**: Ta thấy những từ và n-grams phổ biến nhất trong **subject** xuất hiện một số từ ít cung cấp nhiều ý nghĩa stopwords.



Hình 12: Các biểu đồ thống kê các từ và n-grams phổ biến của subject khi chưa xử lý nội dung văn bản

Nhóm thực hiện một số xử lý nội dung văn bản của subject để khám phá cốt lõi hơn. Cụ thể, nhóm thực hiện xử lý nội dung bằng thuật toán 1:

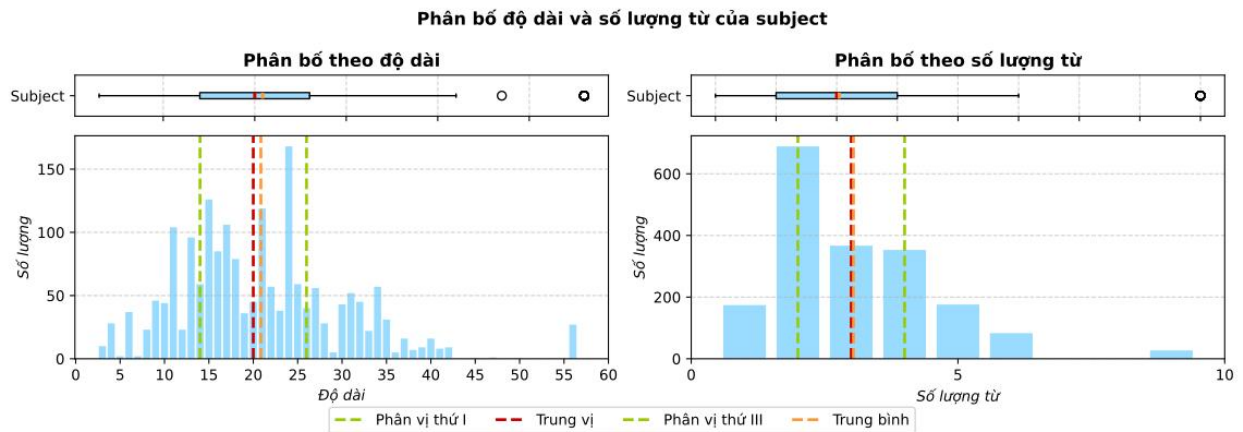
Algorithm 1 Tiền xử lý văn bản cho Subject và Construct

Input Văn bản chưa được xử lý

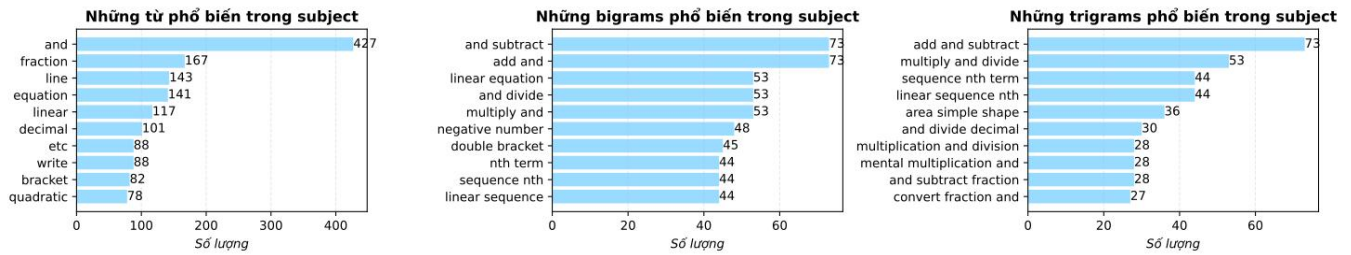
Output Văn bản đã được xử lý

- 1: Chuyển tất cả các ký tự thành chữ thường;
 - 2: Thay thế chuỗi n't bằng not;
 - 3: Loại bỏ tất cả dấu câu, ngoại trừ dấu câu kèm theo số không cần loại bỏ;
 - 4: Chuyển từ về dạng gốc bằng Lemmatizer;
 - 5: Loại bỏ các từ ít mang ý nghĩa trong câu (stopwords);
 - 6: Loại bỏ các khoảng trắng thừa;
-

Kết quả sau khi xử lý nội dung văn bản của subject: (trực quan tại hình 13 và 14)



Hình 13: Các biểu đồ biểu diễn sự phân bố kích thước của subject đã xử lý nội dung văn bản



Hình 14: Các biểu đồ thống kê các từ và **n-grams** phổ biến của **subject** đã xử lý nội dung văn bản

– Phân bố độ dài:

- * Độ dài của các **subject** giảm từ 24 xuống còn khoảng 19 ký tự, tập trung trong [14, 26] ký tự. Việc này làm văn bản gọn hơn, giảm nhiễu và chuẩn hóa tốt hơn;
- * Sự giảm độ dài giúp mô hình xử lý nhanh hơn và cải thiện tính tổng quát khi áp dụng lên dữ liệu mới.

– Phân bố số lượng từ:

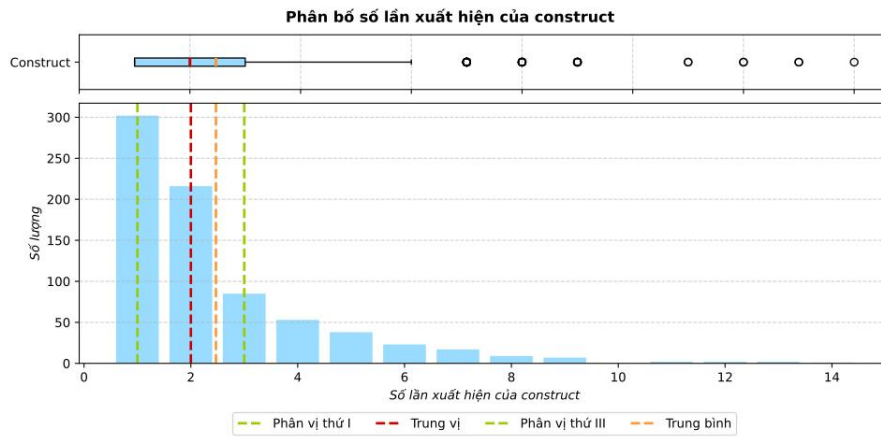
- * Số lượng từ tập trung trong [2, 4] từ. Văn bản tinh gọn, chỉ giữ lại từ mang ý nghĩa cốt lõi;
- * Biểu đồ phân bố đã hạn chế được đuôi phải dày, thu hẹp khoảng cách giá trị trung bình và trung vị;
- * Điều này giúp mô hình tập trung vào đặc trưng quan trọng, giảm nhiễu và tối ưu tài nguyên huấn luyện.

– Những từ và **n-grams** phổ biến:

- * Các từ và **n-grams** phổ biến sau xử lý có tính đồng nhất cao, tập trung vào từ khóa quan trọng nhờ loại bỏ **stopwords** và chuẩn hóa;
- * Sự đồng nhất này cải thiện khai thác đặc trưng ngữ nghĩa, nâng cao hiệu suất mô hình học sâu.

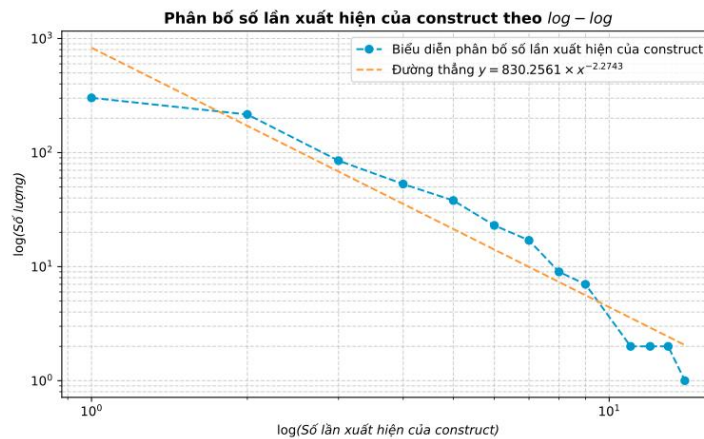
3.2.1.5 Khám phá Construct

Quan sát phân bố số lần xuất hiện của **construct**: (trực quan tại hình 15)



Hình 15: Các biểu đồ biểu diễn sự phân bố số lần xuất hiện của **construct**

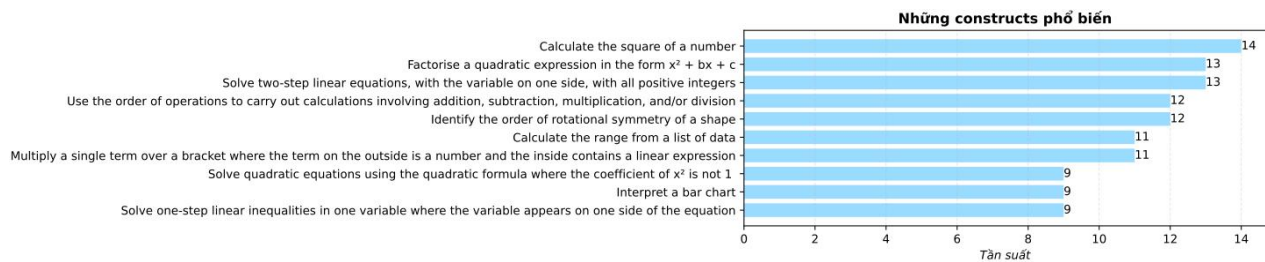
- Số lần xuất hiện của hầu hết các loại **construct** tập trung trong $[1, 3]$, với giá trị trung vị là 2 và trung bình khoảng 2.5. Bên cạnh đó, một số **construct** mà số lần xuất hiện lớn hơn 6 có tần suất thấp hơn đáng kể;
- Phân bố có thiên hướng lệch dương mạnh;
- Phân bố số lần xuất hiện của **construct** và mối quan hệ giữa chúng có xu hướng tuân theo phân phối bậc tuân theo quy luật lũy thừa (power-law). Kết quả kiểm định: (trực quan tại hình 16)



Hình 16: Biểu đồ biểu diễn sự phân bố số lần xuất hiện của **construct** theo $\log - \log$

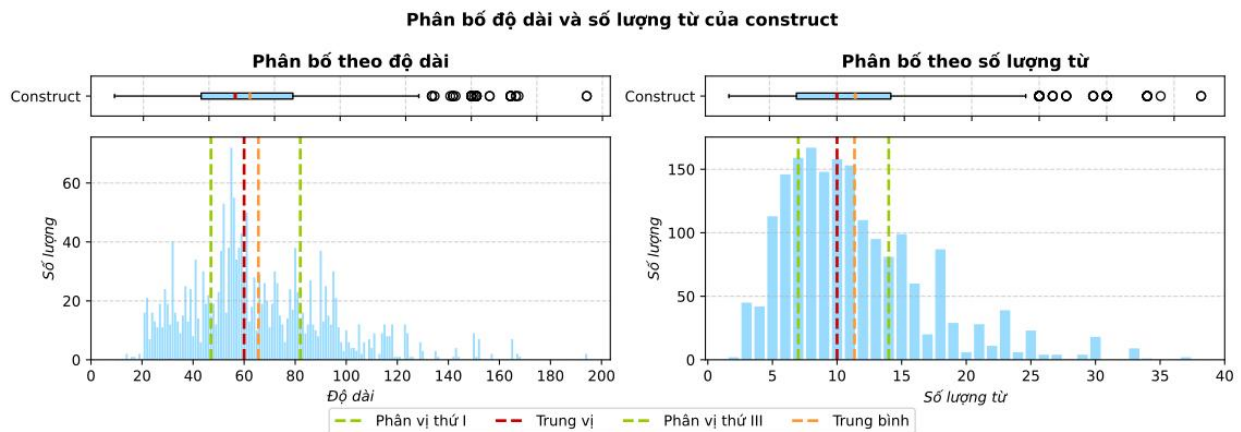
- * **Lũy thừa của luật mũ:** 2.274273829477023;
- * **Hệ số góc:** 6.721734202631855;
- * **R-squared:** 0.9225515372036349;

- * **P-value:** 1.8866639899702487e-07;
 - * Kết quả cho thấy phân bố tuân theo phân phối bậc tuân theo quy luật lũy thừa của luật mũ xấp xỉ 2.27 với mức độ phù hợp 92.25% có độ tin cậy 99.99%;
 - * Kết quả này cho ta thấy một cách giải thích khoa học cho việc hầu hết số lần xuất hiện của **construct** là 2 lần nhưng lại tồn tại một số **construct** có số lần xuất hiện lớn hơn nhiều.
- Những **construct** phổ biến nhất:



Hình 17: Biểu đồ thống kê **construct** phổ biến

Quan sát phân bố kích thước của **construct**:

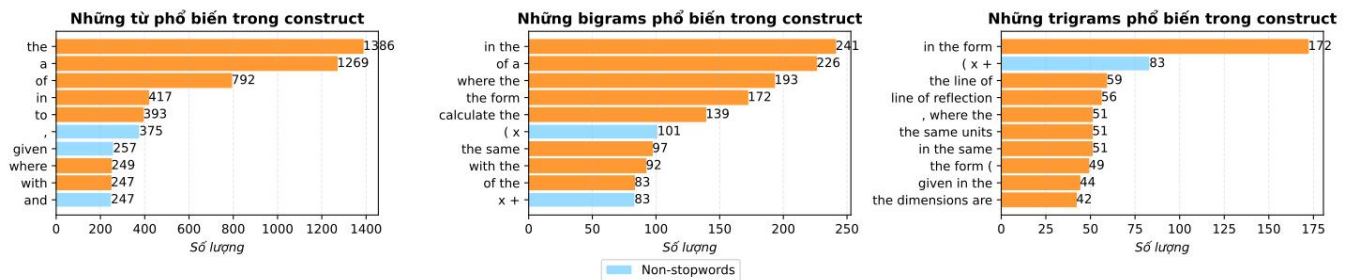


Hình 18: Các biểu đồ biểu diễn sự phân bố kích thước của **construct** khi chưa xử lý nội dung văn bản

– **Phân bố độ dài:**

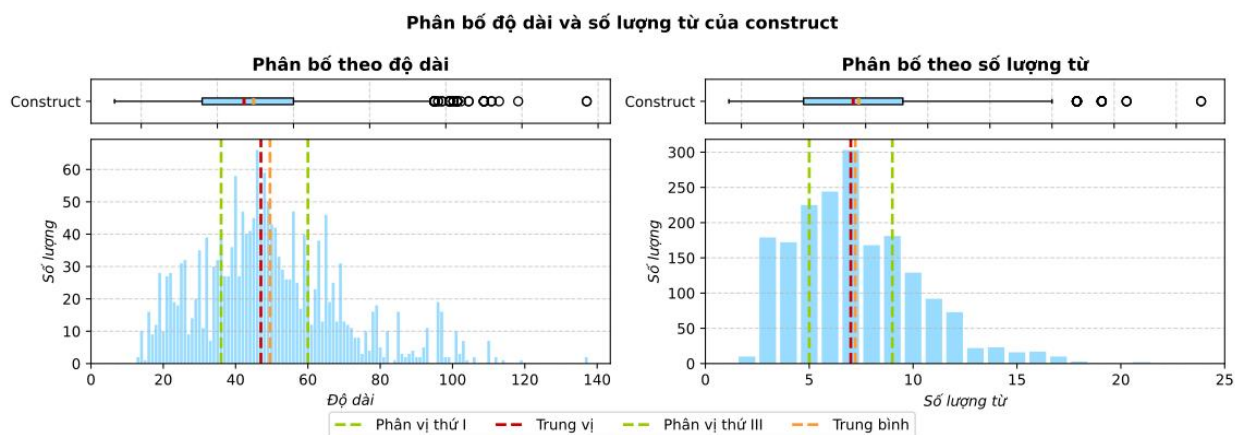
- * Độ dài của **construct** chủ yếu tập trung trong khoảng từ 45 đến 85 ký tự, với trung vị 60;
- * Vẫn có một số **construct** có độ dài xuất hiện ở các giá trị lớn hơn 120 ký tự, như được minh họa qua các điểm ngoài trong biểu đồ boxplot;

- * Giá trị trung bình cao hơn trung vị, cho thấy dữ liệu bị lệch dương, cộng thêm phân bố dữ liệu bị kéo dài ở phần đuôi phải, làm cho đồ thị có phần đuôi phải dày.
- **Phân bố số lượng từ:** Nhóm sử dụng hàm `word_tokenize` của thư viện `nltk` để đếm số lượng từ trong `construct`:
 - * Số lượng từ `construct` tập trung nhiều ở khoảng từ 7 đến 14, với trung vị 10;
 - * Vẫn có một số `construct` có số lượng từ xuất hiện ở các giá trị lớn hơn 24, như được minh họa qua các điểm ngoài trong biểu đồ `boxplot`;
 - * Giá trị trung bình cao hơn trung vị, cho thấy dữ liệu bị lệch dương, cộng thêm phân bố dữ liệu bị kéo dài ở phần đuôi phải, làm cho đồ thị có phần đuôi phải dày.
- **Những từ và n-grams phổ biến:** Ta thấy những từ và n-grams phổ biến nhất trong `construct` hầu hết là những từ ít cung cấp nhiều ý nghĩa **stopwords**.

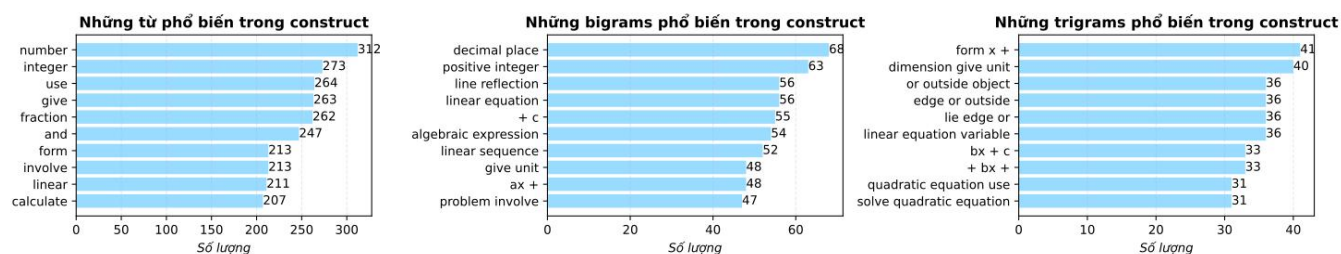


Hình 19: Các biểu đồ thống kê các từ và n-grams phổ biến của `construct` khi chưa xử lý nội dung văn bản

Nhóm thực hiện một số xử lý nội dung văn bản của `construct` để khám phá cốt lõi hơn. Cụ thể, nhóm thực hiện xử lý nội dung bằng thuật toán 1. Kết quả sau khi xử lý nội dung văn bản của `construct`: (trực quan tại hình 20 và 21)



Hình 20: Các biểu đồ biểu diễn sự phân bố kích thước của **construct** đã xử lý nội dung văn bản



Hình 21: Các biểu đồ thống kê các từ và **n-grams** phổ biến của **construct** đã xử lý nội dung văn bản

– Phân bố độ dài:

- * Độ dài của các **construct** giảm đáng kể, từ trung vị 60 ký tự xuống còn khoảng 48 ký tự, tập trung trong [35, 70] ký tự. Điều này cho thấy hiệu quả của các bước tiền xử lý trong việc loại bỏ nhiễu và chuẩn hóa văn bản;
- * Sự giảm này giúp đơn giản hóa dữ liệu đầu vào, giảm độ phức tạp tính toán cho mô hình và cải thiện khả năng học tập trên các đặc trưng quan trọng hơn.

– Phân bố số lượng từ:

- * Số lượng từ trong **construct** giảm từ trung vị 10 xuống khoảng 7, tập trung chủ yếu trong [5, 9] từ. Điều này phản ánh việc tinh gọn văn bản, tập trung vào các thông tin cốt lõi và loại bỏ từ ít ý nghĩa;
- * Phân bố số lượng từ sau xử lý cũng thu hẹp khoảng cách giữa giá trị trung bình và trung vị, giảm thiểu đuôi phải dày, giúp mô hình nhận diện đặc trưng ngữ nghĩa dễ dàng hơn, đồng thời tối ưu tài nguyên khi huấn luyện.

– **Những từ và n-grams phổ biến:**

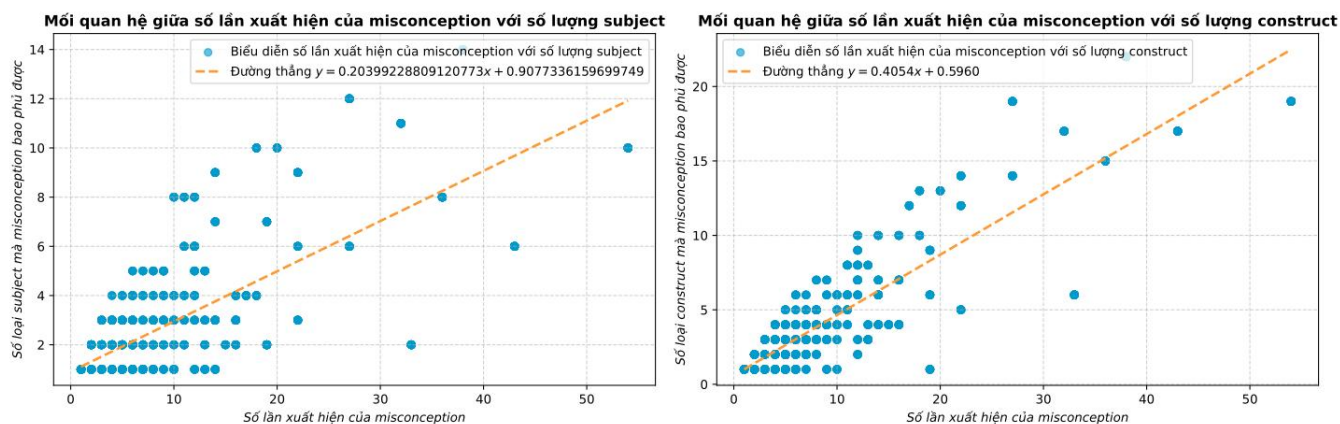
- * Các từ và n-grams phổ biến trở nên đồng nhất và tập trung hơn vào từ khóa sau khi loại bỏ stopwords và thực hiện lemmatization. Điều này làm nổi bật các yếu tố quan trọng trong construct;
- * Việc đồng nhất hóa này không chỉ cải thiện chất lượng đặc trưng mà còn giúp mô hình dễ dàng khai thác các quan hệ ngữ nghĩa, từ đó nâng cao hiệu suất dự đoán.

3.2.1.6 Khám phá mối quan hệ giữa Subject, Construct và Misconception

Quan sát từ dữ liệu, ta thấy dường như Misconception xuất hiện càng nhiều thì bao phủ được nhiều loại Subject và Construct. Cụ thể:

- Đối với mối quan hệ giữa số lần xuất hiện của Misconception với số lượng Subject mà nó bao phủ, gọi giả thiết:
 - * H_0 : Số lần xuất hiện của Misconception ảnh hưởng đến số lượng Subject mà nó bao phủ;
 - * H_1 : Bác bỏ H_0 , không đủ cơ sở để kết luận số lần xuất hiện của Misconception ảnh hưởng đến số lượng Subject mà nó bao phủ.
- Đối với mối quan hệ giữa số lần xuất hiện của Misconception với số lượng Construct mà nó bao phủ, gọi giả thiết:
 - * H_0 : Số lần xuất hiện của Misconception ảnh hưởng đến số lượng Construct mà nó bao phủ;
 - * H_1 : Bác bỏ H_0 , không đủ cơ sở để kết luận số lần xuất hiện của Misconception ảnh hưởng đến số lượng Construct mà nó bao phủ.

Kết quả trực quan (xem tại hình 22) và kiểm định như sau:



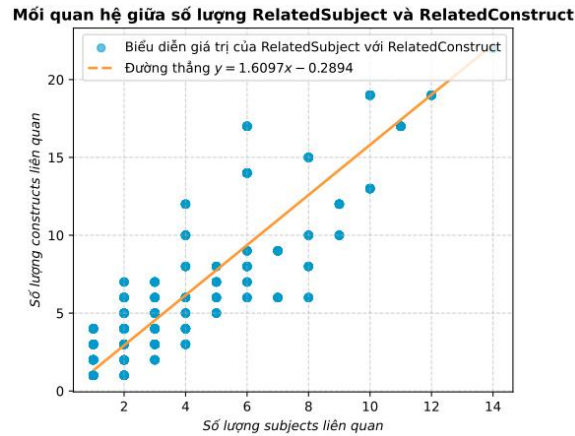
Hình 22: Các biểu đồ biểu diễn mối quan hệ số lần xuất hiện của Misconception với số lượng Subject và Construct mà nó bao phủ

- Mối quan hệ giữa số lần xuất hiện của Misconception với số lượng Subject mà nó bao phủ:
 - * Hiệp phương sai: 0.20399228809120773
 - * R-squared: 0.6314807750677467
 - * P-value: 0.0
- Mối quan hệ giữa số lần xuất hiện của Misconception với số lượng Construct mà nó bao phủ:
 - * Hiệp phương sai: 0.40538197845240015
 - * R-squared: 0.8248263977399055
 - * P-value: 0.0

Kết quả trên là cơ sở cho thấy **misconception** xuất hiện càng nhiều thì càng bao phủ được nhiều **subject** và **construct**, với mức độ phù hợp cao và độ tin cậy gần như tuyệt đối. Để tăng hiệu suất mô hình dự đoán, ta có thể hỗ trợ gom nhóm những **subject** và **construct** liên quan với nhau trong cùng **misconception**, làm giàu thông tin cho dữ liệu.

Sau khi thực hiện thao tác gom nhóm những **subject** và **construct** liên quan với nhau trong cùng **misconception**, gọi những cột mới này lần lượt là các **subjects** liên quan và các **constructs** của một **misconception**, nhóm nhận thấy kích thước các **subjects** liên quan tăng 1 đơn vị thì kích thước các **constructs** liên quan cũng tăng 1.61 lần. Biểu đồ trong hình

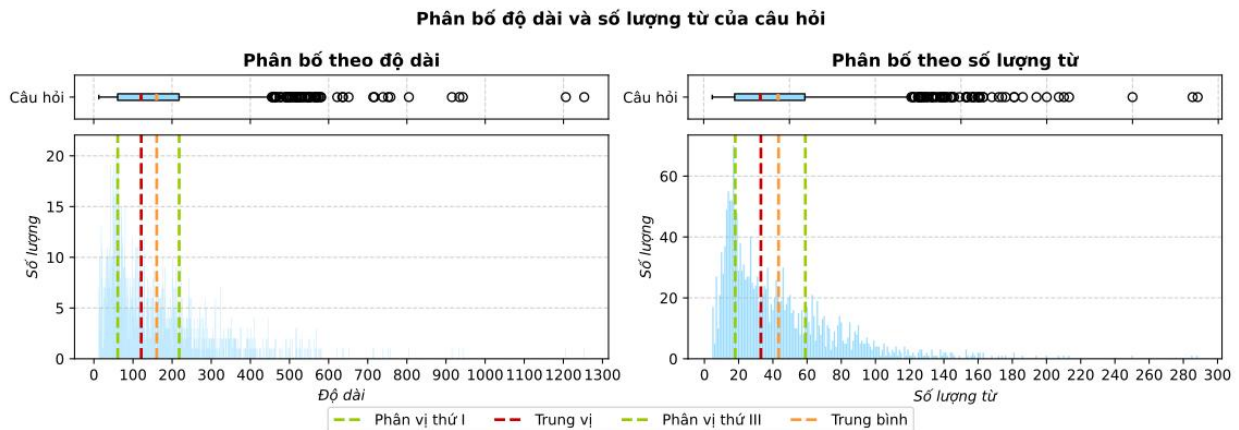
23 đã cho thấy điều đó, có mức độ phù hợp R-squared là 0.857 với độ tin cậy gần như tuyệt đối.



Hình 23: Biểu đồ phân tán biểu diễn mối quan hệ số lượng các subjects liên quan với số lượng các constructs liên quan trong cùng misconception

3.2.1.7 Khám phá QuestionText

Quan sát phân bố kích thước của câu hỏi:



Hình 24: Các biểu đồ biểu diễn sự phân bố kích thước của câu hỏi khi chưa xử lý nội dung văn bản

– Phân bố độ dài:

- * Độ dài của câu hỏi chủ yếu tập trung trong khoảng từ 61 đến 218 ký tự, với trung vị 121;
- * Vẫn có một số câu hỏi có độ dài xuất hiện ở các giá trị lớn hơn 450 ký tự, như được minh họa qua các điểm ngoài trong biểu đồ boxplot;

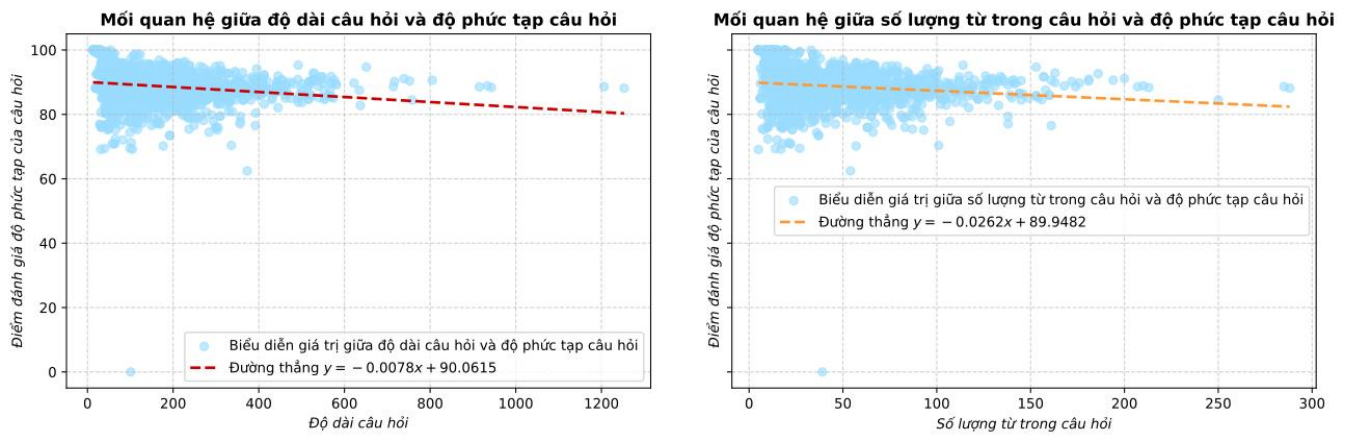
- * Giá trị trung bình cao hơn trung vị, cho thấy dữ liệu bị lệch dương, cộng thêm phân bố dữ liệu bị kéo dài ở phần đuôi phải, làm cho đồ thị có phần đuôi phải dày.
- **Phân bố số lượng từ:** Nhóm sử dụng hàm `word_tokenize` của thư viện `nltk` để đếm số lượng từ trong câu hỏi:
 - * Số lượng từ câu hỏi tập trung nhiều ở khoảng từ 18 đến 59, với trung vị 33;
 - * Vẫn có một số câu hỏi có số lượng từ xuất hiện ở các giá trị lớn hơn 130, như được minh họa qua các điểm ngoài trong biểu đồ `boxplot`;
 - * Giá trị trung bình cao hơn trung vị, cho thấy dữ liệu bị lệch dương, cộng thêm phân bố dữ liệu bị kéo dài ở phần đuôi phải, làm cho đồ thị có phần đuôi phải dày.
- **Điểm đánh giá mức độ phức tạp:** Bởi vì nội dung câu hỏi trong dữ liệu tập trung xoay quanh bài toán toán học nên đôi khi có một số câu hỏi có nội dung phức tạp gây khó hiểu. Độ phức tạp của câu hỏi được đánh giá dựa trên chỉ số **readability** (độ dễ đọc) giúp đánh giá mức độ khó hay dễ hiểu của một đoạn văn bản, thường dựa vào cấu trúc câu, số từ, độ dài từ, và các yếu tố ngôn ngữ khác; bằng cách tính điểm **Flesch Reading Ease Score**. Chỉ số này nằm trong khoảng từ 0 đến 100, trong đó:
 - * 100 – 90: Rất dễ đọc – Thích hợp cho trẻ em lớp 5.
 - * 90 – 80: Dễ đọc – Thích hợp cho học sinh lớp 6-8.
 - * 80 – 70: Khá dễ đọc – Thích hợp cho học sinh lớp 9-10.
 - * 70 – 60: Đọc vừa phải – Thích hợp cho học sinh trung học.
 - * 60 – 50: Hơi khó đọc – Thích hợp cho học sinh đại học.
 - * 50 – 30: Khó đọc – Đòi hỏi trình độ đại học.
 - * 30 – 0: Rất khó đọc – Dành cho các văn bản chuyên ngành phức tạp.

Được tính bằng công thức:

$$206.835 - (1.015 \times ASL) - (84.6 \times ASW)$$

trong đó:

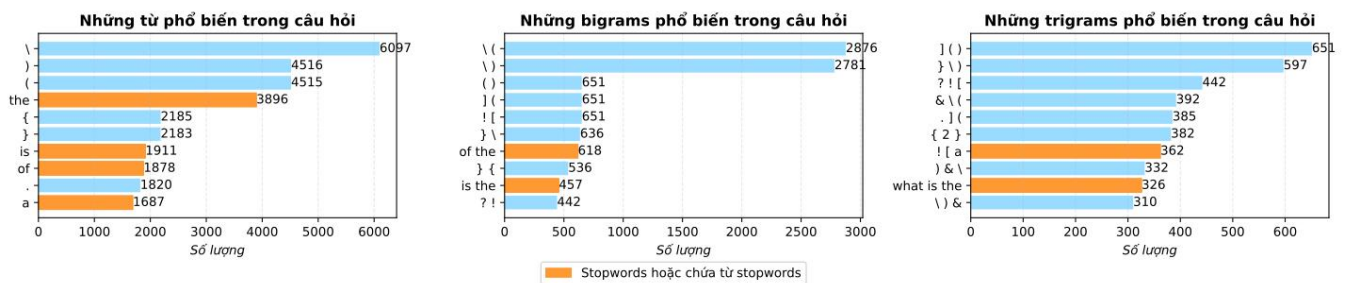
- * ASL: Độ dài câu trung bình (Average Sentence Length).
- * ASW: Độ dài từ trung bình (Average Syllables per Word).



Hình 25: Các biểu đồ biểu diễn mối quan hệ điểm phức tạp của câu hỏi với độ dài và số lượng từ khi chưa tiền xử lý

Kết quả trực quan cho thấy độ dài và số lượng từ của một câu hỏi cũng có thể nói lên được câu hỏi có phức tạp hay không, tuy nhiên kết quả kiểm định mức độ phù hợp khá thấp, có nghĩa rằng người trả lời nên đọc nội dung câu hỏi để đánh giá mức độ phức tạp của câu hỏi so với thang đo của bản thân.

- **Những từ và n-grams phổ biến:** Nhóm sử dụng hàm `word_tokenize` của thư viện `nltk` để đếm số lượng từ trong câu hỏi (trực quan tại hình 26).



Hình 26: Các biểu đồ thống kê các từ và n-grams phổ biến của câu hỏi khi chưa xử lý nội dung văn bản

Đa số các từ và n-grams phổ biến của câu hỏi chưa qua xử lý nội dung văn bản là các cú pháp $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, các từ không cung cấp nhiều ý nghĩa stopwords.

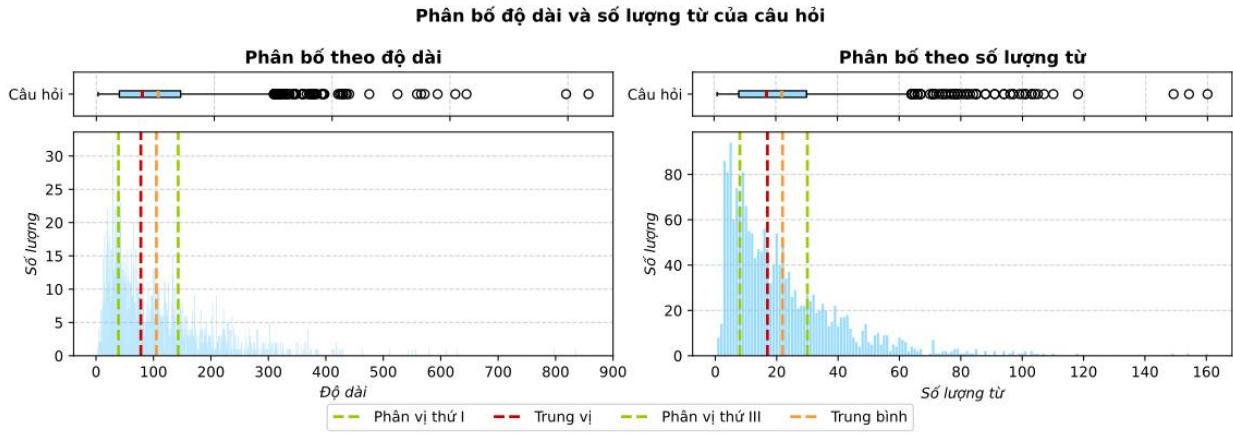
Nhóm thực hiện một số xử lý nội dung văn bản của câu hỏi để khám phá cốt lõi hơn. Cụ thể, nhóm thực hiện xử lý nội dung bằng thuật toán 2. Kết quả sau khi xử lý nội dung văn bản của câu hỏi: (trực quan tại hình 27, 28 và 29)

Algorithm 2 Tiền xử lý văn bản cho Question và Answer

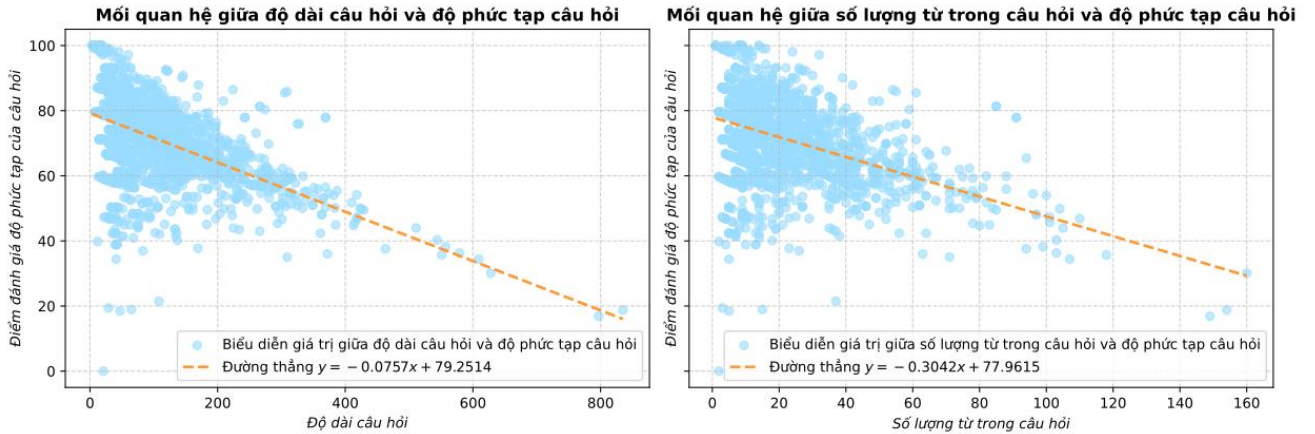
Input Văn bản chưa được xử lý

Output Văn bản đã được xử lý

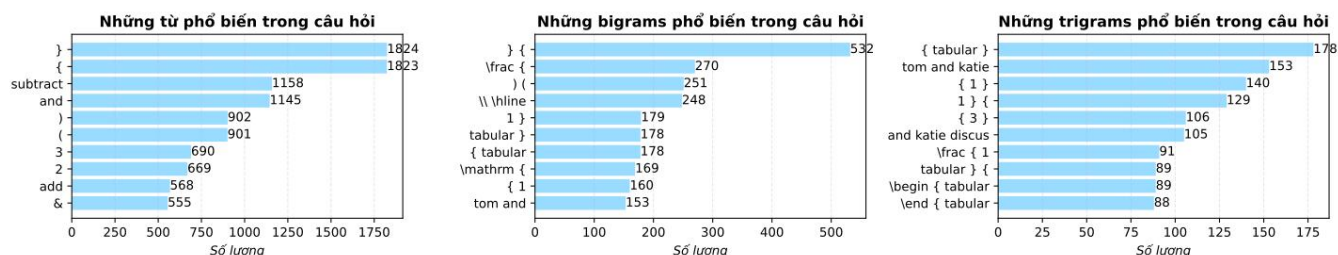
- 1: Chuyển tất cả các ký tự thành chữ thường;
 - 2: Loại bỏ các cú pháp bắt đầu và kết thúc của công thức toán học $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$;
 - 3: Chuyển đổi một số cú pháp $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ thành văn bản thông thường;
 - 4: Loại bỏ khoảng trắng bên cạnh các dấu ngoặc;
 - 5: Chuyển các cú pháp hình ảnh trong markdown thành văn bản thông thường;
 - 6: Chuyển từ về dạng gốc bằng Lemmatizer;
 - 7: Loại bỏ các từ ít mang ý nghĩa trong câu (stopwords);
 - 8: Loại bỏ các khoảng trắng thừa;
-



Hình 27: Các biểu đồ biểu diễn sự phân bố kích thước của câu hỏi đã tiền xử lý nội dung văn bản



Hình 28: Các biểu đồ biểu diễn mối quan hệ điểm phức tạp của câu hỏi với độ dài và số lượng từ đã tiền xử lý



Hình 29: Các biểu đồ thống kê các từ và **n-grams** phổ biến của câu hỏi đã tiền xử lý nội dung văn bản

– Phân bố độ dài:

- * Sau khi tiền xử lý, phân bố độ dài của câu hỏi trở nên gọn gàng hơn, với trung vị giảm từ 121 ký tự xuống còn khoảng 78 ký tự.
- * Phạm vi độ dài tập trung chủ yếu trong khoảng 39 đến 143 ký tự, loại bỏ được nhiều giá trị ngoại lệ vượt quá 450 ký tự. Điều này giúp giảm độ phức tạp trong xử lý ngôn ngữ tự nhiên, tăng tốc độ huấn luyện và giảm nguy cơ gây nhiễu cho mô hình.

– Phân bố số lượng từ:

- * Số lượng từ trong câu hỏi sau tiền xử lý giảm đáng kể, với trung vị giảm từ 33 xuống còn khoảng 17 từ, và phân bố chủ yếu trong khoảng 8 đến 30 từ. Điều này chứng minh rằng việc loại bỏ stopwords và các cú pháp đặc biệt đã giúp câu hỏi trở nên rõ ràng và tập trung vào nội dung chính.
- * Phân bố số lượng từ trở nên đồng nhất hơn, giảm thiểu đuôi phải dày và làm dữ liệu dễ dàng được xử lý bởi các mô hình học sâu. Sự chuẩn hóa này giúp mô hình tập trung vào các từ khóa quan trọng, nâng cao hiệu quả trong việc học các đặc trưng ngữ nghĩa.

– Điểm đánh giá mức độ phức tạp:

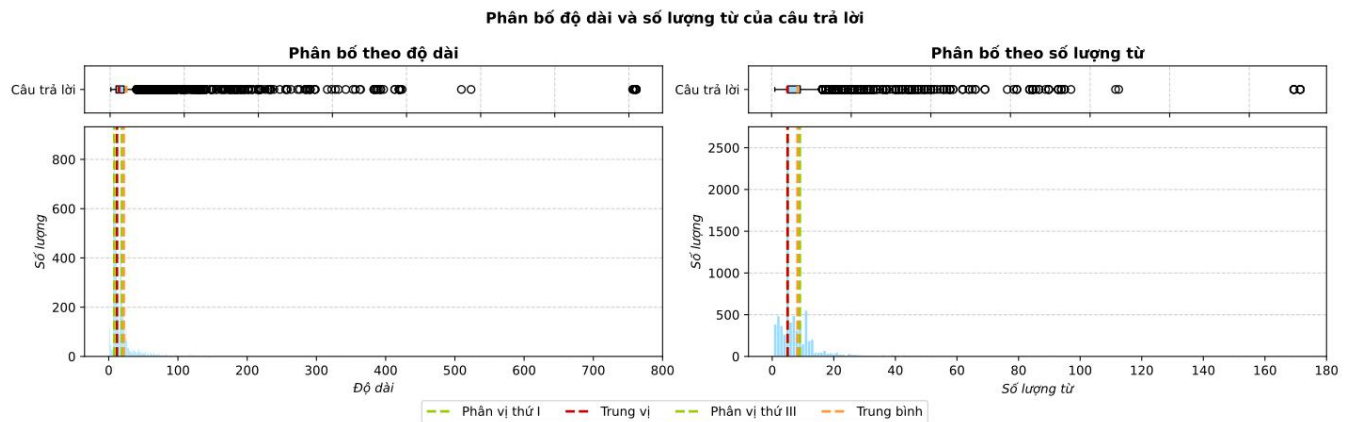
- * Sau tiền xử lý, điểm Flesch Reading Ease Score cho thấy mức sự ảnh hưởng của kích thước câu hỏi ảnh hưởng đến độ phức tạp của nội dung rõ rệt. Điều này cho thấy các câu hỏi đã trở nên đơn giản và dễ hiểu hơn, giúp mô hình xử lý ngôn ngữ tự nhiên dễ dàng nắm bắt ý nghĩa mà không bị nhiễu bởi cú pháp phức tạp.
- * Độ phức tạp giảm còn mang lại lợi ích trong việc cải thiện độ chính xác của các mô hình dự đoán nhờ loại bỏ được các yếu tố gây khó khăn không cần thiết trong văn bản.

– Những từ và n-grams phổ biến:

- * Sau khi tiền xử lý, các từ và n-grams phổ biến trở nên rõ ràng hơn, với sự xuất hiện chủ yếu của các từ khóa liên quan đến nội dung cốt lõi của câu hỏi. Các stopwords và cú pháp đặc biệt không cần thiết đã được loại bỏ, giúp mô hình tập trung vào các đặc trưng quan trọng.
- * Việc chuẩn hóa và loại bỏ nhiễu trong văn bản dẫn đến sự đồng nhất cao hơn giữa các mẫu, cải thiện khả năng học của mô hình Deep Learning, đặc biệt là trong việc khai thác mối quan hệ ngữ nghĩa giữa các từ và cụm từ.
- * N-grams sau tiền xử lý cũng thể hiện sự tập trung vào các cụm từ quan trọng, phản ánh ngữ cảnh và nội dung đặc trưng của bộ dữ liệu. Điều này mang lại lợi ích lớn cho các mô hình dự đoán dựa trên ngữ cảnh.

3.2.1.8 Khám phá Answer[A/B/C/D]Text

Tương tự ở phần trước, quan sát phân bố kích thước của câu trả lời:



Hình 30: Các biểu đồ biểu diễn sự phân bố kích thước của câu trả lời khi chưa tiền xử lý

– Phân bố độ dài:

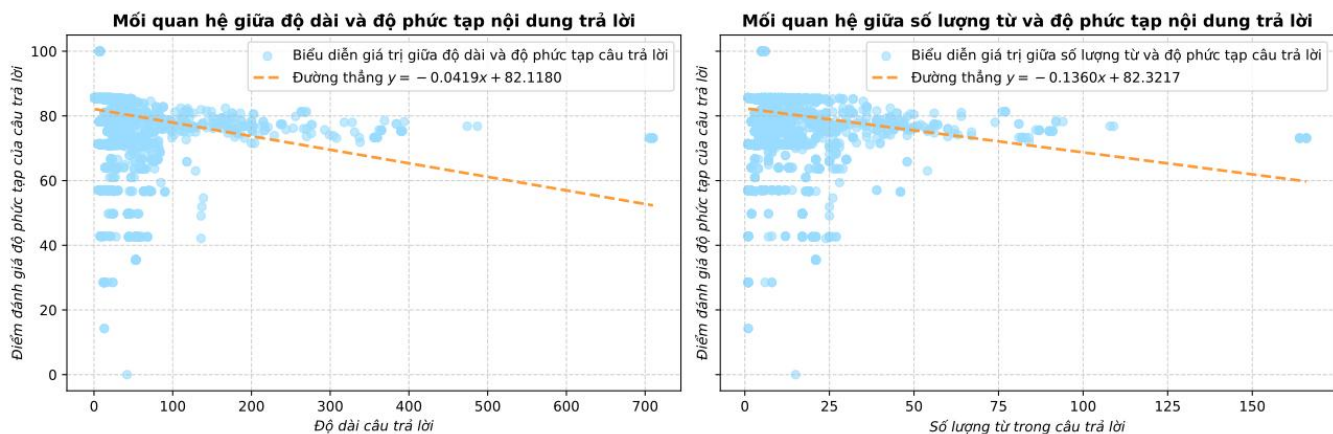
- * Độ dài của câu trả lời chủ yếu tập trung trong khoảng từ 8 đến 19 ký tự, với trung vị 12;
- * Số lượng câu trả lời có độ dài xuất hiện ở các giá trị lớn hơn 50 ký tự, như được minh họa qua các điểm ngoài trong biểu đồ boxplot;
- * Giá trị trung bình cao hơn tứ phân vị thứ III, cho thấy dữ liệu bị lệch dương rất

nhiều, cộng thêm phân bố dữ liệu bị kéo dài ở phần đuôi phải, làm cho đồ thị có phần đuôi phải dày.

– **Phân bố số lượng từ:** Nhóm sử dụng hàm `word_tokenize` của thư viện `nltk` để đếm số lượng từ trong câu hỏi:

- * Số lượng từ câu hỏi tập trung nhiều ở khoảng từ 5 đến 9, với với trung vị 5;
- * Vẫn có một số câu hỏi có số lượng từ xuất hiện ở các giá trị lớn hơn 20, như được minh họa qua các điểm ngoài trong biểu đồ `boxplot`;
- * Giá trị trung bình bám sát tứ phân vị thứ III, cho thấy dữ liệu bị lệch dương, cộng thêm phân bố dữ liệu bị kéo dài ở phần đuôi phải, làm cho đồ thị có phần đuôi phải dày.

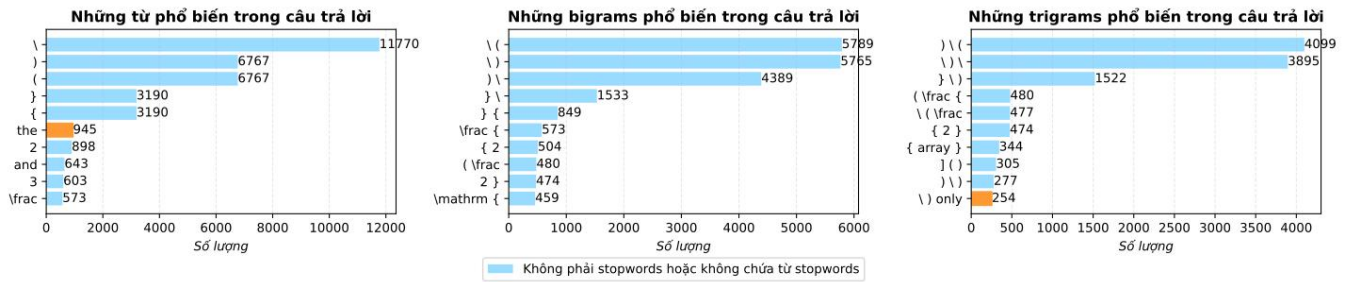
– **Điểm đánh giá mức độ phức tạp:** Bởi vì nội dung dữ liệu tập trung xoay quanh bài toán toán học nên đôi khi có một số câu trả lời có nội dung phức tạp gây khó hiểu. Độ phức tạp của câu trả lời được đánh giá dựa trên chỉ số `readability` (độ dễ đọc) giúp đánh giá mức độ khó hay dễ hiểu của một đoạn văn bản, thường dựa vào cấu trúc câu, số từ, độ dài từ, và các yếu tố ngôn ngữ khác; bằng cách tính điểm Flesch Reading Ease Score.



Hình 31: Các biểu đồ biểu diễn mối quan hệ điểm phức tạp của câu trả lời với độ dài và số lượng từ khi chưa tiền xử lý

Kết quả trực quan cho thấy nội dung câu trả lời cần phải có sự tư duy để hiểu, đồng thời độ dài và số lượng từ của câu trả lời cũng có thể nói lên phần nào nội dung có phức tạp hay không, tuy nhiên kết quả kiểm định mức độ phù hợp khá thấp.

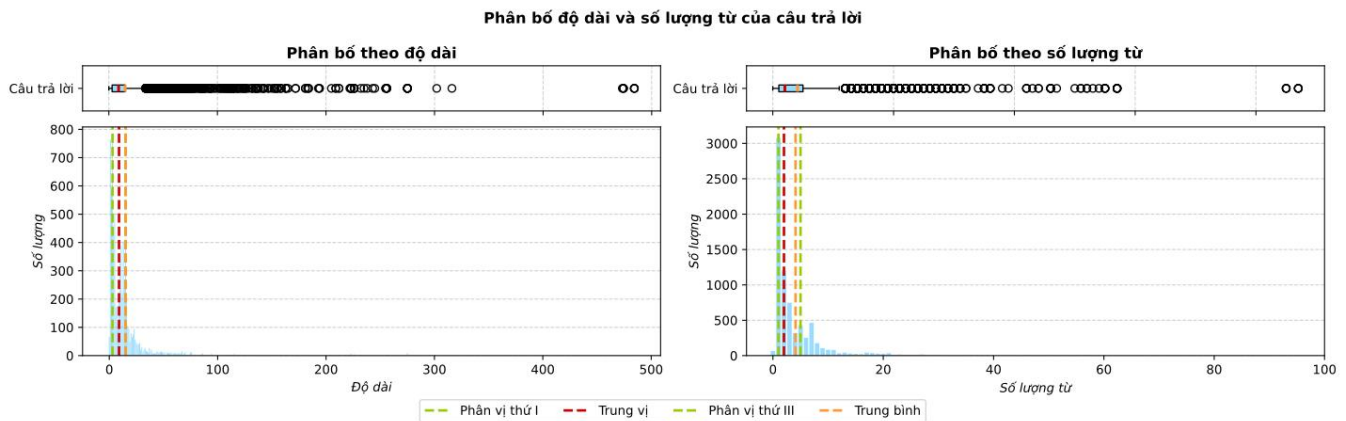
- **Những từ và n-grams phổ biến:** Nhóm sử dụng hàm `word_tokenize` của thư viện `nltk` để đếm số lượng từ trong câu hỏi (trực quan tại hình 32).



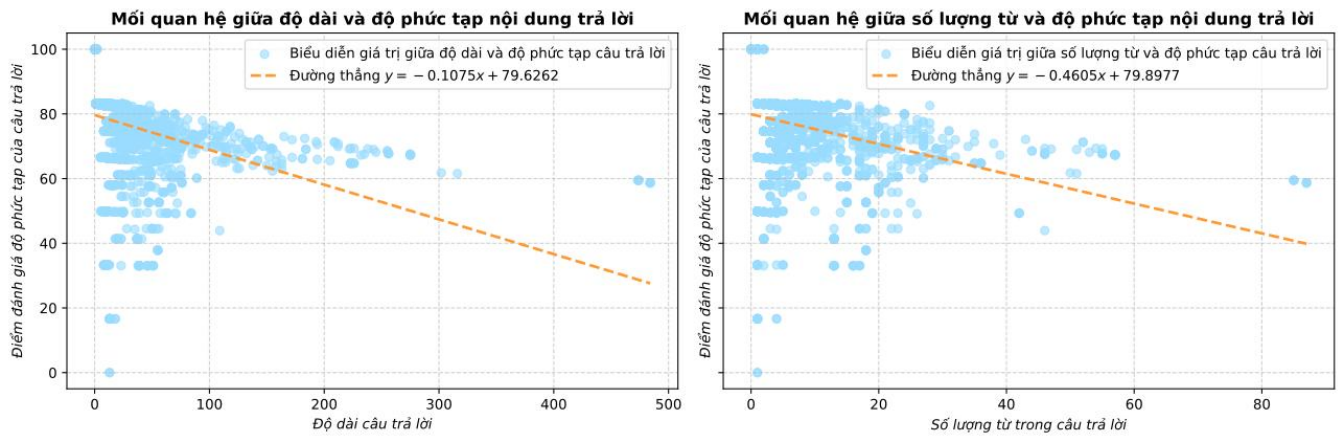
Hình 32: Các biểu đồ thống kê các từ và n-grams phổ biến của câu trả lời khi chưa xử lý nội dung văn bản

Đa số các từ và n-grams phổ biến của câu hỏi chưa qua xử lý nội dung văn bản là các cú pháp $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

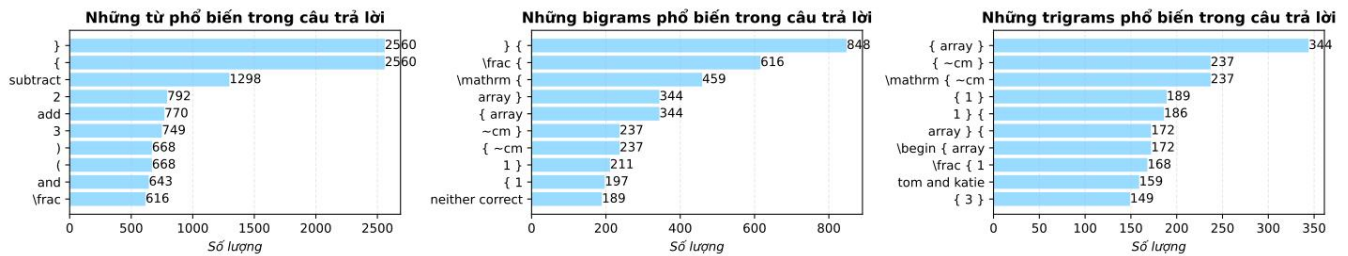
Nhóm thực hiện một số xử lý nội dung văn bản của câu trả lời để khám phá cốt lõi hơn. Cụ thể, nhóm thực hiện xử lý nội dung bằng thuật toán 2. Kết quả sau khi xử lý nội dung văn bản của câu trả lời: (trực quan tại hình 33, 34 và 35)



Hình 33: Các biểu đồ biểu diễn sự phân bố kích thước của câu trả lời đã tiền xử lý nội dung văn bản



Hình 34: Các biểu đồ biểu diễn mối quan hệ điểm phức tạp của câu trả lời với độ dài và số lượng từ đã tiền xử lý



Hình 35: Các biểu đồ thống kê các từ và n -grams phổ biến của câu trả lời đã tiền xử lý nội dung văn bản

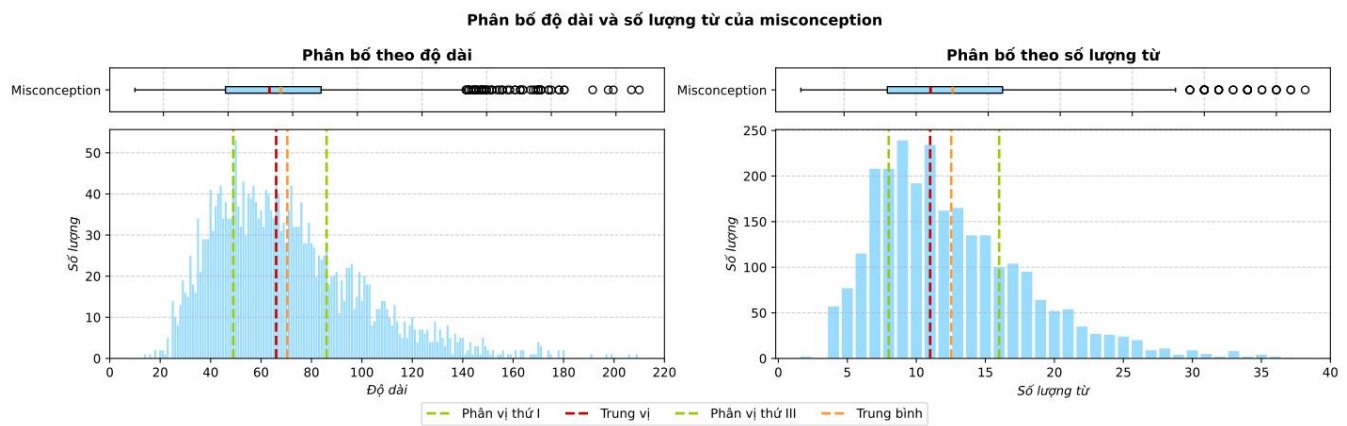
- **Phân bố độ dài và số lượng từ:** Không có sự thay đổi quá nhiều ở các số liệu trung vị và tứ phân vị, điểm thay đổi có thể thấy rõ rệt là phần đuôi dày bên phải được hạn chế không còn kéo quá dài.
- **Điểm đánh giá mức độ phức tạp:** Việc tiền xử lý dữ liệu không làm nội dung câu trả lời trở nên dễ hiểu hơn. Bản chất nội dung câu trả lời vẫn đòi hỏi việc tư duy để lựa chọn đáp án đúng.
- **Những từ và n -grams phổ biến:**
 - * Sau khi tiền xử lý, các từ và n -grams phổ biến trở nên rõ ràng hơn, với sự xuất hiện chủ yếu của các từ khóa liên quan đến nội dung cốt lõi của câu trả lời. Các stopwords và cú pháp đặc biệt không cần thiết đã được loại bỏ, giúp mô hình tập trung vào các đặc trưng quan trọng.
 - * Việc chuẩn hóa và loại bỏ nhiễu trong văn bản dẫn đến sự đồng nhất cao hơn giữa

các mẫu, cải thiện khả năng học của mô hình Deep Learning, đặc biệt là trong việc khai thác mối quan hệ ngữ nghĩa giữa các từ và cụm từ.

- * **N-grams** sau tiền xử lý cũng thể hiện sự tập trung vào các cụm từ quan trọng, phản ánh ngữ cảnh và nội dung đặc trưng của bộ dữ liệu. Điều này mang lại lợi ích lớn cho các mô hình dự đoán dựa trên ngữ cảnh.

3.2.2 Dữ liệu trong misconception_mapping.csv

Quan sát phân bố kích thước của **construct**:



Hình 36: Các biểu đồ biểu diễn sự phân bố kích thước của **construct** khi chưa xử lý nội dung văn bản

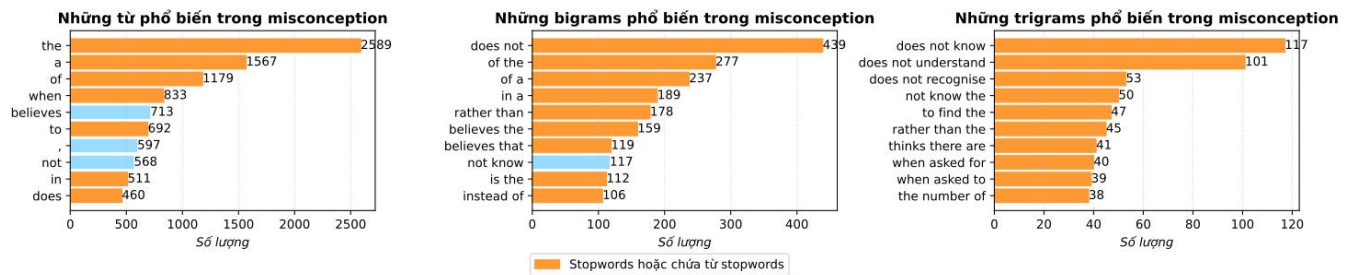
– Phân bố độ dài:

- * Độ dài của **misconception** chủ yếu tập trung trong khoảng từ 49 đến 86 ký tự, với trung vị 66;
- * Vẫn có một số **misconception** có độ dài xuất hiện ở các giá trị lớn hơn 140 ký tự, như được minh họa qua các điểm ngoài trong biểu đồ **boxplot**;
- * Giá trị trung bình cao hơn trung vị, cho thấy dữ liệu bị lệch dương, cộng thêm phân bố dữ liệu bị kéo dài ở phần đuôi phải, làm cho đồ thị có phần đuôi phải dày.

– Phân bố số lượng từ: Nhóm sử dụng hàm **word_tokenize** của thư viện **nltk** để đếm số lượng từ trong **misconception**:

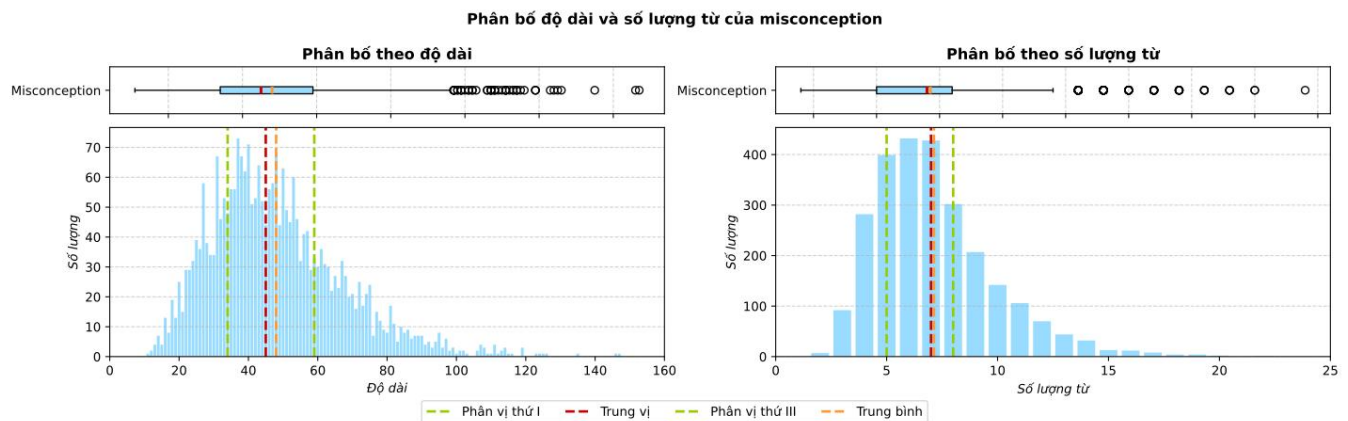
- * Số lượng từ **misconception** tập trung nhiều ở khoảng từ 8 đến 16, với với trung vị 11;

- * Vẫn có một số **misconception** có số lượng từ xuất hiện ở các giá trị lớn hơn 30, như được minh họa qua các điểm ngoài trong biểu đồ **boxplot**;
 - * Giá trị trung bình cao hơn trung vị, cho thấy dữ liệu bị lệch dương, cộng thêm phân bố dữ liệu bị kéo dài ở phần đuôi phải, làm cho đồ thị có phần đuôi phải dày.
- **Những từ và n-grams phổ biến:** Nhóm sử dụng hàm `word_tokenize` của thư viện `nltk` để đếm số lượng từ trong **construct** (trực quan tại hình 37): Đa số các từ và **n-grams** chứa các từ không cung cấp nhiều ý nghĩa **stopwords**.

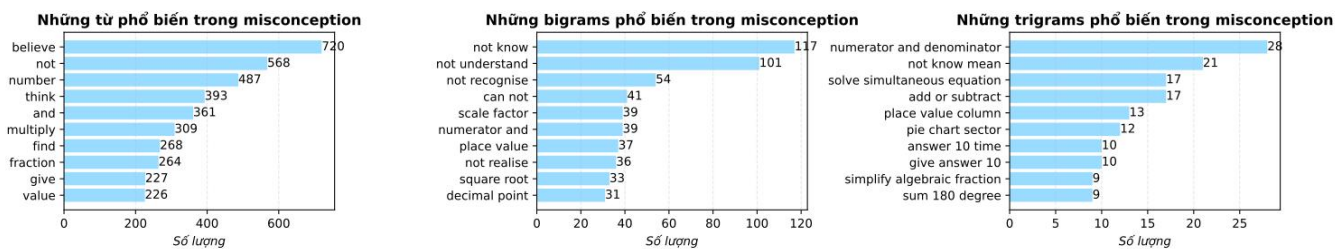


Hình 37: Các biểu đồ thống kê các từ và **n-grams** phổ biến của **construct** khi chưa xử lý nội dung văn bản

Nhóm thực hiện một số xử lý nội dung văn bản của **construct** để khám phá cốt lõi hơn. Cụ thể, nhóm thực hiện xử lý nội dung bằng thuật toán 1. Kết quả sau khi xử lý nội dung văn bản của **construct**: (trực quan tại hình 38 và 39)



Hình 38: Các biểu đồ biểu diễn sự phân bố kích thước của **construct** đã xử lý nội dung văn bản



Hình 39: Các biểu đồ thống kê các từ và n-grams phổ biến của construct đã xử lý nội dung văn bản

– Phân bố độ dài:

- * Độ dài của các misconception giảm đáng kể, từ trung vị 66 ký tự xuống còn khoảng 45 ký tự, tập trung trong [34, 59] ký tự. Điều này cho thấy hiệu quả của các bước tiền xử lý trong việc loại bỏ nhiễu và chuẩn hóa văn bản;
- * Sự giảm này giúp đơn giản hóa dữ liệu đầu vào, giảm độ phức tạp tính toán cho mô hình và cải thiện khả năng học tập trên các đặc trưng quan trọng hơn.

– Phân bố số lượng từ:

- * Số lượng từ trong misconception giảm từ trung vị 11 xuống khoảng 7, tập trung chủ yếu trong [5, 8] từ. Điều này phản ánh việc tinh gọn văn bản, tập trung vào các thông tin cốt lõi và loại bỏ từ ít ý nghĩa;
- * Phân bố số lượng từ sau xử lý cũng thu hẹp khoảng cách giữa giá trị trung bình và trung vị, giảm thiểu đuôi phải dày, giúp mô hình nhận diện đặc trưng ngữ nghĩa dễ dàng hơn, đồng thời tối ưu tài nguyên khi huấn luyện.

– Những từ và n-grams phổ biến:

- * Các từ và n-grams phổ biến trở nên đồng nhất và tập trung hơn vào từ khóa sau khi loại bỏ stopwords và thực hiện lemmatization. Điều này làm nổi bật các yếu tố quan trọng trong misconception;
- * Việc đồng nhất hóa này không chỉ cải thiện chất lượng đặc trưng mà còn giúp mô hình dễ dàng khai thác các quan hệ ngữ nghĩa, từ đó nâng cao hiệu suất dự đoán.

3.3 Khám phá ngữ nghĩa

3.3.1 Phương pháp thực hiện

Mục tiêu của phương pháp tiếp cận này là để hiểu khái quát nội dung ngữ nghĩa của bộ dữ liệu huấn luyện được cho. Ngữ nghĩa ở đây nói đến việc hiểu chi tiết hơn mỗi cột về ý nghĩa và giới hạn các ý tưởng/chủ đề đại diện trong một cột dữ liệu cụ thể.

Để làm được điều này nhóm sử dụng mô hình Large Language Model - cụ thể là mô hình **Claude Sonnet 3.5** của Anthropic. Nhóm đưa ra ý tưởng này vì:

- Nhóm có thực hiện một số thử nghiệm sử dụng clustering bằng TFIDF và sử dụng mô hình embedding cùng với TF-IDF clustering, nhưng kết quả không tốt - và khó xác định chủ đề của các cluster.
- Mô hình Large Language Model có khả năng phân tích và "hiểu" ngôn ngữ rất tốt. Hiện nay, các mô hình này rất dễ tiếp cận và sử dụng.

Sau khi thử nghiệm với việc sử dụng mô hình LLM để thực hiện semantic segmentation, nhóm nhận thấy là phương pháp này có kết quả tốt, và dễ điều chỉnh để đưa ra căn chỉnh đúng với ý tưởng đã có/khám phá ý tưởng mới.

Chi tiết các Prompts được sử dụng đối với phân tích mỗi trường dữ liệu được lưu ở mục **other-source-code**, với:

- Prompting ứng với phân tích **SubjectName**: SubjectNamePrompting.json
- Prompting ứng với phân tích **ConstructName**: ConstructNamePrompting.json
- Prompting ứng với phân tích **Misconception**: MisconceptionPrompting(1).json và MisconceptionPrompting(2).json

Note: Nguyên nhân nhóm cung cấp thông tin prompting qua file **.json** là vì nền tảng Claude Sonnet 3.5 chỉ cung cấp phương pháp export dữ liệu qua file **.json**.

3.3.2 Phương pháp xử lý dữ liệu

Dữ liệu được xử lý tương tự như phương pháp trước: thông tin của mỗi đáp án được chuyển thành từng dòng riêng, tức là đơn vị dòng chuyển từ câu hỏi sang đáp án. Chỉ những đáp án có Misconception tương ứng mới được giữ lại để phân tích trong bộ dữ liệu.

Đối với lựa chọn dữ liệu để phù hợp với context window của LLM, nhóm sử dụng các phương pháp khác nhau để xử lý cho mỗi cột dữ liệu khác nhau:

- **Subject:** Vì số lượng bộ môn và độ dài tên bộ môn - trên trung bình - khá ngắn, nên có thể sử dụng toàn bộ dữ liệu danh sách bộ môn.
- **Construct:** Số lượng và độ dài của các kỹ năng (Construct) lớn hơn nhiều so với bộ môn, chúng ta chỉ sử dụng 3-grams đầu tiên của mỗi Construct.
- **Misconception:** Giống như Construct.

3.3.3 Khám phá dữ liệu

3.3.3.1 Khám phá Subject

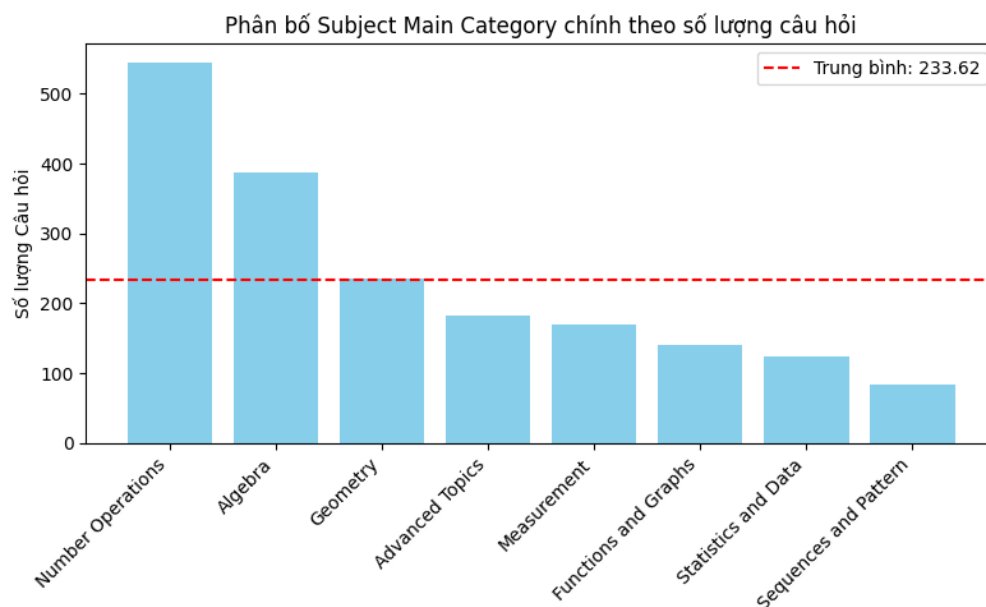
Phân loại:

- **Number Operations:** Bao gồm các Phép tính toán cơ bản (cộng, trừ, nhân, chia...); Thao tác với phân số, số thập phân, phần trăm; Ước lượng; Ước số và bội số
- **Algebra:** Bao gồm các bài toán về Đại số cơ bản; Phương trình; Phân thức đại số; Dấu ngoặc; Phân tích đa thức
- **Geometry:** Bao gồm các bài toán về Hình học 2 chiều; Hình học 3 chiều; Góc; Đường thẳng; Phép biến hình
- **Functions and Graphs:** Bao gồm các bài toán về Hình học tọa độ; Hàm số tuyến tính; Hàm số phi tuyến
- **Measurement:** Bao gồm các bài toán về Độ dài và Diện tích; Thể tích; Tỷ lệ; Các đơn vị đo khác
- **Statistics and Data:** Bao gồm các bài toán về Biểu diễn dữ liệu; Phân tích dữ liệu; Xác suất

- **Sequence and Pattern:** Bao gồm các bài toán về Dãy số
- **Advance Topics:** Bao gồm các bài toán về Lượng giác; Định lý Pythagoras; Số mũ và căn; Tỷ lệ và các chủ đề nâng cao khác.

Khám phá

Câu hỏi 01: Các chủ đề toán học được phân bố như thế nào?

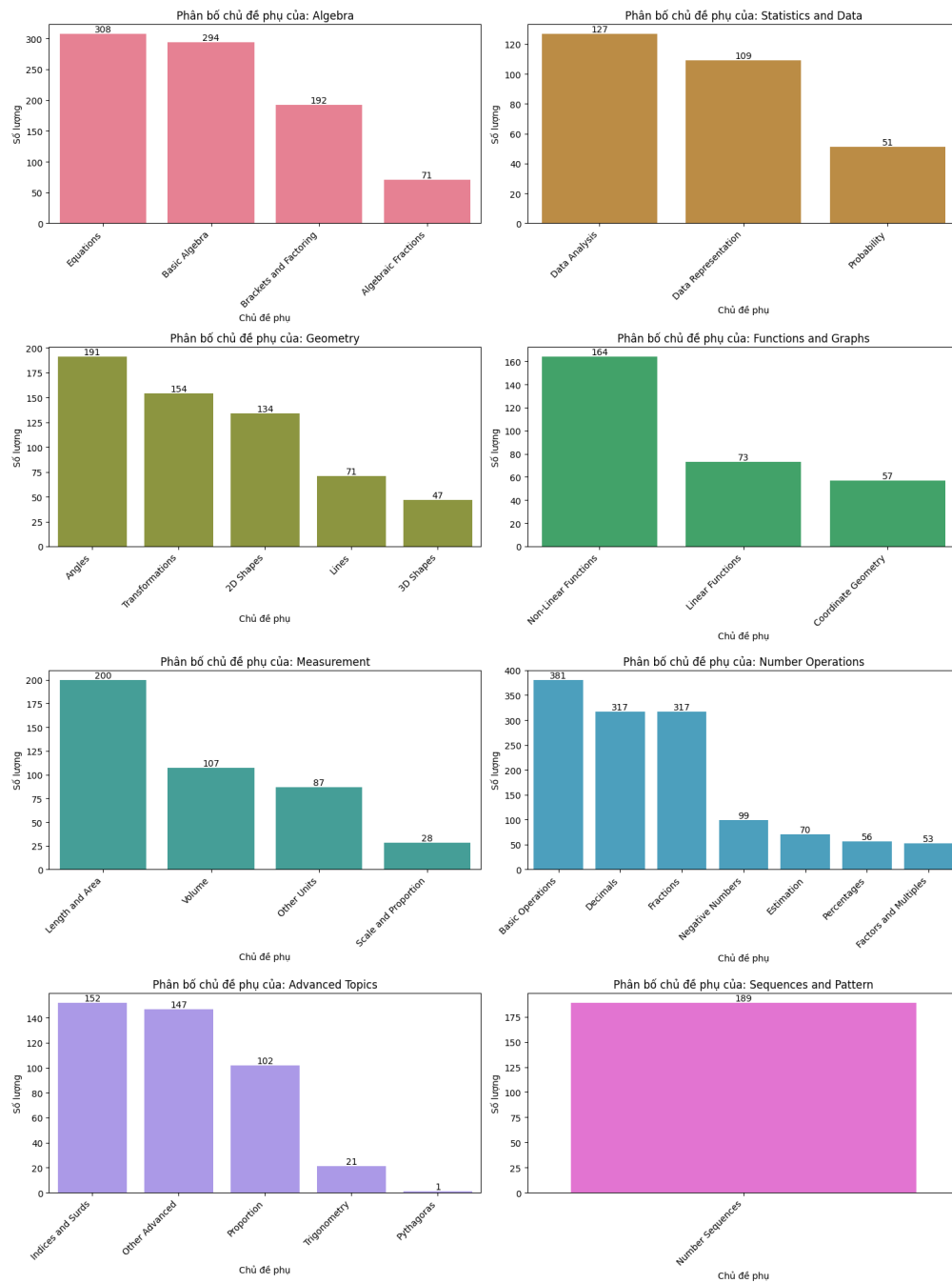


Hình 40: Biểu đồ cột phân bố số lượng câu hỏi thuộc về mỗi Chủ đề toán học

Nhận xét:

- Chủ đề phổ biến nhất là **Number Operations**. Điều này có thể chỉ ra rằng phần lớn các môn học trong tập dữ liệu thuộc cấp độ Toán Tiểu học.
- **Number Operations** và **Algebra** chiếm khoảng một nửa số **QuestionId** có trong tập dữ liệu.
- **Sequence and Pattern** chứa ít nhất số lượng **QuestionId**, với chưa đến 100 câu hỏi trong tập dữ liệu.

Câu hỏi 02: Các chủ đề toán học phụ được phân bố như thế nào?



Hình 41: Biểu đồ cột phân bố số lượng câu hỏi thuộc về mỗi Chủ đề phụ toán học

Nhận xét:

- Các SubjectSubCategory có phân bố tỉ lệ không đồng đều, với một số chủ đề phụ (thường là 1 đến 2) bao gồm phần lớn các câu hỏi thuộc mỗi chủ đề chính.
- Number Operations là chủ đề có nhiều chủ đề phụ nhất với 6 SubjectSubCategory.

- **Sequence and Pattern** là chủ đề có ít chủ đề phụ nhất với 1 **SubjectSubCategory**.
- Về số lượng, 5 chủ đề phụ có số lượng lớn nhất là:
 - * **Number Operations - Basic Operations**: 381
 - * **Number Operations - Decimals**: 317
 - * **Number Operations - Fractions**: 317
 - * **Algebra - Equations**: 308
 - * **Algebra - Basic Algebra**: 294
- Phản ánh tỉ lệ các chủ đề chính mà chúng ta đã khám phá. Số lượng các chủ đề liên quan đến tính toán với số có tỉ lệ xuất hiện cao.

Đối với từng SubjectMainCategory:

1. Algebra:

- * Có 4 chủ đề con là **Equation** (Phương trình), **Basic Algebra** (Đại số cơ bản), **Bracket and Factoring** (Bài toán về ngoặc và phân tích thành nhân tử) và **Algebraic Fractions** (Phân số đại số).
- * Chủ đề phụ về Phương trình có tỉ lệ xuất hiện cao nhất, và các chủ đề phụ về Đại số cơ bản có số lượt ít hơn xấp xỉ.

2. Statistic and Data:

- * Có 3 chủ đề con là **Data Analysis** (Phân tích dữ liệu), **Data Representation** (Biểu diễn dữ liệu) và **Probability** (Xác suất).
- * Chủ đề phụ về Phân tích dữ liệu có tỉ lệ xuất hiện cao nhất, và chủ đề về Biểu diễn dữ liệu có số lượng ít hơn xấp xỉ.

3. Geometry:

- * Có 5 chủ đề con là **Angles** (Góc cạnh), **Transformation** (Biến đổi hình học), **2D shapes** (Hình học 2 chiều), **Lines** (Đường thẳng) và **3D shapes** (Hình học 3 chiều).
- * Chủ đề phụ về Góc cạnh có tỉ lệ xuất hiện cao nhất. Chủ đề về Hình học 3 chiều là ít nhất.

4. Functions and Graphs:

- * Có 3 chủ đề con là **Non-Linear Function** (Phương trình phi tuyến), **Linear Function** (Phương trình tuyến tính) và **Coordinate geometry** (Hình học tọa độ).
- * Chủ đề phụ về Phương trình phi tuyến có tỉ lệ xuất hiện cao nhất. Chủ đề về Hình học tọa độ là ít nhất.

5. Measurement:

- * Có 4 chủ đề con là **Length and Area** (Chiều dài và diện tích), **Volume** (Thể tích), **Other Units** (Các đơn vị đo khác) và **Scale and proportion** (Quy mô và tỉ lệ).
- * Chủ đề phụ về Chiều dài và diện tích có tỉ lệ xuất hiện cao nhất. Chủ đề về Quy mô và tỉ lệ là thấp nhất (dưới 30 câu hỏi).

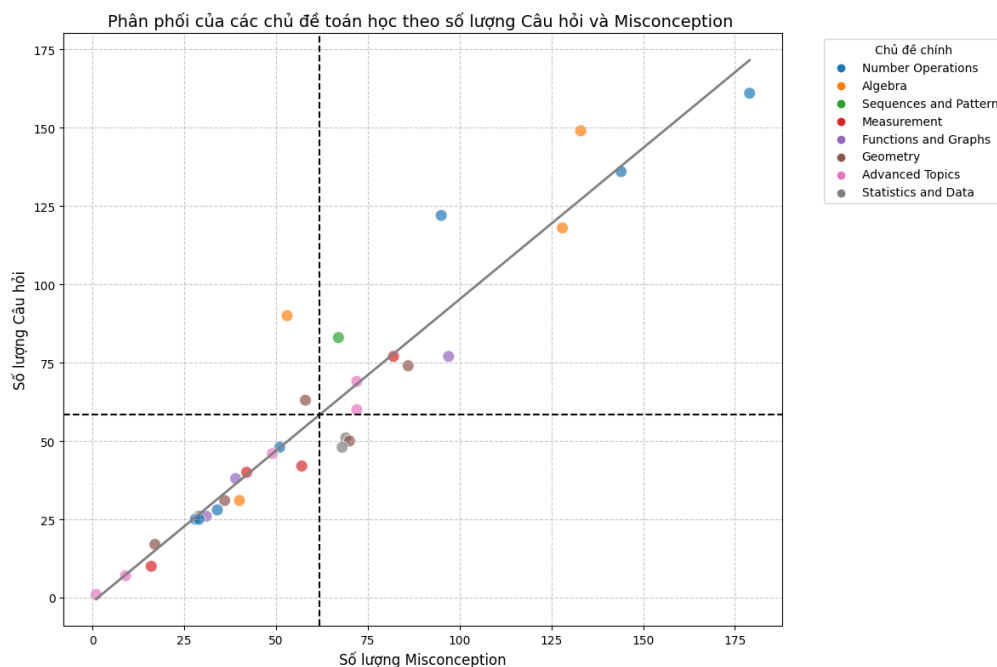
6. Number Operation:

- * Có 6 chủ đề con là **Basic Operation** (Phép toán cơ bản), **Fraction** (Phân số), **Decimal** (Số thập phân), **Negative numbers** (Số âm), **Estimation** (Ước lượng), **Percentage** (Phần trăm), **Factors and multiples** (Thừa số và bội số).
- * Các chủ đề chia ra phân bố khá đồng đều: Một nửa có số lần xuất hiện cao xấp xỉ nhau (Phép toán cơ bản, Phân số, Số thập phân) và một nửa có số lượt xuất hiện thấp xấp xỉ nhau (Số âm, Ước lượng, Phần trăm, Thừa số và bội số).

7. Advanced topics:

- * Có 5 chủ đề con là **Indices and Surds** (Chỉ số và căn thức), **Other advanced** (Các bài toán nâng cao khác), **Proportion** (Tỉ lệ), **Trigonometry** (Đại số lượng giác) và **Pythagoras** (Các bài toán liên quan đến Pythagoras).
- * Chủ đề về Chỉ số và căn thức có số lần xuất hiện nhiều nhất. Các bài toán về Pythagoras xuất hiện ít nhất với chỉ 1 bài toán.

Câu hỏi 03: Các chủ đề toán học được phân phối như thế nào dựa trên số lượng câu hỏi và Misconception tương ứng?



Hình 42: Biểu đồ phân phối các Chủ đề toán học theo số lượng Câu hỏi và Misconception

Các đường màu xám phía trên là đường hồi quy tuyến tính được điều chỉnh cho các điểm trong đồ thị.

Đường chấm ngang đại diện cho đường trung bình của số lượng Misconception.

Đường chấm dọc đại diện cho đường trung bình của số lượng QuestionId.

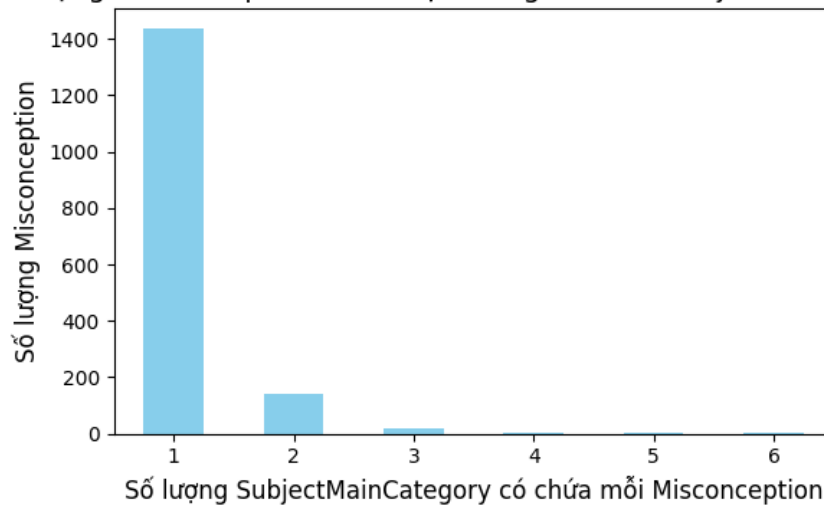
Nhận xét:

- Các điểm có hướng theo một đường tuyến tính. Điều này có nghĩa là đối với hầu hết các SubjectMainCategory, việc có nhiều QuestionId hơn cũng liên quan đến việc có nhiều loại Misconception hơn. Điều này cho thấy số lượng câu hỏi có thể đại diện cho sự đa dạng của các vấn đề trong từng danh mục - điều này liên quan đến các quan niệm sai lầm cụ thể cho từng câu hỏi.
- Bằng cách phân chia phân phối dựa trên các đường trung bình của trục x và y, ta có thể chia nó thành bốn phần tư:
 - * Phần tư trên bên phải đại diện cho các điểm có số lượng câu hỏi và quan niệm sai lầm lớn nhất.
 - * Phần tư dưới bên trái đại diện cho các điểm có số lượng câu hỏi và quan niệm sai lầm ít nhất.

- * Phần tư trên bên trái đại diện cho các điểm có số lượng câu hỏi cao hơn nhưng ít quan niệm sai lầm hơn.
- * Phần tư dưới bên phải đại diện cho các điểm có số lượng câu hỏi nhỏ hơn.
- **Number Operations** và **Algebra** có sự phân bố khá rộng - với các điểm nằm ở cả phần tư trên bên phải và phần tư dưới bên trái.
- Các danh mục khác ít phân tán hơn - tập trung xung quanh phía bên trái của giao điểm đường trung bình x-y.

Câu hỏi 04: Có bao Misconception xuất hiện trong 1,2,3... chủ đề toán học?

Số lượng Misconception xuất hiện trong 1/nhiều SubjectMainCategory

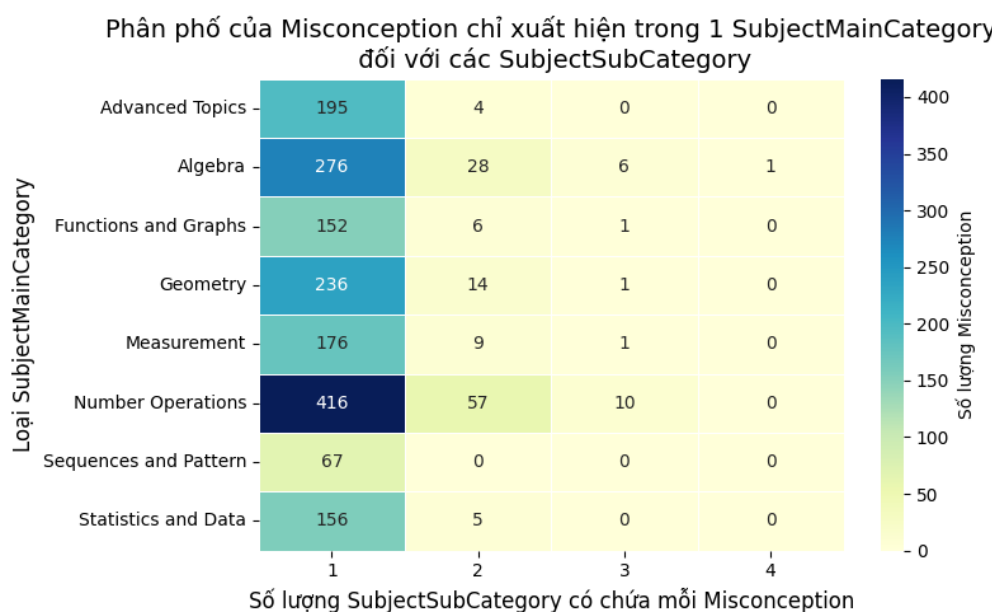


Hình 43: Biểu đồ phân bố số lượng Misconception xuất hiện trong các số lượng chủ đề chung

Nhận xét:

- Phần lớn các quan niệm sai lầm chỉ thuộc về 1 SubjectMainCategory (> 1400 Misconceptions $\sim 78\%$).
- Số lượng SubjectMainCategory cao nhất mà một Misconception có thể thuộc về là 6.

Câu hỏi 05: Đối với các Misconception chỉ xuất hiện trong 1 chủ đề toán học, có bao nhiêu Misconception xuất hiện trong 1,2,3... chủ đề phụ?

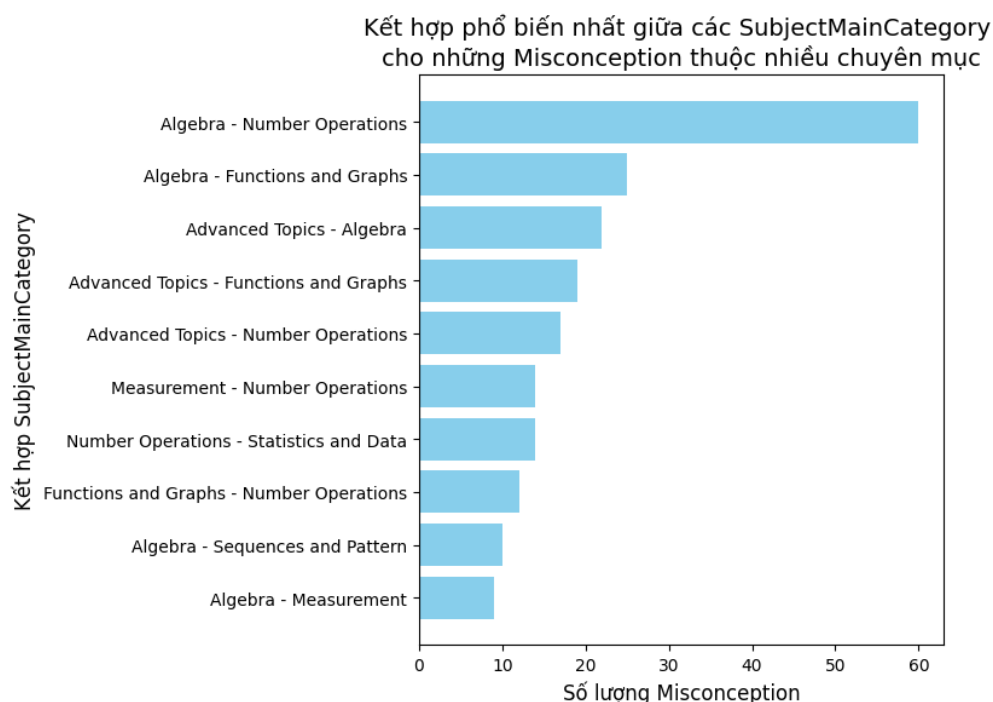


Hình 44: Biểu đồ phân bố các Misconception xuất hiện trong số lượng chủ đề phụ chung (Đối với Misconception chỉ xuất hiện trong 1 chủ đề)

Nhận xét:

- Phần lớn các Misconceptions chỉ xuất hiện trong 1 SubjectSubCategory.
- Algebra là chủ đề duy nhất có Misconception xuất hiện ở cả 4 SubjectSubCategory.
- Number Operations, Measurement, Geometry, Functions and Graphs có Misconception xuất hiện ở 3 SubjectSubCategory. Trong đó, Measurement, Geometry, Functions and Graphs chỉ có 1 Misconception chung cho 3 chủ đề phụ.
- Statics and Data, Advanced Topics có Misconception xuất hiện ở 2 SubjectSubCategory.

Câu hỏi 06: Đối với các Misconception xuất hiện trong 2 chủ đề toán học, các sự kết hợp phổ biến nhất của các chủ đề là gì?



Hình 45: Biểu đồ các sự kết hợp phổ biến của các chủ đề toán học đối với Misconception thuộc 2 chủ đề

Nhận xét:

- Cặp kết hợp phổ biến nhất là **Algebra** và **Number Operations**. Lưu ý rằng đây cũng là hai SubjectMainCategory phổ biến nhất trong tập dữ liệu.
- **Algebra** có khả năng cao nhất chứa các Misconception giao thoa với các danh mục khác, xét theo số lượng.

3.3.3.2 Khám phá Construct Dựa trên dữ liệu, nhóm khám phá được các Construct thường được cấu trúc 3 phần:

- Action word: Nằm ở đầu của một construct. Đây là một động từ, biểu thị hành động cần thực hiện.
- Subject: Thường nằm ở giữa một construct. Miêu tả đối tượng cần đạt được khi thực hiện hành động.
- Context: Thường nằm ở cuối một construct. Miêu tả ngữ cảnh diễn ra hành động.

Do độ dài và số lượng các loại Construct, chúng ta không thể thực hiện đưa dữ liệu vào LLM vì vi phạm độ lớn của context window. Do vậy, trong phân tích nhóm chỉ thực hiện chiết xuất Action word. Để thực hiện việc này, nhóm lấy 3-grams đầu tiên của mỗi construct và kết hợp danh sách tạo thành bộ dữ liệu.

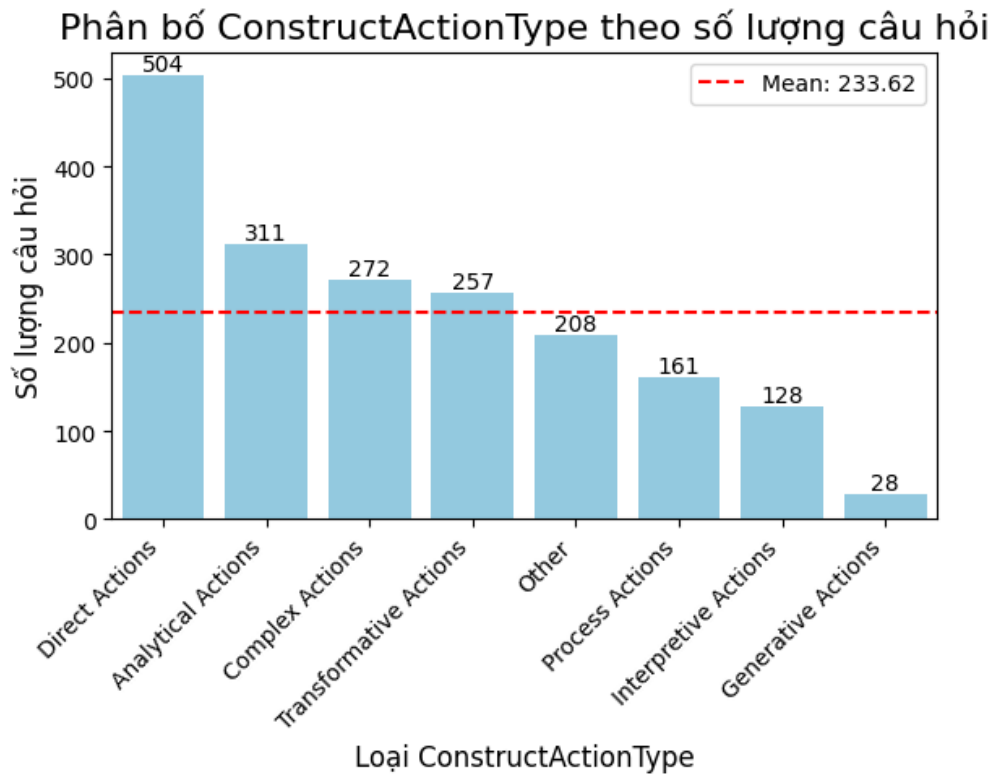
Do Action word biểu thị yêu cầu bài làm nên cũng có thể được coi là dạng đề bài. Trong bài khám phá này nhóm sử dụng dạng đề/construct mang tính có thể hoán đổi cho nhau.

Phân loại:

- Direct Actions: Hành động đơn giản, trực tiếp. Ví dụ: Cộng hai số; Đếm các số trong dãy...
- Transformative Actions: Hành động yêu cầu biến đổi một đối tượng. Ví dụ: Chuyển đổi phân số sang số thực; Làm tròn số,...
- Analytical Actions: Hành động yêu cầu đánh giá, phân tích. Ví dụ: Nhận diện tam giác, số bình phương; Sắp xếp số theo một trình tự nào đó...
- Complex Actions: Hành động nhiều bước hoặc trừu tượng. Ví dụ: Chứng minh mệnh đề; Rút gọn biểu thức đại số...
- Interpretive Actions: Hành động yêu cầu sự hiểu/giải thích. Ví dụ: Miêu tả một bài giải; Giải thích biểu đồ tròn...
- Process Actions: Hành động thực hiện theo quy trình. Ví dụ: Sử dụng hướng dẫn để thực hiện...; Sử dụng kiến thức về ... để thực hiện...
- Generative Actions: Hành động yêu cầu sáng tạo. Ví dụ: Vẽ biểu đồ; Tạo pictogram...
- Other: Hành động có ý nghĩa nội dung không phân chia rõ được vào các mục trên.

Khám phá

Câu hỏi 07: Các dạng bài toán được phân bố như thế nào?

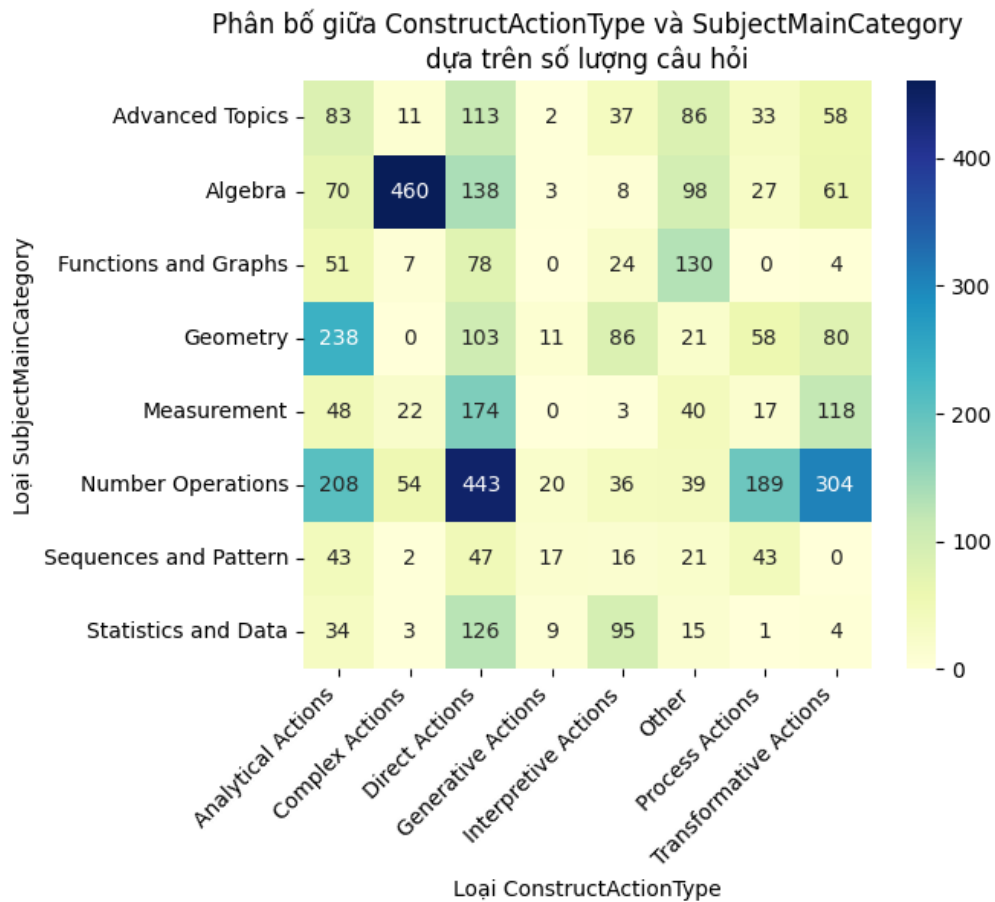


Hình 46: Biểu đồ phân bố số lượng câu hỏi thuộc về mỗi loại dạng bài toán (Construct Action Type)

Nhận xét:

- Construct Action phổ biến nhất là Direct Actions, Analytical Actions và Complex Actions.
- Construct Action ít phổ biến nhất là Generative Actions với chỉ 28 instances.
- Dựa vào đường trung bình, ta có thể phân chia các Construct Action thành 2 nhóm:
 - * Nhóm nửa trên trung bình: Direct Action, Analytical Actions, Complex Actions, Transformative Actions.
 - * Nhóm nửa dưới trung bình: Other, Process Actions, Interpretive Actions, Generative Actions.

Câu hỏi 08: Các bạng bài toán được phân phối như thế nào đối với các Chủ đề toán học dựa trên số lượng câu hỏi?



Hình 47: Biểu đồ phân bố giữa chủ đề toán học và dạng bài toán dựa trên số lượng câu hỏi

Nhận xét:

- Có những Construct Actions chủ yếu xuất hiện trong một loại SubjectCategory duy nhất: Complex Actions trong Algebra; Transformative Actions, Process Actions trong Number Operations.
- Có những Construct Actions xuất hiện trong nhiều SubjectCategory: Direct Actions trong Number Operations, Measurement, Statistics and Data, Algebra; Analytical Actions trong Number Operations và Geometry. Đây cũng là hai Construct Actions phổ biến nhất và thứ hai.
- Có những Construct Actions hiếm khi xuất hiện: Generative Actions.
- Number Operations có tập hợp các Construct Actions tập trung nhất.
- Sequence and Pattern có tập hợp các Construct Actions phân tán nhất.

Phân tích theo Construct Action:

- * **Direct Actions:** Được sử dụng nhiều nhất trong **Number Operations** nhưng cũng được sử dụng nhiều trong các chủ đề khác. Điều này có thể cho thấy rằng dạng bài này đại diện một phần cho các dạng câu hỏi **đơn giản, cơ bản** hơn của các chủ đề toán học.
- * **Analytical Actions:** Được sử dụng nhiều trong chủ đề **Geometry** và **Number Operation**. Điều này có thể cho thấy rằng so với các chủ đề khác thì Hình học và Tính toán với số yêu cầu đánh giá, **nhận diện** mẫu nhiều hơn so với các chủ đề khác.
- * **Complex Actions:** Được hầu hết sử dụng trong **Algebra**. Điều này có thể cho thấy rằng các bài toán thuộc **Algebra** có xác suất yêu cầu người học phải hiểu các ý tưởng **trừu tượng, thực hiện nhiều bước** hơn so với các chủ đề toán khác.
- * **Transformative Actions:** Được sử dụng hầu hết trong chủ đề về **Number Operations** và hầu hết không được sử dụng trong **Sequence and Pattern, Statistics and Data, Function and Graph**. Điều này có thể cho thấy rằng đối với các chủ đề không sử dụng nhiều dạng bài này không có cấu trúc của đối tượng dễ dàng chuyển đổi dạng - ví dụ như chuyển đổi đơn vị.
- * **Process Actions:** Được sử dụng hầu hết trong chủ đề về **Number Operations**. Điều này có thể cho thấy rằng các tác vụ có **quy trình** xử lý **rõ ràng** sẽ thường sẽ thuộc về Tính toán toán học, so với các chủ đề khác.
- * **Interpretive Actions:** Được sử dụng nhiều trong chủ đề **Statistics and Data** và **Geometry**. Điều này có thể cho thấy rằng các bài toán yêu cầu sự **miêu tả, giải thích** thì sẽ thuộc về các chủ đề về Thống kê và Hình học nhiều hơn - so với các chủ đề sử dụng rất ít dạng bài này như **Measurement** và **Algebra**.
- * **Generative Actions:** Hầu hết không được sử dụng nhiều, trong đó có những chủ đề không/sử dụng ít hơn 5 lần như **Measurement, Functions and Graphs, Algebra, Advanced Topics**. Điều này có thể cho thấy rằng đa số các dạng bài toán khác sẽ không có yêu cầu người học xây dựng cấu trúc đối tượng trong bài học (đối với đại số là phương trình, đối với Đo đạc là đơn vị...) - vì đối tượng của các chủ đề toán trên phức tạp hơn hoặc khó **minh họa** hơn.
- * **Other:** Hầu hết các chủ đề chưa được phân chia rõ hơn nằm ở chủ đề **Function and**

graph, Algebra, Advanced topics.

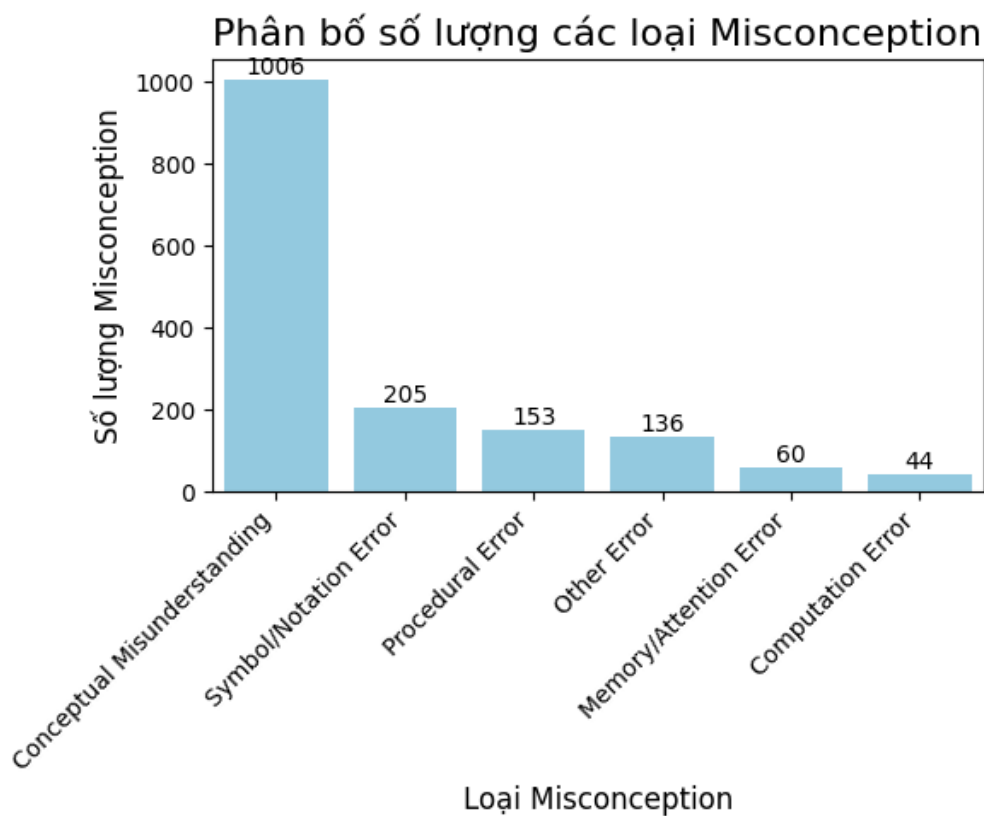
3.3.3.3 Khám phá Misconception Phương pháp xử lý dữ liệu của Misconception giống như cách xử lý Construct Action, nên nhóm sẽ không trình bày lại.

Phân loại:

- Conceptual Misunderstanding: Các lỗi liên quan đến mức hiểu, nhầm lẫn và việc xác định các khái niệm toán
- Procedural Error: Các lỗi liên quan đến thực hiện quy trình từng bước, dãy và biến hóa
- Computation Error: Các lỗi liên quan đến tính toán
- Memory/Attention Error: Các lỗi liên quan đến việc quên, bỏ qua dữ kiện trong trọng trong đề bài
- Symbol/Notation Error: Các lỗi dùng sai ký hiệu, biểu tượng trong bài
- Other Error: Các lỗi khác. Các lỗi này thường nằm trong các lỗi liên quan đến vẽ hình, thói quen làm bài (quá nhanh, chậm), chưa hoàn thành bài

Khám phá

Câu hỏi 09: Các misconception được phân phối như thế nào?



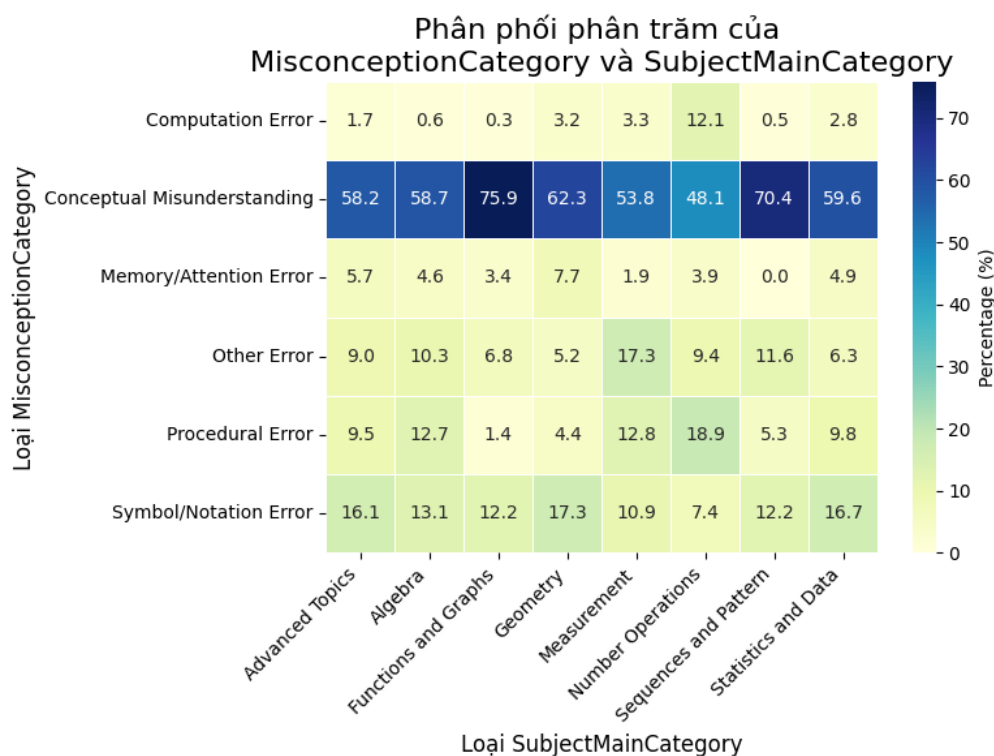
Hình 48: Biểu đồ phân bố số lượng loại Misconception dựa trên số lượng Misconception

Nhận xét:

- Conceptual Misunderstanding là loại lỗi phổ biến nhất với hơn 1000 Misconception.
- Các loại lỗi khác có số lượng Misconception thấp hơn, chỉ khoảng 200 - 40 Misconception.

Điều này có thể cho thấy rằng đa số lỗi người học gặp phải là do lỗi hiểu, mức hiểu bài trong quá trình học tập, thay vì các lỗi liên quan đến quá trình làm bài, tính toán.

Câu hỏi 10: Phân phối phần trăm của các Misconception theo các chủ đề toán học là gì?

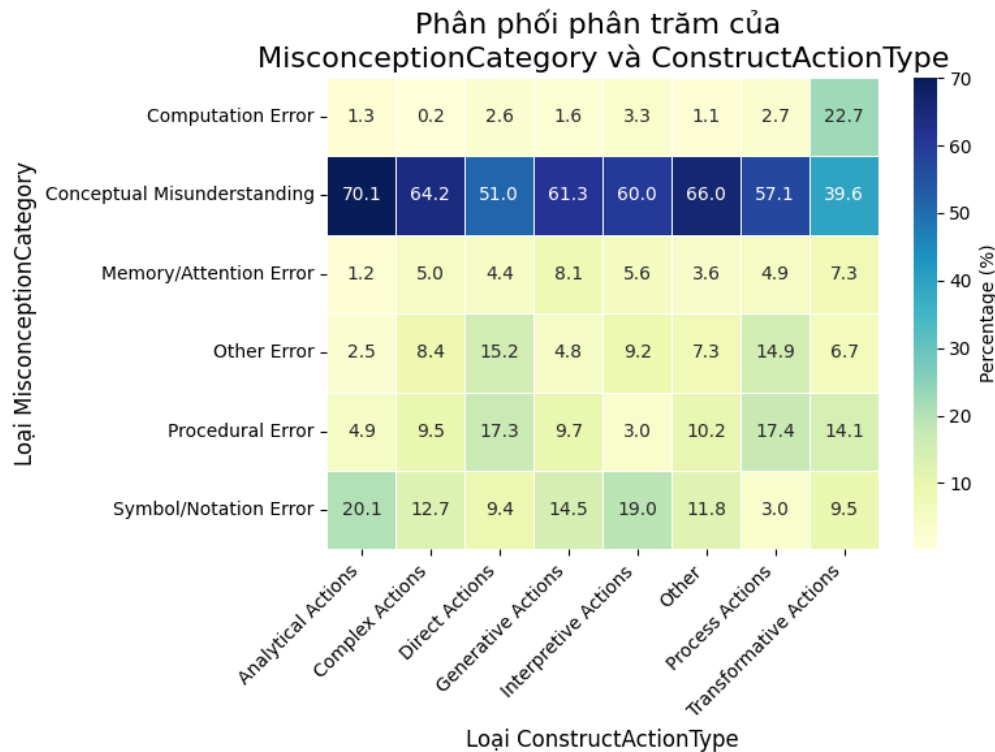


Hình 49: Biểu đồ phân bố phần trăm giữa phân loại Misconception và chủ đề toán học tính theo chủ đề toán học

Nhận xét:

- **Conceptual Misunderstanding** là loại lỗi được mắc phải nhiều nhất (từ 48.1% - 75.9%) đối với tất cả các chủ đề toán. Điều này đóng góp vào giả thuyết trên của chúng ta, rằng đa số lỗi của người học mắc phải liên quan đến mức hiểu đối với nội dung câu hỏi.
- **Computation Error** là loại lỗi ít bị mắc nhất của hầu như các chủ đề toán học, trừ **Number Operation** (12.1%). Tương tự, **Memory/Attention Error** cũng là một lỗi bị mắc khá ít (nhiều nhất chỉ 7.7% - thuộc về chủ đề **Geometry**), và thậm chí không xuất hiện trong chủ đề **Sequence and Pattern**.
- **Procedure Error** thường bị mắc bởi các chủ đề **Number Operation**, **Measurement**, **Algebra**, **Statistics and Data** ($\geq 9.5\%$), và ít xuất hiện trong chủ đề **Functions and Graphs**, **Geometry** và **Sequence and Pattern**.
- **Symbol/Notation Error** thường bị mắc phải ở tất cả các chủ đề, và thấp nhất ở **Number Operations** (7.4%).

Câu hỏi 11: Phân phối phần trăm của các Misconception theo các dạng bài toán là gì?



Hình 50: Biểu đồ phân bố phần trăm giữa phân loại Misconception và dạng bài tính theo dạng bài toán

Nhận xét:

- Giống như với biểu đồ trên, **Conceptual Misunderstanding** cũng là lỗi bị mắc phải nhiều nhất đối với tất cả các loại dạng bài (**Action**) (từ 39.6% - 70.1%).
- **Computation Error** và **Memory/Attention Error** lại là những lỗi **hầu như** ít bị mắc phải nhất, ngoại trừ trường hợp của **Computation Error** và **Transformation Actions**. Điều này có thể cho thấy rằng các dạng bài toán yêu cầu biến đổi thường yêu cầu tính toán phức tạp/dài, dẫn đến sai sót trong tính toán nhiều.
- Nhìn chung, người học thường sai một số loại lỗi (ngoại trừ **Conceptual Misunderstanding**) nhất định. Đối với các dạng bài yêu cầu phân tích, thấu hiểu, phức tạp hơn thì thường sai lỗi về **Symbol/Notation Error** hơn (**Analytical Actions**, **Complex Actions**, **Generative Actions**, **Interpretive Actions** - với tỉ lệ >12%), còn đối với dạng bài với yêu cầu cụ thể, dễ hiểu hơn thì thường sai lỗi **Procedure Error** (**Direct Action**, **Process Actions**).

3.3.3.4 Tổng hợp insights

- Các chủ đề trong bộ dữ liệu huấn luyện tập trung vào các nội dung toán học từ bậc tiểu học (**Number Operation**, **Measurement**) đến trung học cơ sở (các chủ đề khác).
- Chủ đề phổ biến nhất là **Number Operation** (Thao tác số học) và **Algebra** (Đại số).
- Các chủ đề toán học có nhiều câu hỏi (**QuestionId**) xuất hiện trong dữ liệu thường cũng có nhiều loại lỗi (**Misconception**) bị mắc phải.
- Đa số các lỗi (**Misconception**) chỉ thuộc về một chủ đề duy nhất. Trong đó, phần lớn các lỗi này chỉ liên quan đến một chủ đề phụ duy nhất, cho thấy rằng các lỗi thường rất cụ thể đối với một bối cảnh hoặc nội dung cụ thể.
- Đối với các lỗi liên quan đến hai chủ đề, đa số các cặp giao thoa giữa các chủ đề thường là các chủ đề phổ biến như **Number Operation** và **Algebra**.
- **Direct Actions** là dạng bài phổ biến nhất, được sử dụng trong tất cả các chủ đề toán học. Điều này cho thấy nhiều câu hỏi tập trung vào việc kiểm tra các ý tưởng cơ bản của toán học.
- Một số loại dạng bài (**Action**) tập trung vào một chủ đề cụ thể (**Complex Actions** trong **Algebra**), một số khác xuất hiện ở nhiều chủ đề (**Direct Actions**), và có loại không tập trung vào bất kỳ chủ đề nào (**Generative Actions**).
- Mặc dù **Number Operation** và **Algebra** là hai chủ đề lớn, phân bố dạng bài của chúng khác nhau.
 - * **Number Operation** sử dụng tất cả các dạng bài đã phân tích.
 - * **Algebra** tập trung chủ yếu vào các dạng bài **Complex Actions** và **Direct Actions**.
- Phân tích tỉ lệ và phân phối theo chủ đề và dạng bài cho thấy người học thường mắc lỗi nhiều nhất là do hiểu sai nội dung (**Conceptual Misunderstanding**). Điều này có thể xuất phát từ việc chưa nắm rõ hoặc chưa học kỹ nội dung được kiểm tra.
- Các lỗi về trí nhớ và độ tập trung (**Memory/Attention Error**) và tính toán (**Computation Error**) là các lỗi ít bị mắc phải nhất, ngoại trừ dạng bài biến đổi (**Transformative Actions**) và chủ đề tính toán số (**Number Operation**) liên quan đến lỗi **Computation Error**.

- Hầu hết các chủ đề toán học đều có nhiều lỗi liên quan đến ký hiệu và ký pháp (Symbol/Notation Error).
- Nhìn chung:
 - * Đối với các dạng bài yêu cầu phân tích và hiểu sâu, người học thường mắc lỗi Symbol/Notation Error.
 - * Đối với các dạng bài có yêu cầu cụ thể hơn, lỗi phổ biến hơn là Procedure Error.

4 Xây dựng mô hình

4.1 Bài toán: Information Retrieval

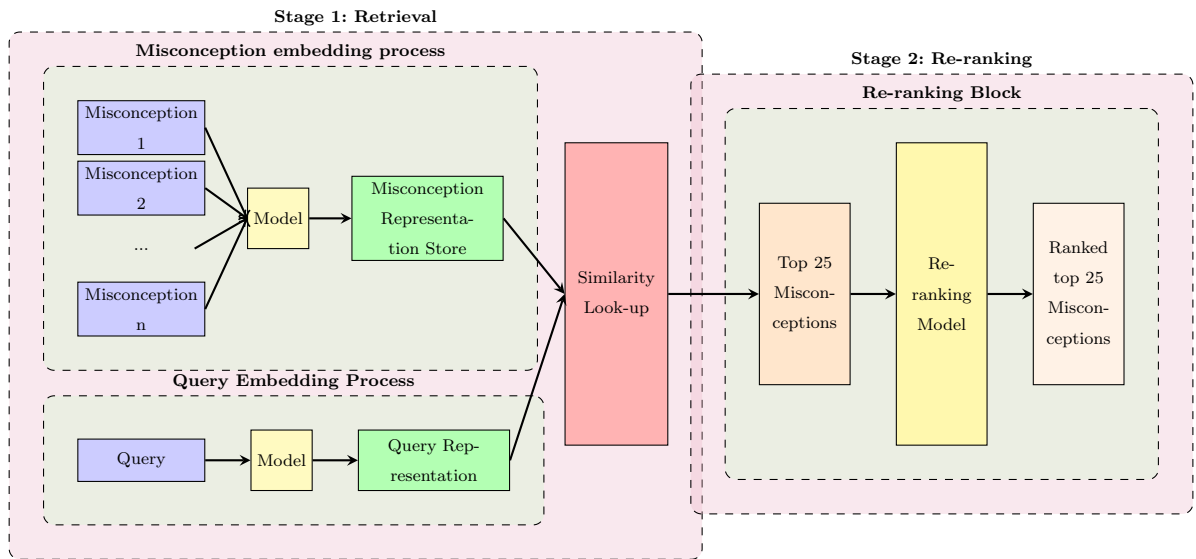
Trước tiên, sau tìm hiểu và tham khảo các giải pháp hiện có được sử dụng trong cuộc thi, nhóm em nhận thấy yêu cầu của cuộc thi là một bài toán về **Information Retrieval**. Đây là bài toán tập trung vào tìm kiếm và truy xuất các thông tin (**Document**) có liên quan đến câu truy vấn (**Query**) của người dùng, và trả về danh sách các thông tin đã được sắp xếp dựa trên tính liên quan.^[2]

Trong ngữ cảnh cụ thể của cuộc thi **Eedi**, bài toán này sẽ được phát biểu cụ thể như sau:

- **Query**: Là một câu truy vấn bao gồm các thông tin về chủ đề (**Subject**), cấu trúc chung (**Construct**), bài toán(**Question**), câu trả lời đúng (**Correct Answer**) và câu trả lời sai (**Wrong Answer**), đây là một metadata liên quan đến toàn bộ thông tin về một câu hỏi và một câu trả lời sai tương ứng với câu hỏi đó.
- **Document**: Là danh sách các mô tả về các Misconception được cung cấp.

Như vậy, mục tiêu của bài toán ở đây, là truy vấn được các mã Misconception có liên quan nhất đến câu trả lời sai được cung cấp trong truy vấn.

Để tiếp cận bài toán này, một pipeline phổ biến được dùng để tối ưu hóa hiệu suất truy xuất mà một pipeline gồm 2 giai đoạn: **Retrieval** và **Re-ranking**. Về mặt lý thuyết, một tập nhỏ các document ứng viên đầu tiên sẽ được truy xuất bởi một mô hình mang hiệu suất tính toán tốt, các ứng viên sau đó sẽ được chấm điểm lại bởi một mô hình lớn hơn, mạnh hơn, quy trình tổng quan được mô tả như sau:



Ở đây, quá trình cụ thể sẽ được mô tả như sau

– Giai đoạn 1: Retrieval :

Trong giai đoạn này, các mô tả về **Misconception**, cũng như **Query** chứa thông tin về câu trả lời sai, dưới dạng văn bản text, văn bản đó sẽ được định dạng phù hợp và được đưa vào một mô hình ngôn ngữ. Output thu được sẽ là các biểu diễn - representation, hay có thể hiểu là các vector embedding. Mỗi vector embedding là một biểu diễn số học giúp mô tả nội dung, ý nghĩa cho từng misconception hay từng query cụ thể.

Các embedding đó, sẽ được dùng trong quá trình so sánh độ tương đồng giữa:

- * Embedding cho câu hỏi, câu trả lời sai cũng các thông tin hỗ trợ (**Query**).
- * Embedding của tất cả các Misconception có thể xảy ra.

Mục tiêu là tìm ra top **25 misconceptions** có độ tương đồng cao nhất với **query**. Việc này được thực hiện bằng cách tính độ tương đồng giữa 2 vector, với độ đo nhóm dùng là **Cosine similarity**. Điều này cho phép tìm ra top 25 misconceptions có liên quan chặt chẽ đến câu trả lời sai.

- **Giai đoạn 2: Re-ranking:** Trong giai đoạn này, mục tiêu là sắp xếp lại thứ tự của top 25 misconceptions đã được chọn ở giai đoạn 1 để đưa ra thứ tự chính xác, hợp lý hơn, được thực hiện dưới sự hỗ trợ của một mô hình ngôn ngữ lớn hơn.

Điểm mấu chốt ở đây chính là sự cân bằng giữa mức độ hiệu quả và sự chính xác:

- * Nếu chỉ dùng giai đoạn 1, thì tốc độ truy vấn sẽ tăng, nhưng đổi lại độ chính xác sẽ bị ảnh hưởng do việc sử dụng một mô hình nhẹ hơn khiến nó không nắm bắt được các mối quan hệ ẩn sâu trong dữ liệu.
- * Ngược lại, nếu dùng trực tiếp giai đoạn 2, sắp xếp dựa trên tất cả các misconception có thể có, thì chi phí tính toán cực kì lớn, do độ phức tạp của mô hình và bài toán tăng cao.

Như vậy, việc tận dụng tốc độ của giai đoạn 1, cộng với độ chính xác cao của giai đoạn 2, với chi phí tính toán không quá cao khi chỉ phải xử lý một số lượng nhỏ các misconceptions, có thể đem lại hiệu quả tốt hơn

Bây giờ, chúng ta sẽ tìm hiểu kĩ hơn các kỹ thuật, cũng như quy trình cho từng giai đoạn.

4.2 Stage 1: Retrieval

4.2.1 Phân tích bài toán

Ở giai đoạn này, như đã mô tả ở trên, nhiệm vụ chính là tìm ra được top 25 misconceptions tiềm năng cho câu trả lời sai. Mục tiêu này được thực hiện bằng việc sử dụng "*embeddings*" đại diện cho các câu hỏi, cụ thể:

- **Giới thiệu:** Embedding là một kỹ thuật dùng để biểu diễn các đối tượng như từ ngữ, câu, hoặc văn bản dưới dạng các vector số học mà máy tính có thể hiểu và xử lý. Trong NLP, embedding giúp biểu diễn các từ hoặc câu trong một không gian đa chiều mà ở đó, các từ có ý nghĩa tương tự sẽ có vị trí gần nhau.
- **Mục tiêu:** Sử dụng embedding để biểu diễn các questions (bài toán) và misconceptions dưới dạng các vector trong không gian đa chiều, sau đó tính toán độ tương đồng giữa chúng để tìm ra những misconceptions gần nhất với mỗi câu hỏi.

4.2.2 Quá trình xây dựng mô hình

4.2.2.1 Chuẩn bị dữ liệu

Dữ liệu trong quá trình này đến từ 2 nguồn chính sau:

– **Dataset Misconceptions:**

Dữ liệu về các misconceptions được tải từ file "misconception.csv" với mỗi dòng là dữ liệu về nội dung các loại quan niệm sai lầm misconception.

– **Dataset Questions:**

Bộ dữ liệu câu hỏi bao gồm câu hỏi, nội dung câu trả lời đúng và cả câu trả lời sai, được tải từ file "test.csv".

Tiền xử lý dữ liệu:

Đầu tiên, nhóm nhận thấy rằng với cấu trúc hiện tại của dataframe dữ liệu, khi mà mỗi dòng đại diện cho một câu hỏi và tất cả 4 câu trả lời (1 đúng, 3 sai), với mục tiêu là tìm ra được các misconceptions cho từng câu trả lời sai của từng câu hỏi, việc xây dựng câu truy vấn cho từng cặp câu hỏi - câu trả lời sai sẽ rất khó khăn. Do đó, cần thực hiện các biến đổi về cấu trúc của dataframe dữ liệu:

- Tách các câu trả lời sai từ dữ liệu câu hỏi trắc nghiệm và chuyển đổi chúng sang dạng hàng để dễ dàng xử lý từng câu hỏi và đáp án sai riêng lẻ.
- Nếu có dữ liệu misconceptions (chỉ trong tập train), sẽ chuyển đổi các misconceptions thành dạng hàng để kết hợp với bộ dữ liệu câu hỏi.
- Loại bỏ các dòng không hợp lệ và chuẩn hóa dữ liệu để phù hợp với các bước phân tích và suy luận tiếp theo.

Tạo Query với Template : Template được sử dụng để định dạng dữ liệu đầu vào các mô hình ngôn ngữ để mô hình dễ dàng hiểu và xử lý dữ liệu. Template sẽ bao gồm các features chính sau:

- **Question:** Nội dung câu hỏi
- **Subject:** Tên môn học liên quan đến câu hỏi
- **Construct:** Nội dung hướng dẫn cấu trúc thực hiện để trả lời câu hỏi
- **Correct Answer:** Nội dung câu trả lời đúng
- **Wrong Answer:** Nội dung câu trả lời sai

Mỗi dòng dữ liệu, sau khi được apply vào template, sẽ là một văn bản tổng hợp thông tin của câu hỏi và các câu trả lời sai, sẵn sàng để thực hiện quá trình embedding.

4.2.2.2 Biểu diễn dữ liệu bằng embedding

Mã hóa văn bản (Tokenization): Tokenizer (AutoTokenizer) mã hóa các questions và misconceptions thành các token số (tensors) phù hợp với mô hình.

Mô hình ngôn ngữ lớn (LLM): Mô hình ngôn ngữ lớn (LLM) được nhóm sử dụng là Qwen-2.5 14B. Đây là một mô hình ngôn ngữ lớn được phát triển bởi Alibaba Group, xây dựng dựa trên kiến trúc transformer cho phép mô hình hiểu sâu hơn về ngữ cảnh và mối quan hệ giữa các từ trong câu. Nhóm sử dụng mô hình này để tạo ra các vector biểu diễn (embedding) cho câu hỏi và các misconceptions trong dữ liệu.

Kỹ thuật LoRA: Sử dụng kỹ thuật tinh chỉnh LoRA để giảm số lượng tham số cần cập nhật khi huấn luyện mô hình, tiết kiệm tài nguyên tính toán và tối ưu hóa hiệu suất mô hình trên các thiết bị CPU bị giới hạn.

Tính toán embedding:

- Dữ liệu văn bản được xử lý theo từng batch để giảm thiểu sử dụng bộ nhớ. Các vector embedding cho mỗi batch được tính toán và chuẩn hóa (normalized) để đảm bảo tổng giá trị của các phần tử trong vector bằng 1, giúp tăng độ chính xác khi tính toán độ tương đồng.
- Để lấy embedding đại diện nhất cho mỗi câu hỏi và misconception, nhóm sẽ lấy token cuối cùng từ các `last_hidden_state` của mô hình.

4.2.2.3 Tính toán độ tương đồng

Thuật toán Nearest Neighbors: Sử dụng thuật toán Nearest Neighbors với khoảng cách Cosine để tìm kiếm các misconceptions gần nhất với mỗi question. Tính khoảng cách Cosine giữa embedding của mỗi question với tất cả các embedding của misconceptions.

Truy xuất Top 25 misconceptions: Sau khi tính toán độ tương đồng sẽ trả về top 25 misconceptions gần nhất với mỗi câu trả lời sai.

Như vậy, kết thúc bước 1, sử dụng mô hình Qwen 2.5 14B, ta đã có được 25 misconceptions tiềm năng, để tiếp tục tiến vào giai đoạn 2.

4.3 Stage 2: Re-ranking

Ở phần này, nhóm sẽ tiếp tục tìm hiểu chi tiết quá trình thực hiện stage 2: Sắp xếp lại thứ tự của top 25 misconceptions.

Ở đây, việc sắp xếp này sẽ được thực hiện dựa trên ý tưởng về prompt engineering, cung cấp thông tin về câu trả lời cùng các misconceptions tiềm năng và yêu cầu mô hình ngôn ngữ lớn trả về response về misconception tốt nhất. Bằng cách phân tích response đó, thứ tự mới của các misconceptions có thể được xác định.

Quá trình trên được thực hiện dựa trên 2 yếu tố chính: Mô hình Qwen 2.5 32B và bộ xử lý Multiple Choice Logits.

4.3.1 Mô hình Qwen 2.5 32B

Cũng tương tự như bước 1, ở bước này, nhóm em cũng sử dụng mô hình Qwen 2.5, nhưng với số lượng tham số là 32 tỉ, mô hình Qwen 2.5 32B, với số lượng tham số lớn hơn, đem lại độ chính xác tốt hơn mô hình Qwen 2.5 14B trên đa số các benchmark được sử dụng.[\[6\]](#)

Quá trình truy vấn được thực hiện với prompt được tạo dựa trên một template như sau:

```
prompt = "You are an elite mathematics teacher tasked to assess the student's  
understanding of math concepts. Below, you will be presented with: the math question,  
the correct answer, the wrong answer and {k} possible misconceptions that could have  
led to the mistake.
```

```
{question_text}
```

```
Possible Misconceptions:{choices}
```

```
Select one misconception that leads to incorrect answer. Just output a single number  
of your choice and nothing else.
```

```
Answer: "
```


Với các tham số:

- ***k***: Số lượng các misconceptions cần được sắp xếp lại, ở đây, nhóm chúng em sẽ sắp xếp top - 9 các misconceptions đầu tiên, giá trị của ***k*** = **9**
- ***question_text***: Là tổ hợp các thông tin của của một câu trả lời sai, bao gồm các trường đã được sử dụng trong stage 1 như: chủ đề (Subject), cấu trúc (Construct), bài toán (Question), đáp án chính xác (Correct Answer) và câu trả lời sai (Wrong Answer)
- ***choices***: Là tổ hợp top ***k*** các misconception, bao gồm thông tin về mô tả của nó được cung cấp trong file *misconception_mapping.csv*

Với prompt như trên, nhóm mong muốn mô hình Qwen 2.5 sẽ trả về một misconception, đúng nhất để mô tả câu trả lời sai đó.

4.3.2 Bộ xử lý MultipleChoiceLogits

Nhưng làm sao để đảm bảo rằng mô hình Qwen 2.5 sẽ luôn trả về duy nhất một misconception đúng nhất mỗi câu query? Xuất phát từ cơ chế xử lý **logits** của mô hình ngôn ngữ.

1. Cơ chế xử lý logits:

- Trong các mô hình ngôn ngữ, từ tiếp theo được dự đoán dựa trên **phân phối xác suất** của các từ có thể xuất hiện.
- Từ có xác suất cao nhất sẽ được chọn làm đầu ra.
- Để hỗ trợ đa dạng các tác vụ, NVIDIA đã cung cấp một thư viện **Logit Processor Zoo**, cung cấp các cơ chế xử lý logit, phù hợp với từng tác vụ cụ thể.^[5]

2. Bộ xử lý Mutiple Choice Logits:

Trong số các bộ xử lý logit được hỗ trợ, bộ xử lý **MultipleChoiceLogits** trở nên vô cùng phù hợp với yêu cầu re-ranking ở bước 2, khi mà bộ xử lý này hỗ trợ hướng dẫn mô hình trả lời các câu hỏi cần tuân thủ định dạng câu trúc, như câu đó, và đặc biệt những câu hỏi ở dạng trắc nghiệm.

Trong ngữ cảnh ở bước 2, câu hỏi ở đây chính là phần mô tả về câu trả lời sai, còn các lựa chọn trắc nghiệm lần lượt là các misconception nằm trong top *k* đã được chọn lọc. **Multi-**

pleChoiceLogits sẽ trả về một phân phối xác suất khả năng xuất hiện của từng lựa chọn misconception trong top k.

Cũng chính dựa vào phân phối xác suất đó, nhóm cũng có thể dễ dàng decode và sắp xếp lại thứ tự các misconception, dựa trên xác suất của nó trong phân phối, nên mặc dù prompt yêu cầu chỉ trả về misconception tốt nhất, nhưng việc sắp xếp thứ tự vẫn khả thi, vì mô hình không chỉ chọn được misconception phù hợp nhất mà còn cho phép nhóm trích xuất toàn bộ phân phối để sắp xếp các misconceptions khác một cách hiệu quả.

Bên cạnh đó, để tối ưu tốc độ và bộ nhớ cho quá trình truy xuất, nhóm giới hạn số lượng token trả về chỉ là 1, đó chính là lí do mà nhóm em giới hạn số lượng misconception được sắp xếp là 9, khi mà, token được trả về này sẽ đại diện cho chỉ số của misconception được chọn, từ 1 đến 9

Như vậy, kết thúc bước 2, **9 misconceptions** đầu tiên trong số 25 misconceptions được suy luận ở bước 1, sẽ được sắp xếp lại thứ tự, sử dụng mô hình Qwen 2.5 32B. Từ đó, tạo thành một danh sách các misconception mới, với độ chính xác được nâng cao.

5 Thử nghiệm

5.1 Các cách tiếp cận trước của nhóm

Trước khi tìm hiểu và phát hiện cách giải quyết bao gồm 2 giai đoạn như hiện tại, nhóm em cũng có thử nghiệm một số hướng giải quyết như:

- **Mô hình supervised learning:**

Với embedding của câu trả lời làm input, nhóm em cố huấn luyện một mô hình có thể phân loại các misconception dựa trên nhãn là các misconception id.

Và kết quả thật sự tệ, xuất phát từ việc không tận dụng được mô tả của các misconception, cộng với việc thiếu dữ liệu, mất cân bằng giữa các misconception với nhau.

- **Độ tương đồng + Vectorize:**

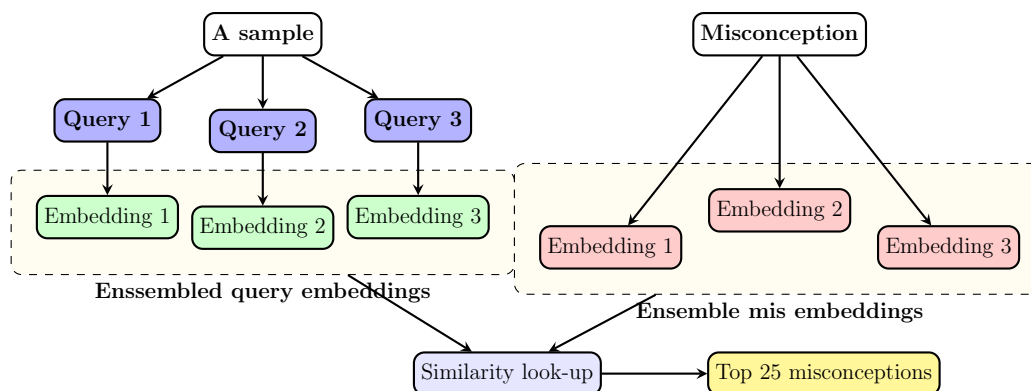
Đây là ý tưởng tiền thân của bước 1, khi mà nhóm cũng sử dụng độ tương đồng Cosine để đưa ra top 25, nhưng các embeddings được tạo ra từ các chỉ số thống kê như TF-IDF, hoặc các mô hình ngôn ngữ có số lượng tham số tương đối nhỏ như Llama, Bert

Kết quả trên hướng tiếp cận này cũng đã có sự cải thiện.

5.2 Kết hợp nhiều embedding khác nhau

Xuất phát từ một ý tưởng của một người tham gia cuộc thi khác: Evan Arlian [1], nhóm nhận ra rằng việc kết hợp các embeddings, dựa trên các cấu trúc query khác nhau, cũng có thể độ chính xác của top 25 các misconceptions đầu tiên, từ đó gián tiếp cải thiện độ chính xác của danh sách misconception cuối cùng. Do đó, nhóm em đã kết hợp các embedding của mô hình Qwen 2.5 14B, của cùng một trả lời sai, trên 3 định dạng query khác nhau, cũng như chỉnh sửa một số thông số của kỹ thuật Lora, để tạo ra những phiên bản embedding của cùng một sample trong dữ liệu hay cùng một misconception.

Về cấu trúc cơ bản có thể được mô tả như sau:



Như vậy, từ việc tạo embedding cho query và misconception bằng cách tổng hợp embeddings đến từ nhiều template và config mô hình khác nhau, kết quả thu được sẽ là:

	Public Score	Private Score
Before ensemble	0.49952	0.44615
After ensemble	0.51298	0.48434

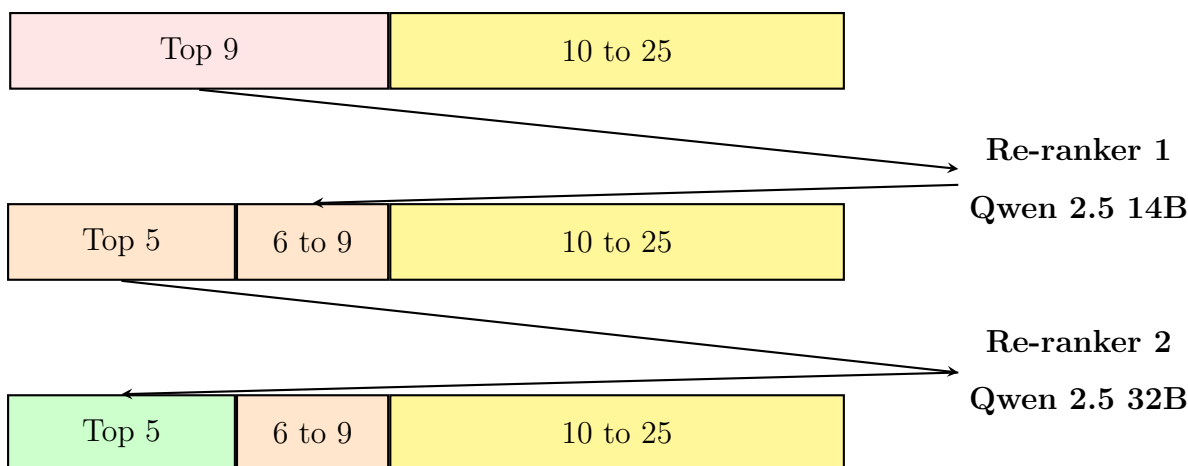
Về notebook , có thể xem [Notebook tại đây](#)

5.3 Kết hợp quá trình re-rank nhiều lần

Xuất phát từ ý tưởng của người chiến thắng cuộc thi Raja Biswas[3], quá trình re-rank của tác giả được thực hiện trải qua 3 giai đoạn:

- **14B ranker**: Dùng mô hình Qwen 2.5 14B để sắp xếp lại thứ tự tất cả các Misconception.
- **32B ranker**: Mô hình Qwen 2.5 32B để sắp xếp lại top 8 misconceptions.
- **72B ranker**: Mô hình Qwen 2.5 72B để sắp xếp lại top 5 misconceptions.

Nhóm em đã cố mô phỏng lại ý tưởng đó, nhưng điểm khó khăn ở đây là mô hình Qwen 2.5 72B là một mô hình quá lớn, và cấu hình GPU cơ bản của Kaggle không đủ để có thể hoàn thành quá trình sắp xếp (hoặc đủ nhưng nhóm em chưa tìm ra được cách tối ưu bộ nhớ), nên để đơn giản hóa nó, thì quá trình re-rank nhóm em thử nghiệm sẽ chỉ gồm 2 giai đoạn thôi:



- **14B ranker:** Mô hình Qwen 2.5 14B sắp xếp cho top 9 misconceptions.
- **32B ranker:** Mô hình Qwen 2.5 32B sắp xếp lại top 5 misconceptions.

Và kết quả sẽ là:

	Public Score	Private Score
1-step re-ranking	0.49952	0.44615
2-steps re-ranking	0.48829	0.46444

Như có thể thấy, thử nghiệm của nhóm em, đem lại kết quả không được cải thiện , mà còn có sự sụt giảm điểm.

Về nguyên nhân, nhóm thấy do cách tiếp cận chưa phù hợp khi lần sắp xếp đầu tiên dùng mô hình Qwen 2.5 14B, dẫn đến có thể đưa ra thứ tự không chính xác, hoặc sai lệch nhiều, làm ảnh hưởng đến việc sắp xếp ở lần thứ 2 với mô hình Qwen 2.5 32B.

Về notebook tham khảo, cũng có thể xem [tại đây](#)

5.4 Fine tune mô hình ngôn ngữ

Dựa trên tìm hiểu và tham khảo những người tham gia cuộc thi khác, nhóm nhận thấy một cách để nâng cao độ chính xác cho dự đoán của các mô hình ngôn ngữ chính là fine tune 1 phần mô hình đó, trên bộ dữ liệu train được cung cấp.

Tuy nhiên, mặc dù đã thử nghiệm fine tune thành công mô hình ngôn ngữ **BGE**, nhưng khi áp dụng cho mô hình Qwen 2.5 14B, xuất hiện vấn đề về bộ nhớ GPU được cung cấp bởi

Kaggle không đủ để thực hiện quá trình fine tune. Mặc dù đã thử nhiều cách nhưng hiện tại nhóm vẫn chưa thành công ở thử nghiệm này.

5.5 Data Augmented

1. Mục tiêu

Mục tiêu chính của thử nghiệm là chuẩn bị một tập dữ liệu phong phú và cân bằng nhằm hỗ trợ quá trình **fine-tune mô hình**, giúp nâng cao hiệu suất và khả năng tổng quát hóa. Trong tập huấn luyện ban đầu, một số **misconceptions** xuất hiện rất ít hoặc thậm chí chưa từng xuất hiện, gây khó khăn cho mô hình trong việc nhận diện và phân loại các nhóm **misconceptions** ít phổ biến hoặc hoàn toàn mới.

Thử nghiệm này tập trung vào việc bổ sung dữ liệu cho các **misconceptions** hiếm và tạo dữ liệu mới cho các **misconceptions** chưa có trong tập huấn luyện. Đồng thời, thử nghiệm hướng đến làm giàu và chi tiết hóa dữ liệu, giúp tăng giá trị ngữ nghĩa và cung cấp thông tin rõ ràng hơn để mô hình có thể học sâu và hiểu rõ hơn về các **misconceptions**.

Kết quả mong đợi là tạo ra một tập dữ liệu đủ phong phú để **fine-tune mô hình** cho các tác vụ tiếp theo, giúp mô hình cải thiện độ chính xác, khả năng nhận diện và phân loại các **misconceptions** khó. Việc cân bằng và chi tiết hóa dữ liệu đảm bảo rằng mô hình có thể đạt hiệu suất cao hơn khi được áp dụng vào các bài toán phức tạp, đồng thời hỗ trợ tốt hơn cho các nhiệm vụ dự đoán và phân tích đa dạng trong tương lai.

2. Ý tưởng cơ bản

- Để giải quyết vấn đề này, nhóm đã khảo sát và tiến hành tìm hiểu với ba phương pháp tiếp cận khác nhau, với mục tiêu tạo ra một tập dữ liệu phong phú, cân bằng và hỗ trợ hiệu quả cho mô hình học máy.

(a) Sử dụng LLM tạo phân tích chuỗi tư duy (Chain of Thought - CoT)

- * Phương pháp này tận dụng Mô hình Ngôn ngữ Lớn (LLM) để sinh ra chuỗi tư duy (Chain of Thought - CoT), nhằm phân tích từng bước cách giải quyết một bài toán cụ thể. Chuỗi tư duy này được thiết kế để làm rõ logic của quá trình giải bài, chỉ ra các lỗi sai tiềm năng, và giải thích chi tiết lý do dẫn đến

những sai lầm đó.

- * Trong quá trình xử lý, LLM được cung cấp các câu hỏi từ tập dữ liệu huấn luyện kèm theo đáp án đúng. Từ đó, mô hình tạo ra một lời giải thích ngắn gọn nhưng đầy đủ về cách giải bài. Phân tích này giúp mô hình hiểu được logic của đáp án đúng, đồng thời xây dựng một nền tảng để so sánh với các đáp án sai.
- * Đối với các đáp án sai, LLM được yêu cầu phân tích lỗi từng bước để xác định misconceptions của người học. Prompt được thiết kế để khuyến khích mô hình mô phỏng quá trình tư duy tự nhiên, đi từ câu hỏi, qua lập luận sai lầm, đến việc nhận diện và giải thích misconceptions một cách chi tiết. Các kết quả này được lưu trữ trong các cột dữ liệu mới, bổ sung thêm ngữ nghĩa và bối cảnh cho tập dữ liệu.
- * Việc thêm chuỗi tư duy này giúp tập dữ liệu không chỉ cân bằng hơn mà còn phong phú hơn, với thông tin ngữ nghĩa chi tiết về cả cách giải bài đúng và các sai lầm tiềm năng. Điều này đặc biệt hữu ích trong việc cải thiện khả năng của mô hình khi phải đối mặt với các misconceptions hiếm hoặc phức tạp.
- * Ngoài ra, phương pháp này còn tạo cơ sở cho các bước fine-tune mô hình, giúp nâng cao hiệu suất trên các bài toán yêu cầu khả năng suy luận logic và phân tích sâu hơn.
- * Link notebook: [Tại đây](#).
- * Link dataset: [Tại đây](#).

(b) Mở rộng nội dung chi tiết cho MisconceptionName

- * Phương pháp này tập trung vào việc sử dụng Mô hình Ngôn ngữ Lớn (LLM) để phân tích và mở rộng thông tin cho mỗi misconception. Trong dữ liệu gốc, các misconceptions thường chỉ được mô tả bằng một câu ngắn gọn, dẫn đến khó khăn cho mô hình trong việc hiểu rõ bối cảnh và bản chất của chúng. Bằng cách yêu cầu LLM giải thích chi tiết, tập dữ liệu trở nên phong phú và đầy đủ ngữ nghĩa hơn.
- * Quy trình thực hiện bắt đầu bằng việc cung cấp cho LLM thông tin về câu hỏi, đáp án đúng, và đáp án sai từ tập dữ liệu gốc. LLM được thiết kế với

prompt đặc biệt để:

- Phân tích các đáp án sai, chỉ ra lý do tại sao chúng sai.
 - Liên kết đáp án sai với các misconceptions tương ứng.
 - Sinh các giải thích chi tiết cho misconceptions, bao gồm nguyên nhân xảy ra lỗi và các ví dụ minh họa thực tế mà lỗi này thường gặp.
- * Các giải thích được lưu vào cột dữ liệu mới, ví dụ như ‘FullResponse’. Cột này đóng vai trò như một lớp thông tin bổ sung, không chỉ giúp mô hình phân loại misconceptions mà còn cung cấp dữ liệu phong phú hơn để mô hình học tập.
 - * Một điểm đáng chú ý là LLM được yêu cầu tập trung hoàn toàn vào phân tích misconceptions mà không cung cấp cách giải bài đúng. Điều này đảm bảo rằng mô hình tập trung vào việc hiểu lỗi sai thay vì sao chép cách giải đúng, từ đó tăng cường khả năng nhận diện misconceptions.
 - * Ngoài ra, phương pháp này còn hỗ trợ cải thiện embedding của misconceptions. Với các giải thích chi tiết hơn, mỗi misconception trở nên rõ ràng và có giá trị ngữ nghĩa lớn hơn, giúp mô hình học sâu hơn về sự liên kết giữa câu hỏi, đáp án, và misconceptions.
 - * Phương pháp này đặc biệt hiệu quả trong việc hỗ trợ fine-tuning mô hình, tạo nền tảng vững chắc cho các tác vụ phức tạp như phân loại lỗi hoặc dự đoán misconceptions hiếm gặp. Tập dữ liệu sau khi mở rộng không chỉ cân bằng hơn mà còn giúp mô hình có khả năng tổng quát hóa tốt hơn.
 - * Link notebook: [Tại đây](#).
 - * Link dataset: [Tại đây](#).

(c) **Tận dụng dữ liệu lịch sử và sinh dữ liệu nhân tạo**

- * Phương pháp này lấy cảm hứng từ giải pháp đạt hạng 2 tại cuộc thi Eedi NeurIPS 2022, với ý tưởng chính là tận dụng dữ liệu lịch sử và sinh dữ liệu nhân tạo để giải quyết vấn đề mất cân bằng và làm giàu tập dữ liệu huấn luyện.
- * Tận dụng dữ liệu lịch sử:
 - Nhóm sử dụng file ‘Subject Metadata’ từ cuộc thi Eedi NeurIPS 2022, chứa thông tin về các môn học cha (parent subject).

- Các thông tin này được tích hợp vào cả ‘train.csv’ và ‘test.csv’, giúp bổ sung ngữ cảnh đầy đủ cho các câu hỏi. Điều này không chỉ cải thiện độ chính xác mà còn tăng khả năng phân tích sâu của mô hình khi tiếp cận với các câu hỏi có bối cảnh phức tạp.
- * Sinh dữ liệu nhân tạo qua ba thế hệ:
 - Generation 1: Sử dụng một misconception và một số ví dụ ít ỏi (few-shot examples) được lấy ngẫu nhiên từ ‘train.csv’ để tạo câu hỏi và đáp án.
 - Generation 2: Mở rộng ngữ cảnh của Generation 1 bằng cách lấy các câu hỏi có cùng misconception từ ‘train.csv’ và Generation 1.
 - Generation 3: Tích hợp câu hỏi từ cả ‘train.csv’, Generation 1, và Generation 2, đồng thời cải tiến prompt dựa trên các bài viết kỹ thuật, tạo ra dữ liệu có chất lượng và tính đa dạng cao hơn.
- * Mở rộng giải thích misconceptions:
 - Các misconceptions ban đầu chỉ là các mô tả ngắn gọn. Phương pháp này sử dụng LLM để sinh các giải thích chi tiết, bao gồm nguyên nhân và ví dụ thực tế mà misconception có thể xảy ra.
 - Các giải thích này không chỉ giúp cải thiện embedding misconceptions mà còn hỗ trợ mô hình học tập và nhận diện lỗi một cách hiệu quả hơn.
- * Chuỗi tư duy (Chain of Thought):
 - Nhóm sử dụng mô hình ‘qwen2.5-32B-Instruct-AWQ’ để sinh chuỗi tư duy (Chain of Thought - CoT).
 - CoT giúp phân tích từng bước lý do dẫn đến misconceptions, từ đó cung cấp thêm thông tin giá trị cho nhiệm vụ truy xuất và xếp hạng dữ liệu.
- * Link github: [Tại đây](#).
- * Link dataset: [Tại đây](#).

3. Phân tích phương pháp

Trong ba phương pháp trên, nhóm quyết định tập trung thực nghiệm sâu vào **Phương pháp 3** vì khả năng cải thiện ngữ cảnh qua Subject Metadata, tạo dữ liệu phong phú qua ba thế hệ sử dụng LLM, và mở rộng giải thích chi tiết cho misconcep-

tions. Phương pháp này được kỳ vọng giải quyết mất cân bằng dữ liệu, đồng thời tăng tính đa dạng và hiệu quả cho mô hình.

– Ý tưởng chính

Phương pháp này tập trung vào việc tận dụng dữ liệu lịch sử từ cuộc thi Eedi NeurIPS 2022 và sinh thêm dữ liệu nhân tạo chất lượng cao để giải quyết vấn đề mất cân bằng dữ liệu và thiếu thông tin về một số misconceptions. Ý tưởng chính được triển khai qua ba thành phần chính:

- * Tận dụng Subject Metadata để bổ sung ngữ cảnh và tăng độ chính xác.

Subject Metadata từ cuộc thi trước chứa thông tin chi tiết về parent subject (môn học cha), cho phép bổ sung ngữ cảnh cho từng câu hỏi trong tập dữ liệu. Việc thêm thông tin này giúp mô hình không chỉ hiểu rõ từng câu hỏi mà còn nắm bắt được mối liên hệ rộng hơn giữa các chủ đề và misconceptions. Điều này tạo ra một bức tranh tổng quan hơn về tập dữ liệu, hỗ trợ mô hình trong quá trình huấn luyện.

- * Sinh dữ liệu qua ba thế hệ (Generation1, Generation2, Generation3).
- * Mở rộng giải thích cho **misconceptions** để cải thiện embedding.

– Quy trình tạo dữ liệu qua ba thế hệ

- * **Generation1:** Sử dụng few-shot learning để tạo câu hỏi nhân tạo.

Dựa trên các misconceptions từ tập train, sử dụng LLM với kỹ thuật few-shot learning để tạo câu hỏi nhân tạo.

- * **Generation2:** Mở rộng câu hỏi từ Generation1 dựa trên ngữ cảnh gần.

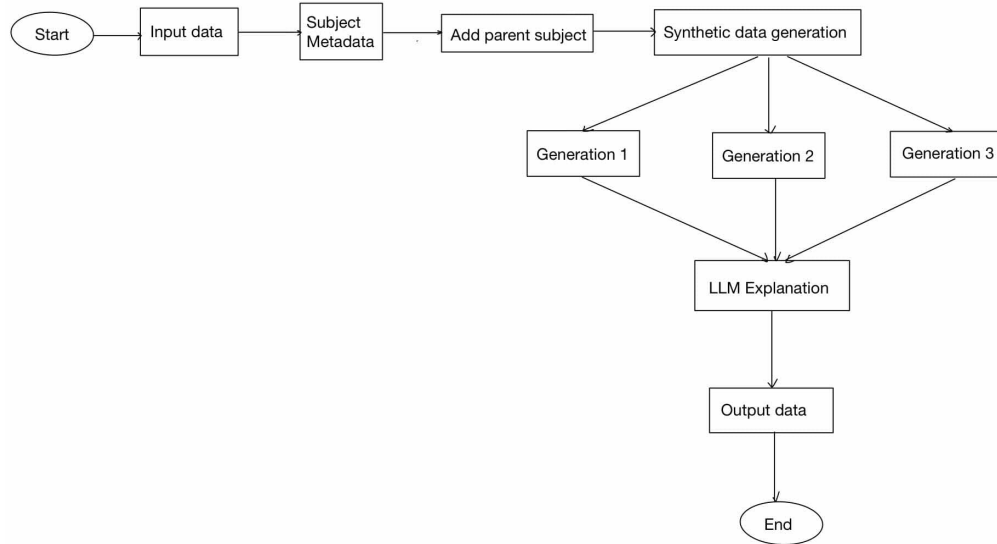
Tiếp tục mở rộng từ Generation1 bằng cách lấy các câu hỏi có cùng misconception trong tập train và Generation1 làm ví dụ.

Phương pháp này tạo ra các câu hỏi phong phú hơn, có ngữ cảnh sát hơn với misconceptions.

- * **Generation3:** Kết hợp ngẫu nhiên các câu hỏi để tăng tính đa dạng và chất lượng.

Cải thiện prompt cho LLM dựa trên các kỹ thuật từ các bài viết chuyên sâu

Kết hợp ngẫu nhiên hai câu hỏi từ mỗi nguồn (train, Generation1, Generation2) để tạo ra các câu hỏi mới với tính đa dạng và chất lượng cao hơn.



Hình 51: Tóm tắt quá trình tạo dữ liệu tổng hợp

– Mở rộng giải thích misconceptions

- * Các misconceptions trong tập dữ liệu thường được biểu diễn bằng một câu ngắn, dẫn đến việc embedding không đủ ý nghĩa. Phương pháp này sử dụng LLM để sinh thêm các giải thích chi tiết cho từng misconception, bao gồm nguyên nhân tại sao misconception xảy ra và các ví dụ điển hình mà lỗi này thường xuất hiện
- * Việc mở rộng giải thích giúp cải thiện embedding của misconceptions, cung cấp thêm ngữ nghĩa quan trọng cho mô hình trong quá trình học và dự đoán.

4. Kết quả

– Tình trạng chạy thử nghiệm

- * Code chạy thành công nhưng thời gian xử lý lâu do tính toán phức tạp.

Thời gian chạy lâu: Quy trình sinh dữ liệu đòi hỏi tính toán phức tạp và tài nguyên lớn, đặc biệt khi sử dụng LLM để sinh dữ liệu qua nhiều giai đoạn.

Hiện tại, chỉ có thể demo kết quả từ một nhánh nhỏ trong Generation1, chưa bao quát được toàn bộ dữ liệu từ các giai đoạn khác.

– Demo kết quả

- * Vì thời gian chạy quá lâu và GPU sử dụng bị hạn chế nên nhóm chỉ demo một phần dữ liệu mới được tạo ra (chưa phải bước cuối cùng).
- * Đoạn code thực hiện quá trình sinh dữ liệu chi tiết để giải thích misconceptions

bằng cách sử dụng mô hình LLM (Qwen2.5-32B-Instruct-AWQ). File đầu vào chứa thông tin về câu hỏi, đáp án đúng, đáp án sai, và các chủ đề liên quan. Đầu ra là các giải thích chi tiết về misconceptions, được sinh ra từ prompt thiết kế riêng cho giáo viên toán.

* Quy trình bao gồm:

Đọc dữ liệu từ file train.csv, chuẩn bị thông tin như QuestionText, CorrectAnswerText, AnswerText và các chủ đề liên quan (FirstSubjectName, SecondSubjectName, ThirdSubjectName).

Tạo prompt với hai phần chính: System Prompt (Cung cấp ngữ cảnh về môn học và chủ đề) và User Prompt (Truyền tải nội dung câu hỏi, đáp án đúng, và đáp án sai của học sinh)

Sinh dữ liệu với LLM

Lưu kết quả vào file CSV mới, với cột bổ sung chứa nội dung giải thích misconceptions.


* Link tập dữ liệu: [Tại đây](#).

* Link notebook: [Tại đây](#).

6 Tổng kết

6.1 Kết quả trên Kaggle

Kết quả điểm trên Kaggle đạt được là:

Submission and Description	Private Score ⓘ	Public Score ⓘ	Selected
 Model-final - Version 1 Succeeded (after deadline) · Huynh Cao Khoi...	0.44615	0.49952	<input type="checkbox"/>

Hình 52: Điểm tổng kết của nhóm trên Kaggle

Notebook trên Kaggle có thể được xem [tại đây](#)

6.2 Tự nhận xét

Ưu điểm:

- Mô hình sử dụng embedding giúp biểu diễn ngữ nghĩa của câu hỏi và misconceptions chính xác hơn, hỗ trợ việc suy luận tốt hơn.
- Kết hợp mô hình ngôn ngữ lớn Qwen-2.5 14B với LoRA, tối ưu hóa hiệu suất mà không cần huấn luyện toàn bộ mô hình.

Hạn chế:

- Mặc dù sử dụng LoRA để tối ưu hóa, việc tính toán embedding cho các đoạn văn bản lớn vẫn còn chậm,
- Việc sử dụng thuật toán tìm kiếm nearest neighbors có thể chưa hoàn toàn chính xác trong tất cả các trường hợp.

6.3 Hướng đi trong tương lai

Để tiếp tục phát triển và cải thiện mô hình, các hướng đi tiếp theo có thể bao gồm:

- Thu thập thêm dữ liệu về các misconceptions đa dạng và phong phú hơn để cải thiện độ chính xác của mô hình.
- Tối ưu hóa thêm việc fine-tune mô hình trên tập dữ liệu cụ thể để cải thiện khả năng hiểu và trích xuất thông tin.
- Sử dụng các thuật toán tối ưu hơn để cải thiện tốc độ tìm kiếm.

Tài liệu

- [1] Eedi - Mining Misconceptions in Mathematics — kaggle.com. <https://www.kaggle.com/competitions/eedi-mining-misconceptions-in-mathematics/discussion/551659>. [Accessed 19-01-2025].
- [2] What is information retrieval? | IBM — ibm.com. <https://www.ibm.com/think/topics/information-retrieval>. [Accessed 19-01-2025].
- [3] Raja Biswas. Mining misconceptions in mathematics, 1st place detailed solution. <https://www.kaggle.com/competitions/eedi-mining-misconceptions-in-mathematics/discussion/551688>, 2024. Accessed: 18 January 2025.
- [4] Hong Cheng, Xifeng Yan, and Jiawei Han. Mining graph patterns. *Frequent pattern mining*, pages 307–338, 2014.
- [5] Hugging Face. Controlling language model generation with nvidia’s logitsprocessor-zoo. <https://huggingface.co/blog/logits-processor-zoo>, n.d. Accessed: 18 January 2025.
- [6] Q. Team. Qwen2.5-llm: Extending the boundary of llms. <https://qwenlm.github.io/blog/qwen2.5-llm/>, September 2024. Accessed: 2025-01-18.