Feedback literacy: Holistic analysis of secondary educators' views of LLM explanations of program error messages in the classroom

Raspberry Pi
Foundation

Definitions

- Large Language Model (LLM) = “complex mathematical representation of language that is based on very large amounts of data and allows computers to produce language that seems similar to what a human might say" (Cambridge University Press, 2024, ‘LLM’ definition)
- Programming Error Messages (PEMs) = diagnostic messages generated by compilers or interpreters when the code violates the specifications of the programming language (Leinonen et al., 2022)

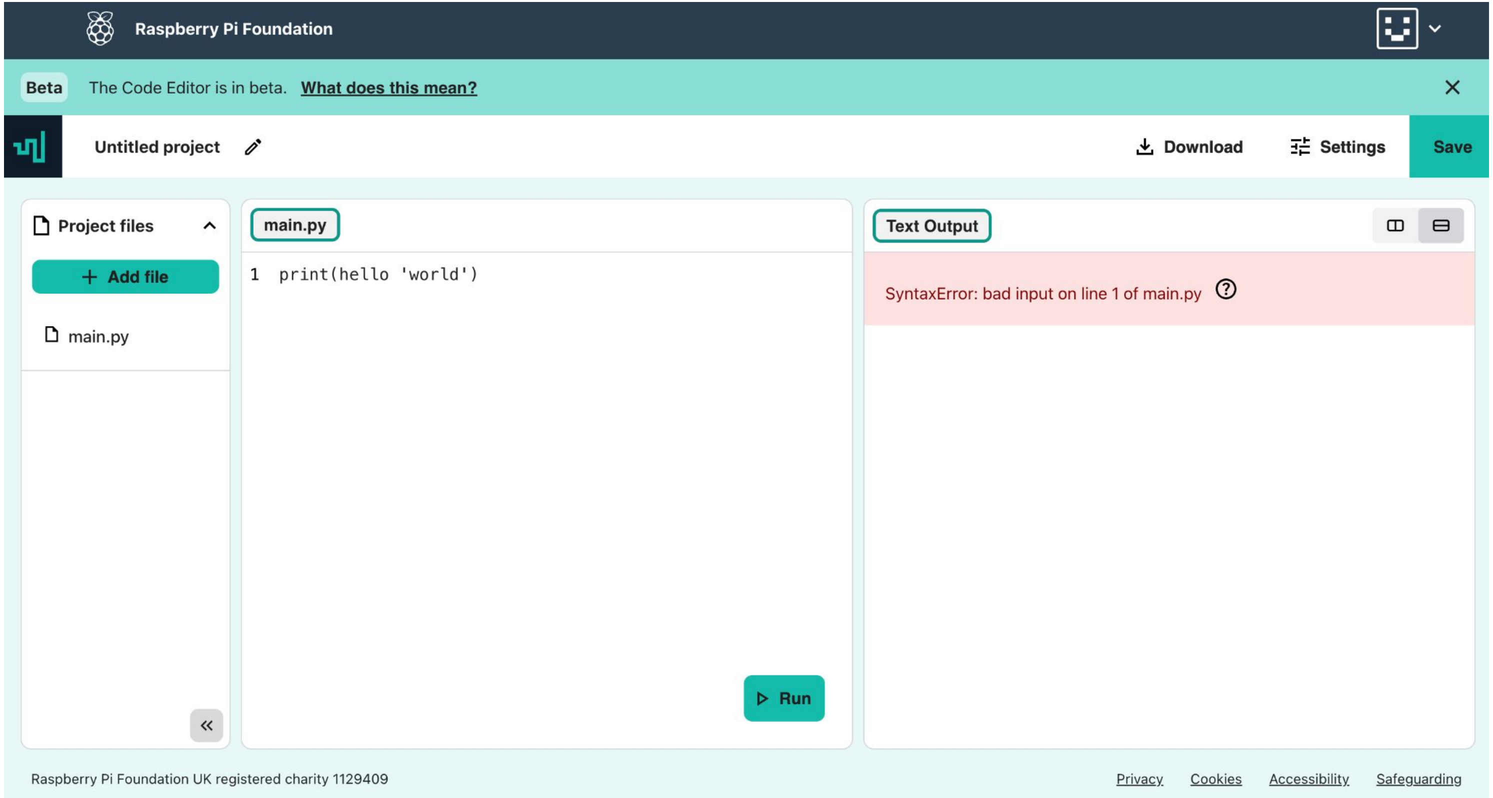
Background

- PEMs difficult to decipher (Denny et al., 2020)
- Using LLMs as a way of enhancing PEMs (Leinonen et al., 2022)
- Conflicting evidence on effective enhanced PEMs (Becker et al., 2019)
- Lack of teachers' voice

RQs

- RQ1: What are secondary educators’ views on the potential classroom use of LLM program error message explanations?
- RQ2: In what ways can a feedback literacy perspective support the analysis of educators’ views of potential classroom use of LLM program error message explanations?

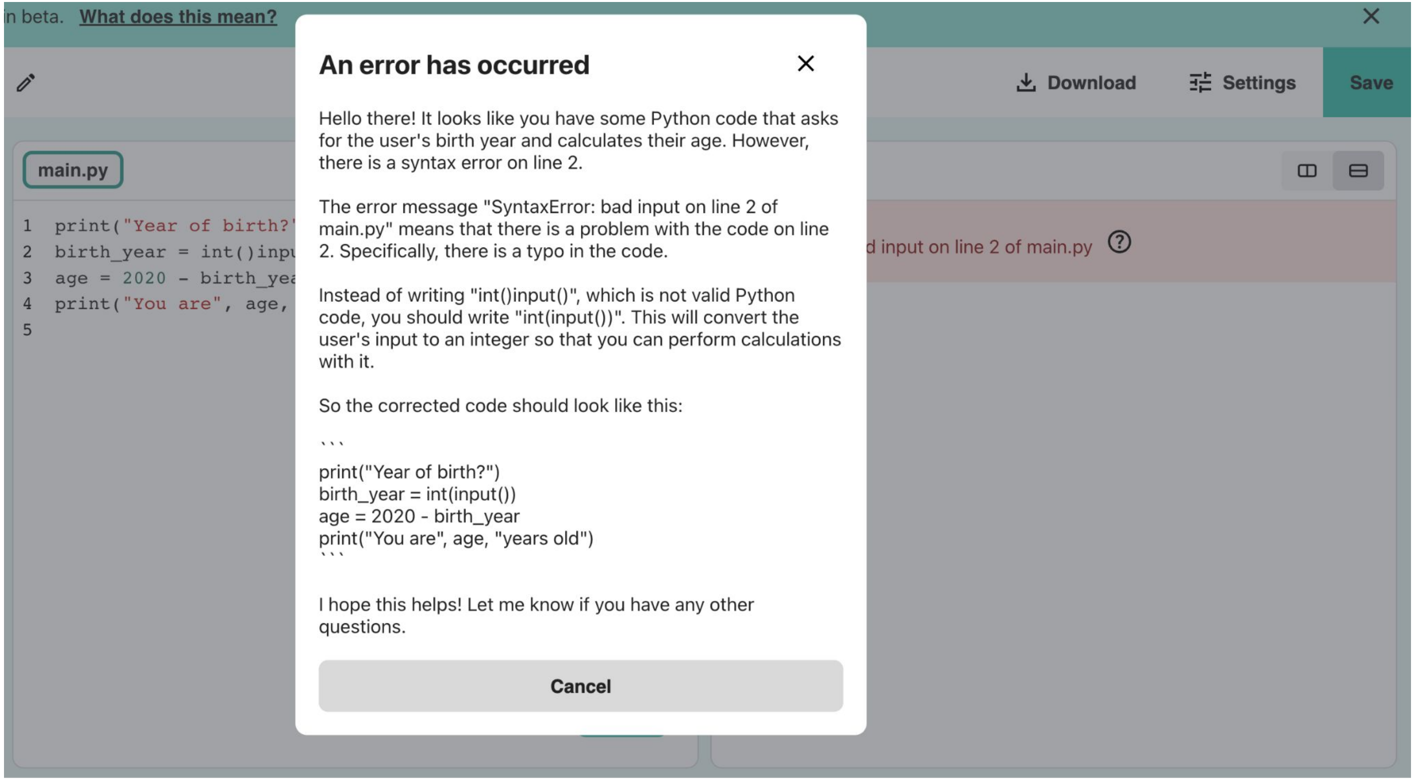
RPF Python code editor - original



```
print(hello 'world')
```

```
SyntaxError: bad input on line 1 of main.py
```

RPF Python code editor - LLM prototype



```
1  print("Year of birth?"
2  birth_year = int()inpu
3  age = 2020 - birth_yea
4  print("You are", age,
5
```

An error has occurred

Hello there! It looks like you have some Python code that asks for the user's birth year and calculates their age. However, there is a syntax error on line 2.

The error message "SyntaxError: bad input on line 2 of main.py" means that there is a problem with the code on line 2. Specifically, there is a typo in the code.

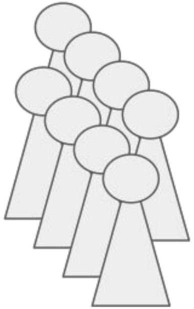
Instead of writing "int()input()", which is not valid Python code, you should write "int(input())". This will convert the user's input to an integer so that you can perform calculations with it.

So the corrected code should look like this:

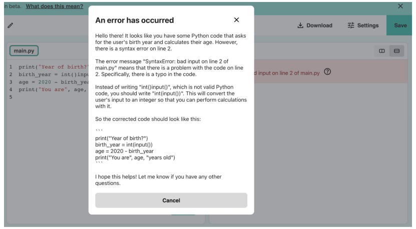
```
print("Year of birth?")
birth_year = int(input())
age = 2020 - birth_year
print("You are", age, "years old")
```

I hope this helps! Let me know if you have any other questions.

Methodology



8 expert secondary educators



```
1  print("Year of birth?"
2  birth_year = int()inpu
3  age = 2020 - birth_yea
4  print("You are", age,
5
```

d input on line 2 of main.py

An error has occurred

Hello there! It looks like you have some Python code that asks for the user's birth year and calculates their age. However, there is a syntax error on line 2.

The error message "SyntaxError: bad input on line 2 of main.py" means that there is a problem with the code on line 2. Specifically, there is a typo in the code.

Instead of writing "int()input()", which is not valid Python code, you should write "int(input())". This will convert the user's input to an integer so that you can perform calculations with it.

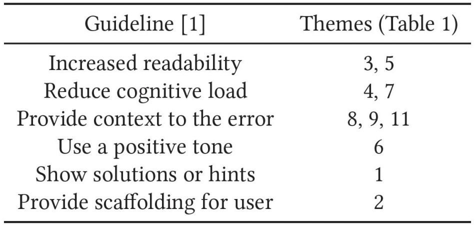
So the corrected code should look like this:

```
```
print("Year of birth?")
birth_year = int(input())
age = 2020 - birth_year
print("You are", age, "years old")
```
```

I hope this helps! Let me know if you have any other questions.

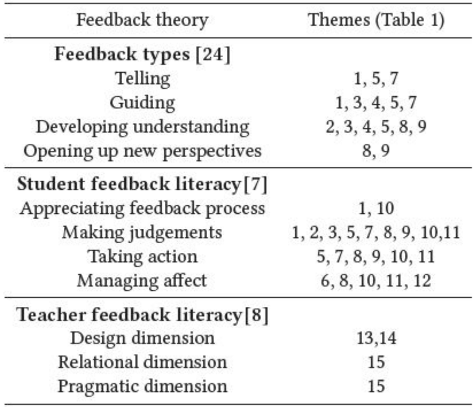
Semi-structured, activity-based interviews

Table 2: Correlation of PEM guidelines [1] to themes



Guideline [1]	Themes (Table 1)
Increased readability	3, 5
Reduce cognitive load	4, 7
Provide context to the error	8, 9, 11
Use a positive tone	6
Show solutions or hints	1
Provide scaffolding for user	2

Table 3: Correlation of feedback theories to themes



Feedback theory	Themes (Table 1)
Feedback types [24]	
Telling	1, 5, 7
Guiding	1, 3, 4, 5, 7
Developing understanding	2, 3, 4, 5, 8, 9
Opening up new perspectives	8, 9
Student feedback literacy[7]	
Appreciating feedback process	1, 10
Making judgements	1, 2, 3, 5, 7, 8, 9, 10,11
Taking action	5, 7, 8, 9, 10, 11
Managing affect	6, 8, 10, 11, 12
Teacher feedback literacy[8]	
Design dimension	13,14
Relational dimension	15
Pragmatic dimension	15

Qualitative inductive / deductive data analysis & Correlation to PEM guidelines & feedback literacy

Themes

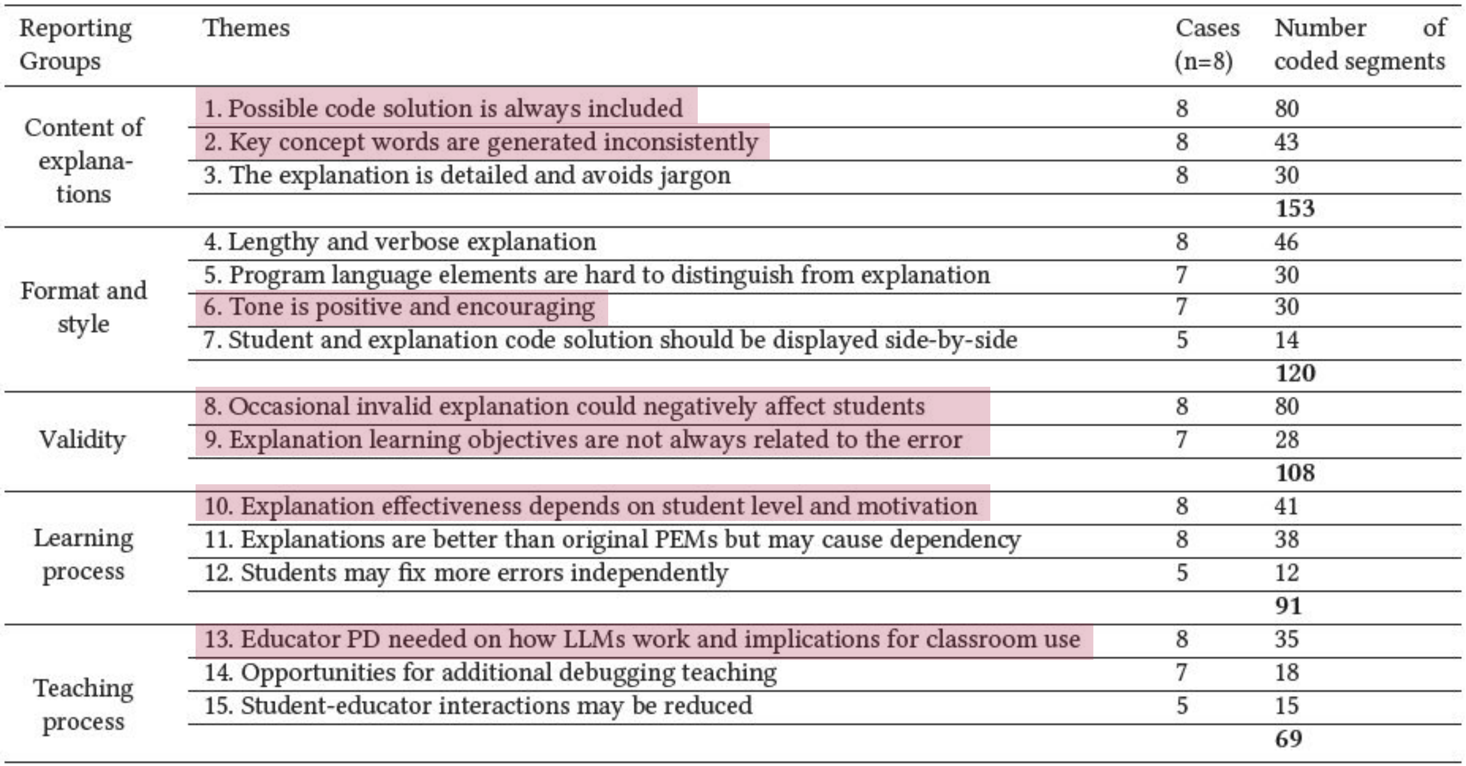


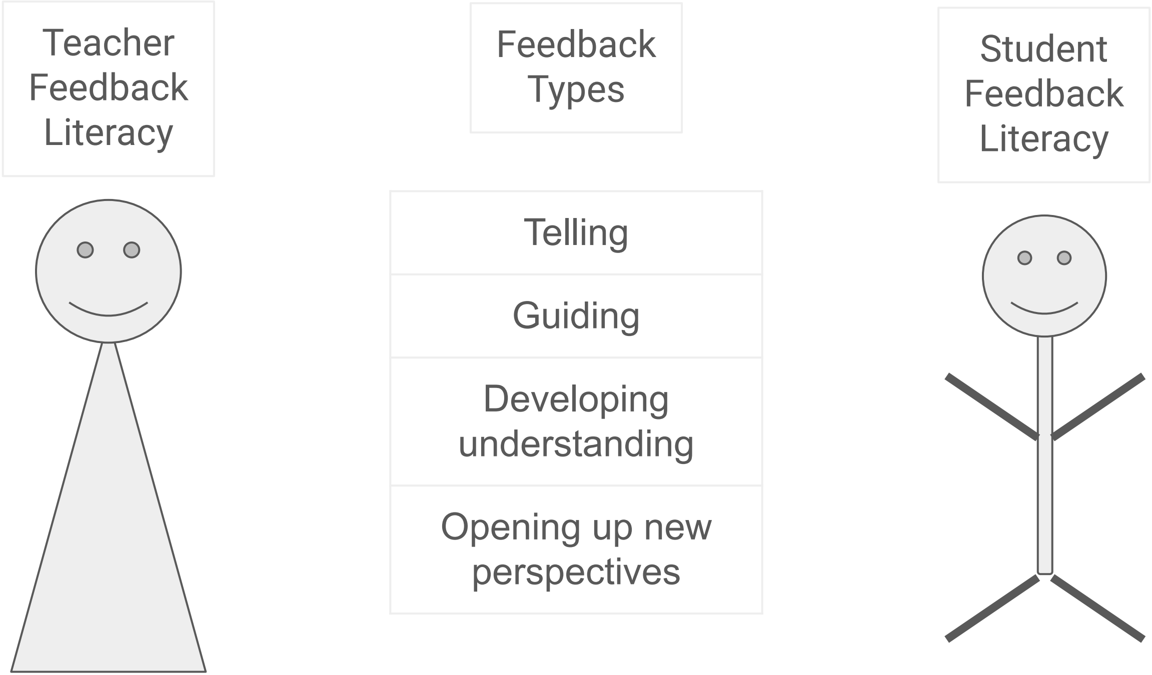
Table 1: Themes - Educators' practical considerations on using LLM explanations of program error messages

Reporting Groups	Themes	Cases (n=8)	Number of coded segments
Content of explanations	1. Possible code solution is always included	8	80
	2. Key concept words are generated inconsistently	8	43
	3. The explanation is detailed and avoids jargon	8	30
			153
Format and style	4. Lengthy and verbose explanation	8	46
	5. Program language elements are hard to distinguish from explanation	7	30
	6. Tone is positive and encouraging	7	30
	7. Student and explanation code solution should be displayed side-by-side	5	14
			120
Validity	8. Occasional invalid explanation could negatively affect students	8	80
	9. Explanation learning objectives are not always related to the error	7	28
			108
Learning process	10. Explanation effectiveness depends on student level and motivation	8	41
	11. Explanations are better than original PEMs but may cause dependency	8	38
	12. Students may fix more errors independently	5	12
			91
Teaching process	13. Educator PD needed on how LLMs work and implications for classroom use	8	35
	14. Opportunities for additional debugging teaching	7	18
	15. Student-educator interactions may be reduced	5	15
			69

Correlation to Enhanced PEM Guidelines [1]

PEM Guideline	Theme	Group
Show solutions or hints	1. Possible code solution is always included	Content of explanations
Provide scaffolding for user	2. Key concept words are generated inconsistently	
Increased readability	3. The explanation is detailed and avoids jargon	
	5. Program language elements are hard to distinguish from explanation	Format and style
Reduce cognitive load	4. Lengthy and verbose explanation	
	7. Student and explanation code solution should be displayed side-by-side	
Use a positive tone	6. Tone is positive and encouraging	
Provide context to the error	8. Occasional invalid explanation could negatively affect students	Validity
	9. Explanation learning objectives are not always related to the error	
	11. Explanations are better than original PEMs but may cause dependency	Learning process

What is feedback literacy?



Teacher Feedback Literacy

Feedback Types

Telling

Guiding

Developing understanding

Opening up new perspectives

Student Feedback Literacy

Providing feedback (an explanation) is a social interaction.

(Inspired by p[2] Rohlfing et al., 2020)

Feedback types [2]

Feedback type	Educator role	Student role
Telling	Unidirectional transmission of correct information	Passive
Guiding	Point in the right direction	Active as applies knowledge
Developing understanding	Targeted teaching	Active as constructs or adjusts knowledge
Opening up new perspectives	Presenting new perspectives	Active as interprets and evaluates new knowledge

(McLead, Bond & Nicholson, 2015)

Student feedback literacy [3]

Student feedback literacy requires students to:

a) appreciating feedback processes;

b) making judgements;

c) taking action;

d) managing affect;

(Carless, 2018)

Teacher feedback literacy [4]

Teacher feedback literacy requires the teacher to:

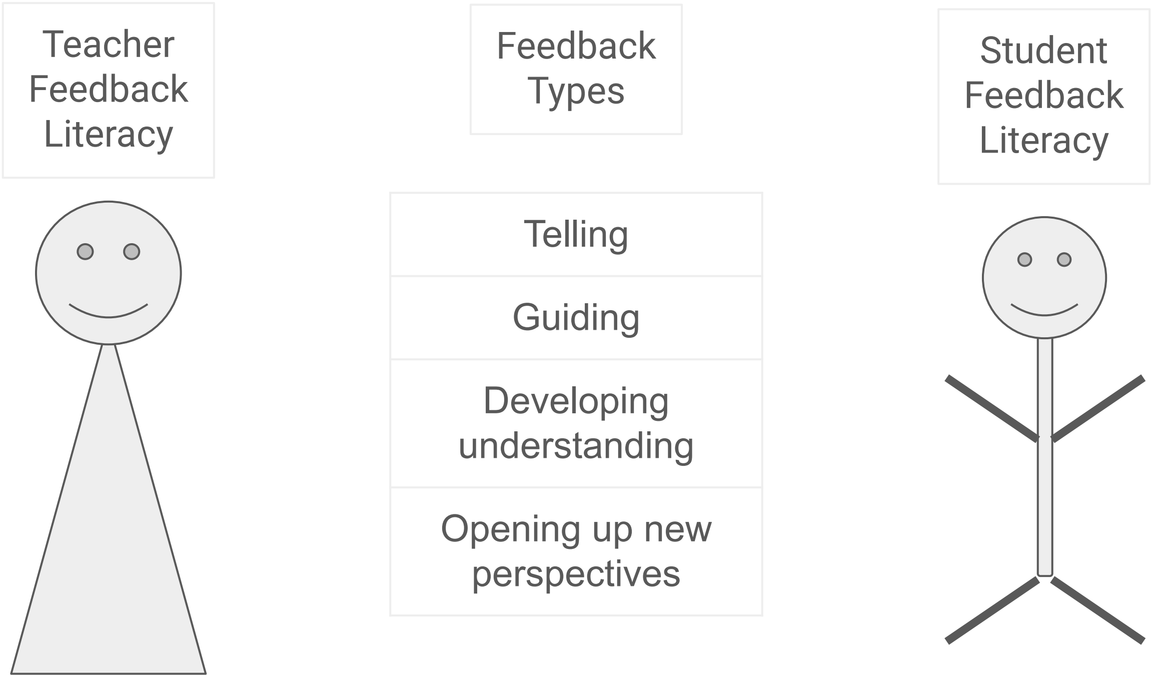
a) design;

b) relational;

c) pragmatic

(Carless & Winstone, 2023)

Feedback literacy



Teacher Feedback Literacy

Feedback Types

- Telling
- Guiding
- Developing understanding
- Opening up new perspectives

Student Feedback Literacy

Providing feedback (an explanation) is a social interaction.

(Inspired by p[2] Rohlfing et al., 2020)

Correlation to Feedback Types [2]

Feedback Type	**Enabling**	**Limiting**
Telling	1. Possible code solution is always included 7. Student and explanation code solution should be displayed side-by-side	5. Program language elements are hard to distinguish from explanation
Guiding	3. The explanation is detailed and avoids jargon	1. Possible code solution is always included 4. Lengthy and verbose explanation 7. Student and explanation code solution should be displayed side-by-side
Developing understanding	3. The explanation is detailed and avoids jargon	2. Key concept words are generated inconsistently 8. Occasional invalid explanation could negatively affect students 9. Explanation learning objectives are not always related to the error
Opening up new perspectives	8. Occasional invalid explanation could negatively affect students 9. Explanation learning objectives are not always related to the error	

Correlation to Feedback Types [2]

Feedback Type	**Enabling**	**Limiting**
Telling	1. Possible code solution is always included 7. Student and explanation code solution should be displayed side-by-side	5. Program language elements are hard to distinguish from explanation
Guiding	3. The explanation is detailed and avoids jargon	1. Possible code solution is always included 4. Lengthy and verbose explanation 7. Student and explanation code solution should be displayed side-by-side
Developing understanding	3. The explanation is detailed and avoids jargon	2. Key concept words are generated inconsistently 8. Occasional invalid explanation could negatively affect students 9. Explanation learning objectives are not always related to the error
Opening up new perspectives	8. Occasional invalid explanation could negatively affect students 9. Explanation learning objectives are not always related to the error	

Correlation to Student Feedback Literacy [3]

SFL dimension	Enabling	Limiting
	10. Explanation effectiveness depends on student level and motivation	
Appreciating feedback processes		1. Possible code solution is always included
Making judgements	3. The explanation is detailed and avoids jargon	1. Possible code solution is always included 2. Key concept words are generated inconsistently 5. Program language elements are hard to distinguish from explanation 8. Occasional invalid explanation could negatively affect students 9. Explanation learning objectives are not always related to the error
Taking action	11. Explanations are better than original PEMs but may cause dependency	7. Student and explanation code solution should be displayed side-by-side
Managing affect	6. Tone is positive and encouraging 12. Students may fix more errors independently	8. Occasional invalid explanation could negatively affect students

Correlation to Student Feedback Literacy [3]

SFL dimension	Enabling	Limiting
	10. Explanation effectiveness depends on student level and motivation	
Appreciating feedback processes		1. Possible code solution is always included
Making judgements	3. The explanation is detailed and avoids jargon	1. Possible code solution is always included 2. Key concept words are generated inconsistently 5. Program language elements are hard to distinguish from explanation 8. Occasional invalid explanation could negatively affect students 9. Explanation learning objectives are not always related to the error
Taking action	11. Explanations are better than original PEMs but may cause dependency	7. Student and explanation code solution should be displayed side-by-side
Managing affect	6. Tone is positive and encouraging 12. Students may fix more errors independently	8. Occasional invalid explanation could negatively affect students

Correlation to Student Feedback Literacy [3]

SFL dimension	Enabling	Limiting
	10. Explanation effectiveness depends on student level and motivation	
Appreciating feedback processes		1. Possible code solution is always included
Making judgements	3. The explanation is detailed and avoids jargon	1. Possible code solution is always included 2. Key concept words are generated inconsistently 5. Program language elements are hard to distinguish from explanation 8. Occasional invalid explanation could negatively affect students 9. Explanation learning objectives are not always related to the error
Taking action	11. Explanations are better than original PEMs but may cause dependency	7. Student and explanation code solution should be displayed side-by-side
Managing affect	6. Tone is positive and encouraging 12. Students may fix more errors independently	8. Occasional invalid explanation could negatively affect students

Correlation to Teacher Feedback Literacy [4]

TFL dimension	**Enabling**	**Limiting**
Design	14. Opportunities for additional debugging teaching	13. Educator PD needed on how LLMs work and implications for classroom use
Relational		15. Student-educator interactions may be reduced
Pragmatic	15. Student-educator interactions may be reduced	

Correlation to Teacher Feedback Literacy [4]

TFL dimension	**Enabling**	**Limiting**
Design	14. Opportunities for additional debugging teaching	13. Educator PD needed on how LLMs work and implications for classroom use
Relational		15. Student-educator interactions may be reduced
Pragmatic	15. Student-educator interactions may be reduced	

Combining enhanced PEM guidelines with Feedback Literacy theory

- LLM content: encouraging, detailed, jargon-free, use keywords consistently, be in line with learning objectives, and a solution code should not be included or delayed.
- IDE design should ensure that educators can see how their students use LLM explanations, and enable users to manage invalid or unrelated explanations.
- To optimise programming teaching using LLMs, professional development and student learning materials combining feedback literacy, PEMs and LLMs should be researched, co-created and delivered.

Limitations

- Limited number of teachers
- No prompting experimentation
- No direct student investigation

Conclusions

Educators prefer the LLM explanations to fulfil a **guiding** and **developing understanding** role, rather than **telling**;

Educators talked about the ways in which the LLM explanations help or hinder students to **making judgements** and **action the feedback** in the explanations;

Educators discussed the need for PD to manage feedback processes inclusive of LLM feedback (**design**) and address issues resulting from reduced opportunities to interact with students (**relational**)

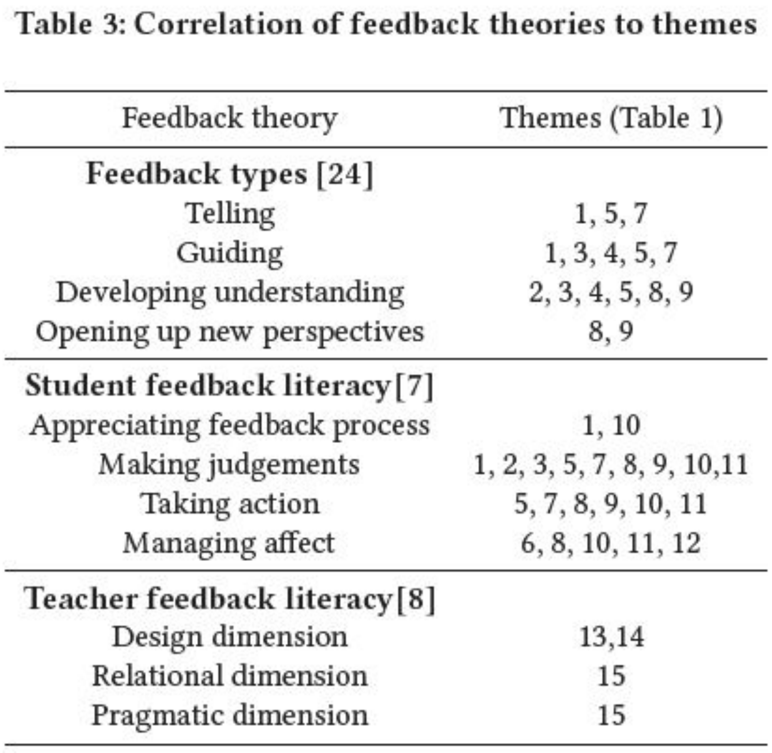


Table 3: Correlation of feedback theories to themes

Feedback theory	Themes (Table 1)
Feedback types [24]	
Telling	1, 5, 7
Guiding	1, 3, 4, 5, 7
Developing understanding	2, 3, 4, 5, 8, 9
Opening up new perspectives	8, 9
Student feedback literacy [7]	
Appreciating feedback process	1, 10
Making judgements	1, 2, 3, 5, 7, 8, 9, 10,11
Taking action	5, 7, 8, 9, 10, 11
Managing affect	6, 8, 10, 11, 12
Teacher feedback literacy [8]	
Design dimension	13,14
Relational dimension	15
Pragmatic dimension	15

Feedback Literacy - thoughts

- Is feedback literacy a helpful concept for your practice? If so, in what way?
- Does feedback literacy change how you think of LLM explanations, or the feedback we think might be useful for students in general?
- How does feedback literacy already manifest practically in your work?
- What other concepts do you relate to feedback literacy?

References

[1] Brett Becker, Paul Denny, Raymond Pettit, Durell Bouchard, Dennis Bouvier, Brian Harrington, Amir Kamil, Amey Karkare, Chris Mcdonald, Peter-Michael Osera, Janice Pearce, and James Prather. 2019. Compiler Error Messages Considered Unhelpful: The Landscape of Text-Based Programming Error Message Research. In Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education (Aberdeen, Scotland Uk) (ITiCSE-WGR '19). ACM, New York, NY, USA, 177–210. https://doi.org/10.1145/3344429.3372508

[2] Angela J McLean, Carol H Bond, and Helen D Nicholson. 2015. An anatomy of feedback: a phenomenographic investigation of undergraduate students' conceptions of feedback. Studies in Higher Education 40, 5 (2015), 921–932. https://doi.org/10.1080/03075079.2013.855718

[3] David Carless and David Boud. 2018. The development of student feedback literacy: enabling uptake of feedback. Assessment & Evaluation in Higher Education 43, 8 (2018), 1315–1325. https://doi.org/10.1080/02602938.2018.1463354

[4] David Carless and Naomi Winstone. 2023. Teacher feedback literacy and its interplay with student feedback literacy. Teaching in Higher Education 28, 1 (2023), 150–163

[5] Veronica Cucuiat and Jane Waite. 2024. Feedback Literacy: Holistic Analysis of Secondary Educators' Views of LLM Explanations of Program Error Messages. In Proceedings of the 2024 Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024), July 8–10, 2024, Milan, Italy. ACM, New York, NY, USA, https://doi.org/10.1145/3649217.3653595

Thank you!

Project website

Raspberry Pi
Foundation