Department for Education

Use Cases for Generative AI in Education

Building a proof of concept for Generative AI feedback and resource generation in education contexts: Technical report

August 2024 (updated October 2024)

Government Social Research

Contents

List of figures

List of tables

Executive summary

This report sets out the findings of the technical development work completed as part of the Use Cases for Generative AI in Education project, commissioned by the Department for Education in September 2023. It has been published alongside the User Research Report, which sets out the findings from the ongoing user engagement activity conducted as part of this project.

As part of exploratory work by DfE into the use of Generative AI within education, Faculty has built a bespoke AI-powered tool to demonstrate the potential of using Large Language Models (LLMs) for a specific use case. The tool is a proof of concept (PoC), limited to just the evaluation of Year 4 Literacy work, however, its design acts as a good approximation for other bespoke or open-source tooling that an institution could feasibly set up for its staff or pupils.

Evaluating work against the National Curriculum and providing formative feedback and activities to pupils has previously been identified as a time-intensive process for teachers, and one which AI could support teachers with. This could not only reduce their workload, it is also an area where increased consistency would improve educational outcomes for students.

A key aim of this work was to produce a set of learnings to be made available for reference by the EdTech sector and other education stakeholders, to support the development of similar tools within the sector. These learnings were generated through two approaches:

- An iterative process of experimentation to rate its alignment with educational standards (such as the Early Career Framework) and evaluate whether the feedback and activities meet best teaching practice.
- Extensive user testing with teachers to better understand the performance and usability of the tool.

The insights gained have been distilled into the following set of key learnings, articulated in detail in Section 8.2

- Learnings from building an AI-powered educational tool:
  - Educators must be engaged throughout the development cycle.
  - Highlighting what a pupil has demonstrated and not just their mistakes is crucial.
  - Assessing pupil work in isolation decreases the effectiveness of AI-generated insights.
  - Tools should be able to be customised by and to the specific educator.
  - Considering the educator’s wider workflow is crucial for ensuring usability.

- Learnings from assessing pupil work against the National Curriculum:
  - The most useful applications will blend deterministic and AI-based approaches.
  - Model selection should consider costs as well as performance.
  - Generative AI is more efficient and accurate when given highly structured requirements.
  - Synthetic data can provide benefits over just being a proxy for real pupil work.
- Learnings from generating education-specific content with generative AI:
  - Few-shot learning does not necessary improve performance.
  - While the structuring of instructions is not essential for prompts, it is useful for maintaining them.
  - LLMs are inherently good at providing feedback.
  - Using generative AI to “mark its own homework” is an effective evaluation technique.
- Learnings from generating education-specific content with generative AI:
  - It is crucial to gain parent and school perspectives on how pupils’ work is processed and used.
  - Making use of schools’ existing lines of communication is the most effective way to seek agreement from parents.
  - It will be necessary to design a scalable process for sharing data and removing personal data , if such a tool is to be used across multiple schools, such as a multi-academy trust (MAT).
  - In the information provided to parents and schools, it is important to be clear that the removal of all personal is not guaranteed.

This is an independent report and is not intended to represent the policy position of the Department. Public sector organisations are bound by solution agnosticism and there is no implication that tools mentioned throughout this reporting are the only tools available on the market. The Department was tool agnostic when contracting for this project, and the successful contractor (Faculty Ltd) chose the tools they felt were most applicable. The methodology for utilising a large language model explored in this report would be applicable to other models available on the market.

Introduction

Aims of the technical build

Working with the National Institute of Teaching (NIoT) and the Department for Education (DfE), we have been exploring potential applications for Generative AI in the education sector as part of a wider effort to 'transform a teacher's day to day work'[1] – reducing workload and improving educational outcomes by automating routine tasks. As part of this exploratory work, we have built a proof of concept (PoC) tool to explore the potential for using Large Language Models (LLMs) for these purposes. The tool is targeted towards supporting teachers with evaluating work against the National Curriculum and providing formative feedback and activities to pupils. This has previously been identified as a time-intensive process for teachers, and an area where increased consistency would improve educational outcomes for students. The scope of the PoC was limited to just evaluation of Year 4 Literacy work, however, its design means it could easily be used as a basis for wider tools, extending to other subjects or year groups.

While the tool has been built around a specific education use case, and allows us to understand the effectiveness of Generative AI to successfully undertake that specific use case, the work also supports a wider set of aims, to:

- test the practicality of using Generative AI for predefined applications in educational settings;
- gather more information about what works and the limitations of current GenAI models and approaches for educational purposes, including testing how educators feel about these potential tools and uses; and
- share information about the approaches to optimising a model in order to support the sector to build robust tools, including testing which kinds and formats of content work best in this setting.

The overall purpose of this work was to establish key methods for optimising the performance of Generative AI, rather than primarily creating a usable product. The work therefore prioritised testing and assessing approaches to the use case and not on the core usability of the tool.

Justification for the selected education use case

Prior to this project we held a Generative AI in Education Hackathon, inviting participants from across the education sector to test a range of education-related use cases for LLMs. The findings of this event produced a short list of options from which the PoC use case

[1] Quote from Secretary of State Gillian Keegan's speech addressing BETT 2023

was chosen. A set of criteria was developed with the DfE to guide this decision, which is summarised in Table 1.

Table 1: Criteria for assessment of PoC candidate use cases

Criteria	Description
Innovation	Does the PoC demonstrate the art of the possible with GenAI across the sector?
Learning potential	Does the PoC allow us to test and learn what works when using GenAI for education? (Considering the viability of both the use cases and techniques for model optimisation)
Practicality	Does the PoC represent a tangible, practical solution for an actual user or group of users?
Feasibility	Would learnings from the PoC provide a reasonable basis for a product, should the sector wish to use it as a basis for further development?
Novelty	To what degree has the PoC's aims already been explored by other initiatives?
Strength of evidence	To what degree is the PoC supported by the Hackathon findings, DfEs Call for Evidence interim findings and/or NIoT's GenAI in Education consultation?

The chosen use case received the following assessment against the criteria:

- **Innovation** – Scored highly: The use of generative AI to provide marking/feedback capabilities and generate revision activities is commonplace with several commercial solutions available however, the combination of the two remains unexplored.
- **Learning potential** – Scored averagely: As both feedback generation and revision activity generation have both been explored as use cases for generative AI, both of these use cases are known to be viable. Learnings therefore would be restricted to how well the technical methodology linking the two processes together functions.
- **Practicality** – Scored highly: A tool able to link feedback with generated revision activities, could save time for teachers, improving the tailoring of homework activities to students' individual needs as closely as possible.

- **Feasibility** – Score averagely: It is plausible that such a tool could be developed further into a future product that could support the sector if taken on by a company or education establishment, however, there would need to be consideration as to how best differentiate from the many tools currently available on the market for feedback generation and revision activity creation.
- **Novelty** – Scored highly: The combination of the two components of the POC is a novel idea, despite both having been explored separately by several commercial tools. An example being Oak National Academy's quiz builder, although, at the time of assessment, this was limited to generating multiple choice options for existing questions.
- **Strength of evidence** – Scored averagely: The output of ChatGPT-generated student feedback was poorly rated by Hackathon participants; however, the outputs of lesson material generation and the use of GenAI as a teaching aid were rated positively.

Project structure and phases

The development of the PoC tool comprised of three main phases of work:

1. **Minimum Viable Product (MVP) build** – 6 weeks

   The first phase of the project focused on designing and building the core functionality of the tool. A 'core user group' of 8 educators was engaged to support the development of the MVP tool, helping to define user requirements and providing early feedback on the design and performance. To facilitate testing of the tool during this phase, a bank of partially synthetic student work was created, enabling us to programmatically assess how well it was evaluating work against the National Curriculum.

2. **User Testing** – 2 weeks

   Once the core tool was completed, a wider round of user engagement with teachers who were part of the AI in School Initiative was performed to better understand the performance and usability of the tool, as well as generate priorities for the optimisation stage. As part of this, teachers were asked to review the feedback generated for the bank of synthetic student work and rate its alignment with educational standards (such as the Early Career Framework) to evaluate whether the feedback and activities meet best teaching practice.

3. **Optimisation** – 4 weeks

   During the final stage, a set of experiments were run with the tool to improve the performance of the Generative AI components and identify key insights for building robust tools. These experiments mainly centred on prompt-engineering techniques and how best to incorporate key reference data (such as the Early Career Framework) into Generative AI applications. To understand the impact on performance of these experiments, assessment criteria were produced with input from the DfE and core user group. This approach guaranteed we could test feedback and tasks were both informative and accessible to students, as well as meeting UK educational guidelines and pedagogical best practices.

The proof-of-concept tool

Exemplar user journey

The tool currently has two main user journeys; one that allows the user to upload their own pupil work, while the other uses a bank of pre-processed essays. These only differ at the start, and as such this document will describe only the user upload journey, which is more realistic to how an end user would interact with the tool if this were to be integrated into their workflow.

User journey:

1. When the user first accesses the tool, they are presented with the essay upload page. This allows them to upload details of the specific task a pupil was set, and the pupil's piece of work. The user will copy and paste these into the relevant input fields and click the "Submit essay" button.

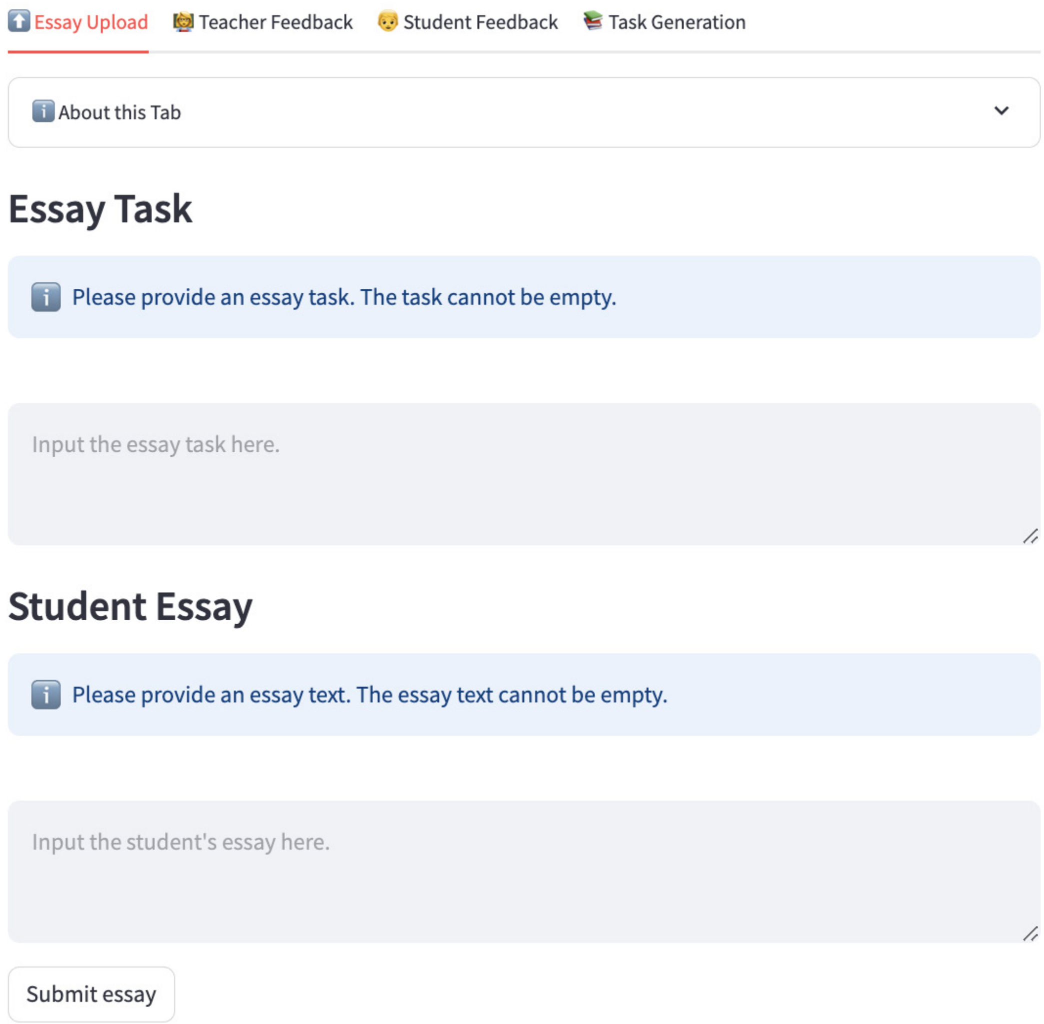


Figure 1: Upload page

2. The tool analyses the work and based on guidance from the National Curriculum and other materials, identifies areas for improvement. These are presented to the user by marking the specific parts of the text in red. The user can then hover over any of the highlighted areas for more information, including the type of error, the correction and the Year group associated with the error according to the National Curriculum.

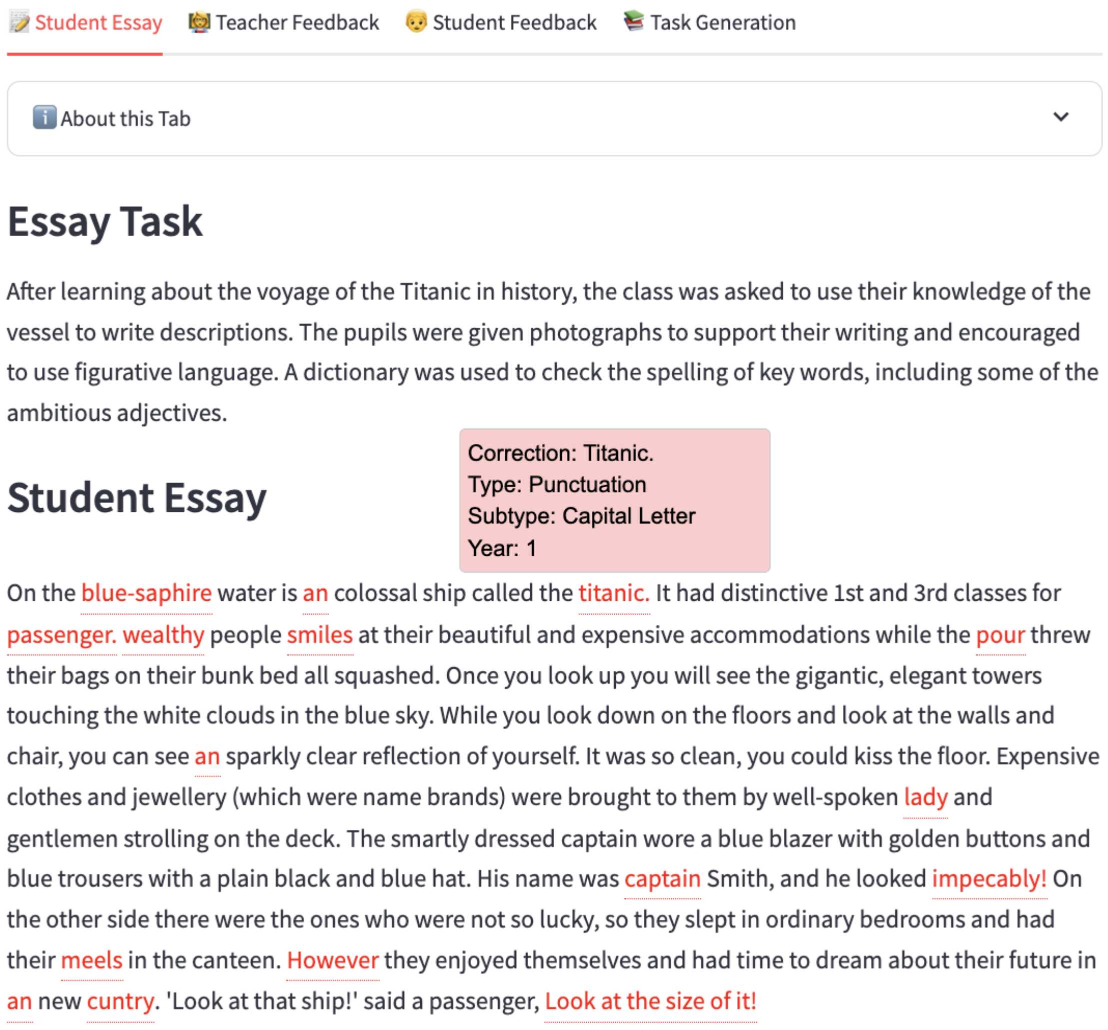
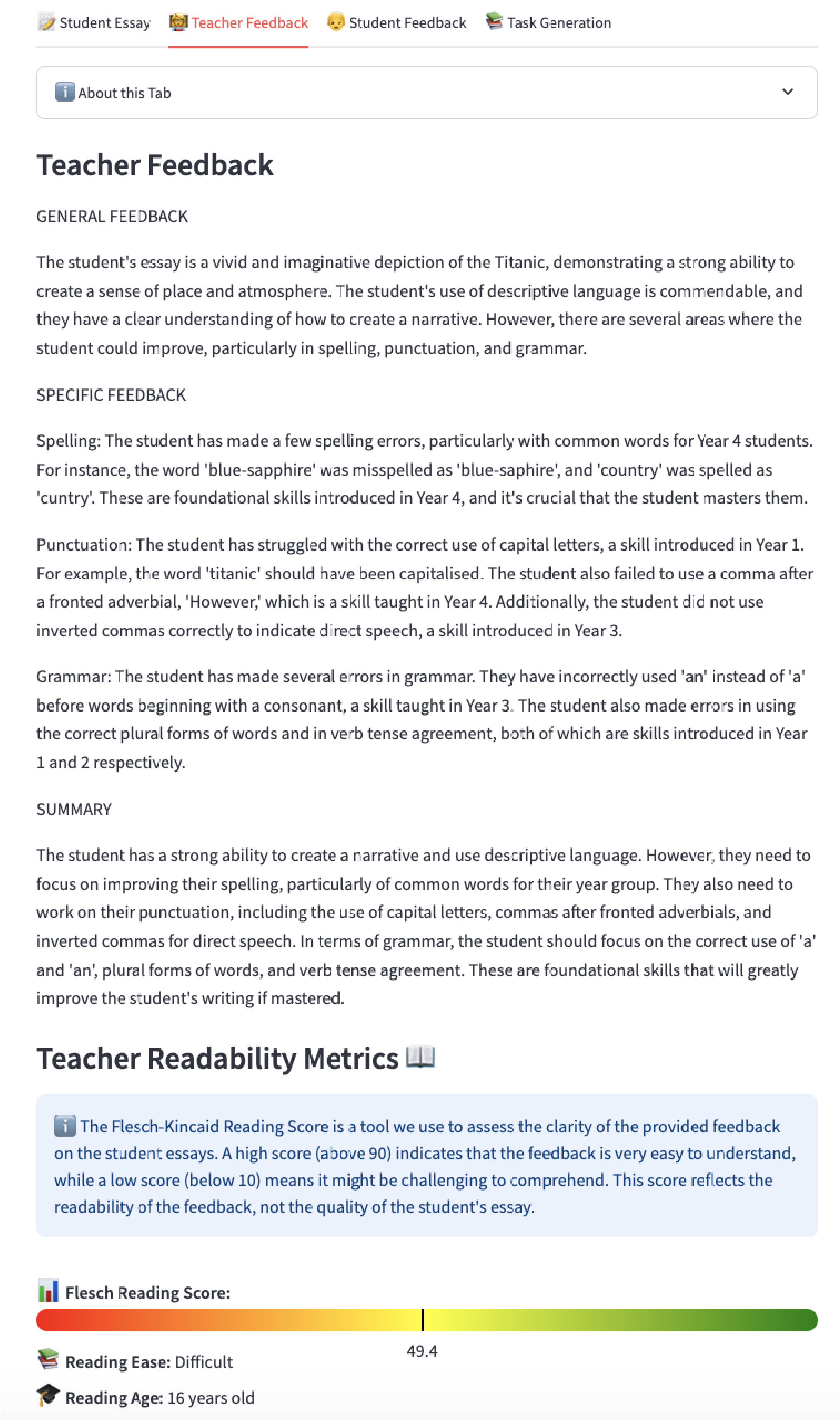


Figure 2: Assessed essay tab

3. The user then navigates to the “Teacher Feedback" tab to view an assessment of the pupil’s work aimed at the teacher. This generated feedback gives both a summary on how well the pupil did in relation to the task, as well as specific details on their spelling, punctuation and grammar based on the errors that the tool has detected (e.g. ‘The pupil made several errors when using fronted adverbials, for example they wrote…’). The tool also displays an assessment of the readability of the generated feedback, using the Flesch-Kincaid Reading Score.



Student Essay | Teacher Feedback | Student Feedback | Task Generation

About this Tab

Teacher Feedback

GENERAL FEEDBACK

The student's essay is a vivid and imaginative depiction of the Titanic, demonstrating a strong ability to create a sense of place and atmosphere. The student's use of descriptive language is commendable, and they have a clear understanding of how to create a narrative. However, there are several areas where the student could improve, particularly in spelling, punctuation, and grammar.

SPECIFIC FEEDBACK

Spelling: The student has made a few spelling errors, particularly with common words for Year 4 students. For instance, the word 'blue-sapphire' was misspelled as 'blue-saphire', and 'country' was spelled as 'cuntry'. These are foundational skills introduced in Year 4, and it's crucial that the student masters them.

Punctuation: The student has struggled with the correct use of capital letters, a skill introduced in Year 1. For example, the word 'titanic' should have been capitalised. The student also failed to use a comma after a fronted adverbial, 'However,' which is a skill taught in Year 4. Additionally, the student did not use inverted commas correctly to indicate direct speech, a skill introduced in Year 3.

Grammar: The student has made several errors in grammar. They have incorrectly used 'an' instead of 'a' before words beginning with a consonant, a skill taught in Year 3. The student also made errors in using the correct plural forms of words and in verb tense agreement, both of which are skills introduced in Year 1 and 2 respectively.

SUMMARY

The student has a strong ability to create a narrative and use descriptive language. However, they need to focus on improving their spelling, particularly of common words for their year group. They also need to work on their punctuation, including the use of capital letters, commas after fronted adverbials, and inverted commas for direct speech. In terms of grammar, the student should focus on the correct use of 'a' and 'an', plural forms of words, and verb tense agreement. These are foundational skills that will greatly improve the student's writing if mastered.

Teacher Readability Metrics

The Flesch-Kincaid Reading Score is a tool we use to assess the clarity of the provided feedback on the student essays. A high score (above 90) indicates that the feedback is very easy to understand, while a low score (below 10) means it might be challenging to comprehend. This score reflects the readability of the feedback, not the quality of the student's essay.

Flesch Reading Score:

49.4

Reading Ease: Difficult

Reading Age: 16 years old

Figure 3: Teacher-focused feedback tab

4. Navigating to the “Student Feedback" tab, the user sees feedback intended for the pupil. The language here is much more encouraging and focuses on how the pupil can improve, rather than just on listing errors. The user can either share this directly with the pupil or use it as the basis for their own feedback. As with the previous tab, the tool displays an assessment of the readability (not shown in the picture opposite), indicating whether the feedback is pitched at a level the pupil will understand.

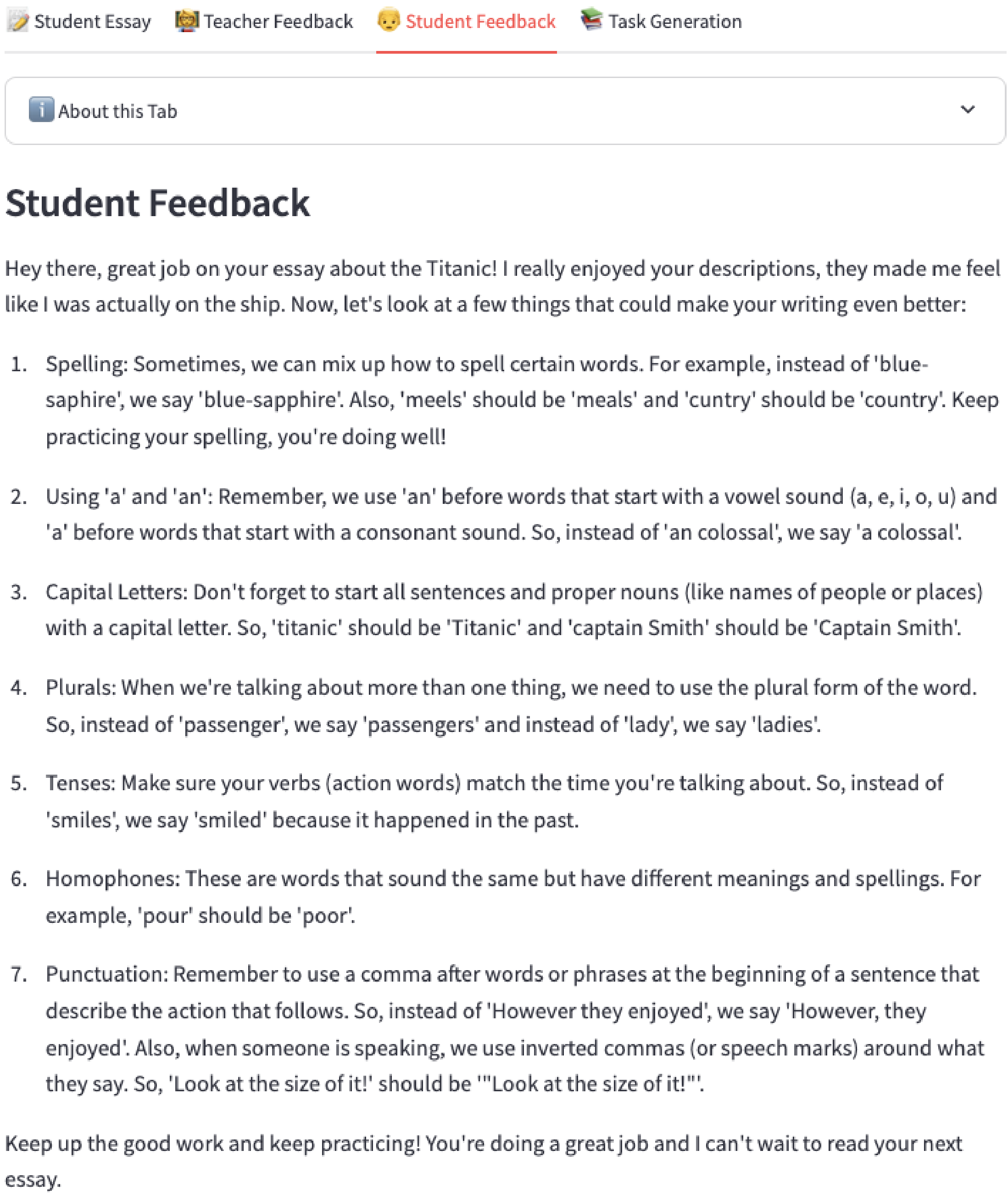


Figure 4: Student-focused feedback tab

5. Finally, the user can view a selection of formative worksheets the tool has generated based on the errors and feedback. Four varieties of worksheet are generated; the first focused generally on the most important errors as defined by their order in the National Curriculum, and rest specifically on spelling, punctuation and grammar respectively. Again, these could be used directly as they are, or as a first draft for the user to refine.

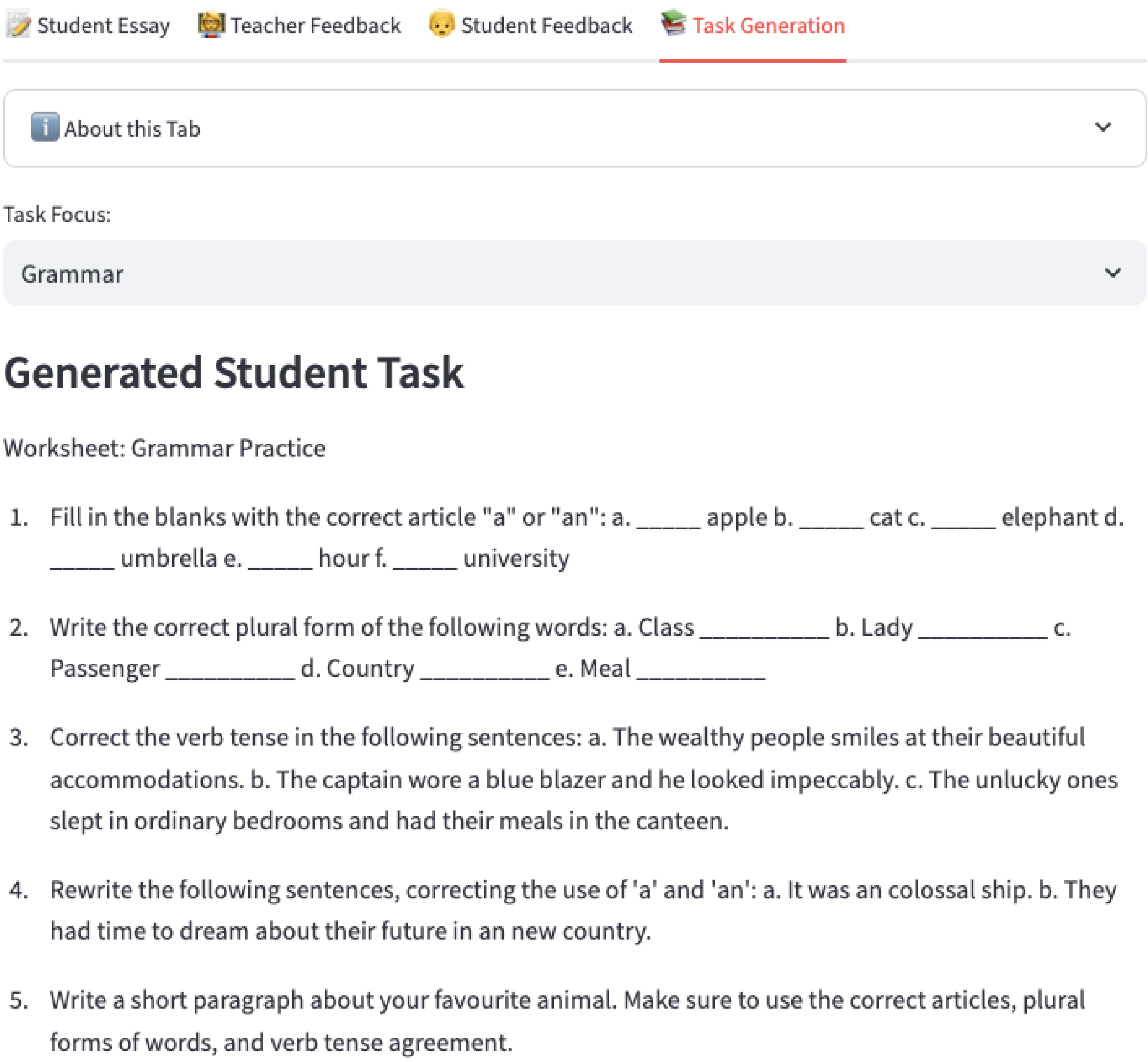


Figure 5: Formative worksheets tab

Overview of the architecture

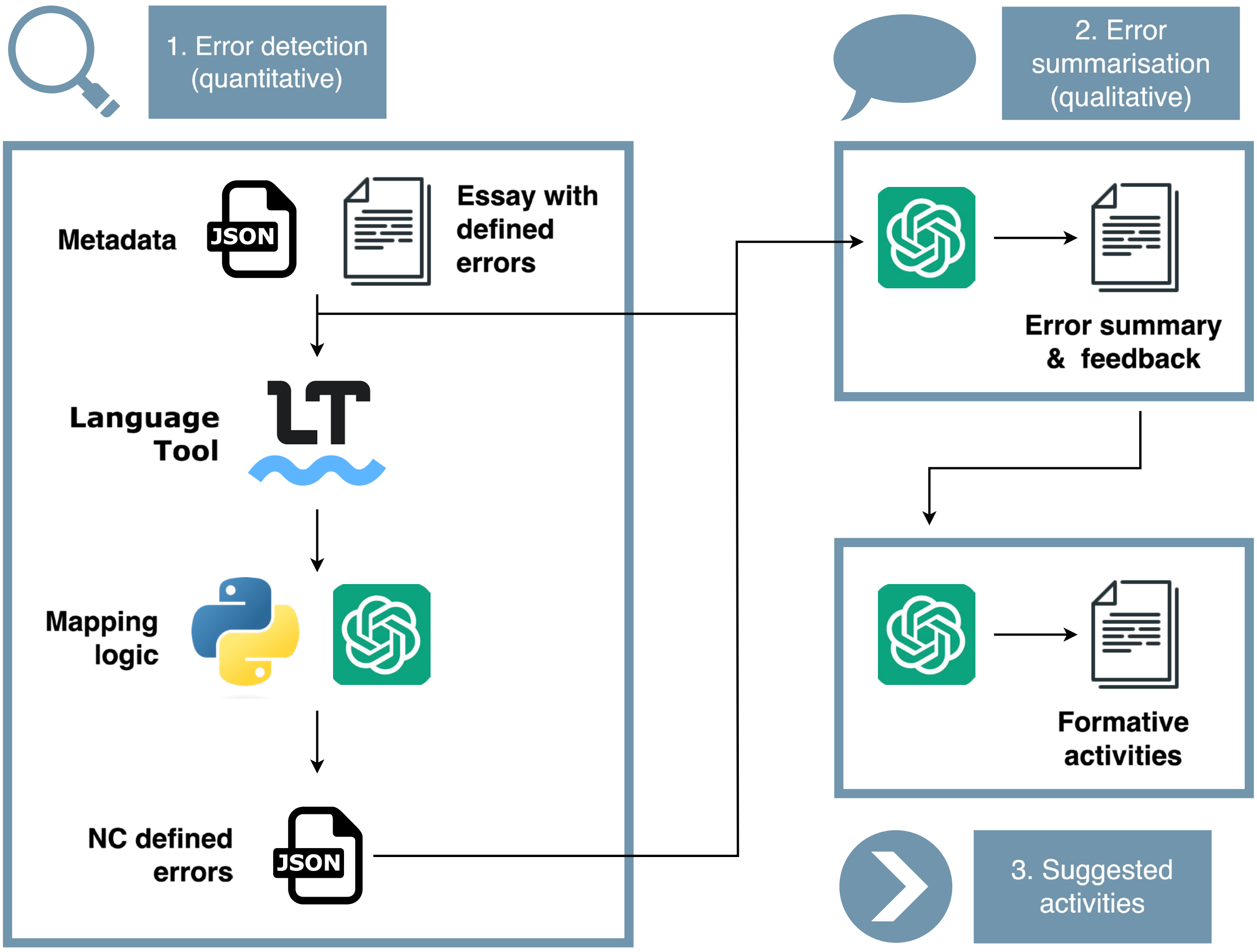


Figure 6: A high level overview of the full tool pipeline

Figure 6 shows how a piece of workflows through the tool, highlighting the specific technologies used during each stage. At a high-level the structure of the tool is split into three main components mimicking the tasks it is designed to support with: assessing work against the National Curriculum, providing actionable feedback, and generating formative worksheet activities. This architecture not only allows for a better user experience, as it follows how a teacher might approach the exercise but takes advantage of the improved performance from LLMs when complex tasks are broken down into more manageable steps.

The tool works in the following way:

1. Assessment against the National Curriculum:

    a. The tool first detects all the grammar, punctuation and spelling errors in the pupil's work. This is done in two parts. Firstly [LanguageTool]() (LT), an open-source spell-checker, is used to detect basic errors, such as misspellings. Secondly, GPT4 is used to detect more nuanced errors such as incorrect uses of vocabulary.

    b. The tool then uses a mixture of rules and GPT4 to align these detected errors to the relevant parts of the National Curriculum, allowing it to assign the year that part of language should have been learnt.
    c. The tool then combines the detected errors, their corrections, types and associated years together and passes them onto the next stage.
2. Formative Feedback Generation:
    a. GPT4 is first instructed to generate feedback for a teacher. It is provided with the original pupil work, along with the information on the detected errors, and a set of guidelines on how to structure the response so that it is appropriate for an educator.
    b. GPT4 is then instructed to generate feedback for the pupil. It is again provided with the original pupil work, along with the information on the detected errors, and a set of guidelines on how to structure the response so that it is appropriate for a pupil.
3. Formative Task Generation:
    a. GPT4 is provided with the original pupil work, the information on the detected errors, and the teacher-focused feedback generated in the previous step. It is then instructed to generate a worksheet for the pupil to support their learning, based on all the errors detected.
    b. GPT4 is then prompted another three times in a similar fashion to generate worksheets focused solely on the grammar, spelling and punctuation errors respectively.

All the outputs from the various stages are presented back to the user as they become available. They are also stored in a database so that they can be retrieved during later stages, but also allowing the user to go back to previously processed work.

Outline of models used

Two sets of models were used in the tool. The first set of models are offered by LanguageTool, used to detect spelling, punctuation, and grammatical errors. LT finds mistakes in text by assigning each word a grammatical category and then comparing the analysed text to a list of spelling and grammar rules. Two different models were tested, LT Lite and LT+, which differ in size and the amount of text context used to detect errors. The performance and cost of these different models is addressed in Section 5.5.1.

The second set of models is the GPT model family developed by OpenAI. These LLMs are generative models based on the transformer architecture, which have an attention mechanism allowing them to generate new text based on previously seen text. GPT models are pre-trained on an extremely large and varied corpus consisting of books and internet content, which allow them to perform text-based tasks described using natural language. Instructions given to GPT models in natural language are called prompts.

In this project, GPT models are used for a range of tasks, including error detection, feedback and task generation, as well as generation evaluation. OpenAI offers a variety of models, differing in size, cost and time to run, and maximum input length. Different models were tested for each of the tool components, comparing performance and cost. A range of models are used in the final version of the tool, including GPT-3.5-Turbo and GPT-4. Further details on the performance and cost of different OpenAI models are described in Section 5.5.2 and Section 5.5.3.

AI safety considerations

While the proof-of-concept tool was never intended to be used for more than testing and demonstration purposes, it is still important to consider how guardrails can be built into a system such as this. The inclusion of AI safety techniques is made more important given its use is in an educational setting, as it is likely students would try to adversarially test any tool known to be using Generative AI. To illustrate how this can be achieved, two types of guardrails were built into the tool; one to perform content moderation and the second to protect against prompt injection.

The content moderation guardrail works to prevent harmful or illicit material being evaluated by the tool. It first uses the OpenAI Moderation API to catch content that breaks OpenAI's usage policy, before some bespoke prompts[2] enable GPT4 to check for less severe content that is not suitable for Year 4 English essays.

Prompt injection is the process of adding additional instructions into the content passed to an LLM, altering the output of the model with a view to it returning sensitive information. It is not inconceivable that if such a tool was to be extended, it could hold or have access to sensitive information on pupils. To protect against this, the tool uses GPT to assess submitted content for potential injected instructions at the earliest stages of the tool. If detected, the tool stops the process.

[2] Using GPT-4 for content moderation https://openai.com/blog/using-gpt-4-for-content-moderation

User testing

Output-focused user testing

Overview of the approach

In this phase of user testing, six teachers were asked to evaluate outputs generated by the tool. To measure responses, Faculty provided teachers with semi-structured questionnaires which took two hours to complete. Teachers were presented with a bank of synthetic student work to review and assess the tool's performance and accuracy. Teachers were prompted to comment on the following outputs:

- Feedback to support the teacher with their assessment of student work.
- Feedback to be provided directly to a student.
- A revision activity to be suggested for a student which is based on the feedback generated.

Structurally, the questionnaire was organised into four sections:

1. **Getting to Know You**: the first section sought to understand teachers' experiences and general approach to producing feedback for pupils on their work.
2. **Evaluating AI-generated Feedback and a Revision Activity**: the second section gave teachers the opportunity to provide their opinions on the student/teacher feedback generated by the tool for three synthetic pieces of Year 4 literacy work. This section was repeated three times, one for each essay.
3. **General Feedback**: the third section gave teachers the opportunity to provide general feedback on generated content, with particular interest in whether teachers found any common issues across the examples of AI-generated content.
4. **Approach to Class-wide Feedback**: the final section explored how teachers approach the process of giving class-wide feedback. Teachers reviewed a group of essays written in response to the same task in section 2, which was encouraged to be thought of as work produced by students in the same class. Teachers were then asked to offer their perspective on how they would approach giving feedback to the whole class in response to these.

Success criteria

Teachers were presented with both closed-ended and open-ended questions to assess the success rate of teacher-focused feedback, student-focused feedback, and generated tasks. By using a semi-structured method, this allowed Faculty to collect quantitative data reflecting teachers' ratings of the tool while also collecting quantitative, descriptive feedback to explain their reasoning and offer general opinions.

For quantitative questions, teachers rated synthetic student work on a scale and were presented with multiple choice. For example, when asked how useful teachers found the teacher-focused feedback, the scale ranged from 1 to 5, with 1 being ‘not useful’ and 5 being ‘extremely useful’. Here, Faculty measured feedback and task success as higher answers on the scale. When asking how appropriate teachers found the length of the teacher-focused feedback, the three options presented were ‘too short’, ‘just right’, and ‘too long’. Here, faculty measured the success of feedback length as ‘just right.’ These questions were replicated in assessing the student-focused feedback. For feedback on the generated tasks, teachers were asked how likely they would use the task to support the teaching of students. The scale ranged from 1 to 5, with 1 being ‘not likely’ and 5 being ‘likely.’ Here, Faculty again measured task success as higher teacher answers on the scale. Additionally, teachers were asked how appropriate they found the length of the task, measured in the same way as teacher and student feedback length.

General insights on the tool

Overall, teachers were impressed with the tool’s ability to support the assessment of student work. Teachers described the concept of the tool as “brilliant” and highlighted that the feedback provided was useful and informative. In addition, teachers were excited by the revision activity which the tool generated. In particular, teachers underlined how valuable the tool is in terms of time-saving for educators when reviewing student work and monitoring student development.

Generally, the main constructive feedback provided by teachers focused on future tool development. While teachers were enthusiastic about the tool’s ability to assess student work, they expressed that the tool would be more useful if it were able to take into account previous work from students. Teachers suggested that if the tool had this feature, it would be highly beneficial in identifying common or recurring errors to monitor students’ progress and pinpoint areas for improvement.

Results for feedback

Teacher-focused feedback: Positive insights

Success on usefulness

Generally, teachers found that the teacher-focused feedback developed by the tool was highly useful. Per Table 2, 72 percent of teachers rated the tool highly in usefulness, with 39 percent scoring “4” and 33 percent scoring “5”. Teachers highlighted that the feedback was clear, accurate, and informative. Teachers expressed that the feedback was helpful in identifying positive aspects of the students’ work, while also identifying fundamental errors and pinpointing areas of focus for each pupil. Additionally, teachers commented that the tool would save teachers huge amounts of time when assessing and grading student work.

Success on incorporating the National Curriculum

In addition, teachers responded enthusiastically to the tool's ability to link directly to the national curriculum. According to them, providing year group expectations is useful as it enables easy identification of knowledge gaps. For instance, one teacher said that the content of the feedback "could be used in English lessons."

Success on feedback length

Overall, teachers responded positively to the length of teacher-focused feedback. Per Table 3, 67 percent of teachers found the length to be "just right" in providing comprehensive and error-focused feedback. Teachers enthusiastically commented on the detail of the feedback; specifically, the tool's ability to identify what students need to work on.

Table 2: Responses to "How useful would you find this teacher-focused feedback?"

Rating (5 - Extremely useful; 1 - Not useful)	Number of responses (out of 18 total)	Percentage
5/5	6	33%
4/5	7	39%
3/5	4	22%
2/5	1	6%
1/5	-	-

Table 3: Responses to "How appropriate is the length of the teacher-focused feedback?"

Rating	Number of responses (out of 18 total)	Percentage
Too Long	6	33%
Just right	12	67%
Too Short	-	-

Teacher-focused feedback: Points for development

Previous work of students required

Nevertheless, teachers provided constructive feedback to improve the usefulness of the teacher-focused feedback. Approximately 28 percent of teachers rated the tool lower in usefulness, with 22 percent scoring “3” and 6 percent scoring “2”. While teachers were impressed with the tool’s ability to identify student errors and areas for improvement, some teachers expressed that the tool would be more useful if feedback was able to consider previous work from students. In doing so, they suggested that the tool would allow teachers to monitor student progress more effectively by flagging recurring mistakes made by pupils.

Additionally, teachers also requested a summary of how well students have achieved the task compared to their year group expectations as opposed to just highlighting this in the text itself (e.g. saying something like ‘good use of simple sentences to create suspense’).

Some identified errors may be simple mistakes and not necessarily areas for improvement

Although teachers were impressed with the tool’s ability to identify substantial numbers of errors, some teachers acknowledged that some errors flagged may be a simple mistake made by the student rather than a knowledge gap or required area for improvement. According to teachers, it is less useful for the tool to generate lengthy feedback on an area which a student may not need to work on extensively.

Revisions required for feedback length

While most teachers responded positively to the length of feedback, 33 percent of teachers felt that the length of the teacher-focused feedback was too long and could be a hindrance when considering that educators typically teach classes of approximately 30 pupils. Moving forward, future versions of the tool may want to include a feature where educators can choose whether they would prefer feedback to be short, medium, or long in length.

Further tool development on language and tone required

Although out of scope, as the proof-of-concept tool focused specifically on grammar, punctuation and spelling, teachers also mentioned that they would like to see further commentary on a student’s use of language and tone in order to improve feedback.

Student-focused feedback: Positive insights

Success on usefulness

Generally, teachers found the student-focused feedback developed by the tool to be useful. Per Table 4, 55 percent of teachers rated the tool highly in usefulness, with 33 percent of teachers scoring “5” and 22 percent scoring “4”. Teachers were impressed with the tool's ability to address errors accurately and relay them to the student appropriately.

Success on personalisation

Teachers were enthusiastic about how the tool personalised feedback for students. They expressed that the feedback was detailed and specifically targeted learning gaps and areas for improvement. For example, teachers felt that the use of examples was very helpful for students to understand their errors, why they were wrong, and what they were required to work on moving forward. In addition, teachers underscored the helpful structure of the feedback, highlighting that the development points were listed clearly and concisely for students to work through.

Success on feedback length

Teachers responded positively to the length of student-focused feedback. Per Table 5, 56 percent of teachers found the length to be "just right". Teachers acknowledged that the feedback did a good job of acknowledging successes and areas for improvement in a clear and descriptive manner.

Success on language and tone

Generally, teachers felt that the language and tone used in the student-focused feedback was highly beneficial. Teachers underscored the importance of relaying student errors, while also using positive language to ensure that students are not discouraged by feedback. With this in mind, teachers found the tool adopted friendly and encouraging language to highlight what pupils had succeeded in, as well as areas they could improve.

Table 4: Responses to "How useful would you find this student-focused feedback?"

Rating (5 - Extremely useful; 1 - Not useful)	Number of responses (out of 18 total)	Percentage
5/5	6	33%
4/5	4	22%
3/5	7	39%
2/5	1	6%
1/5	-	-

Table 5: Responses to "How appropriate is the length of the student-focused feedback?"

Rating	Number of responses (out of 18 total)	Percentage
Too long	8	44%
Just right	10	56%
Too short	-	-

Student-focused feedback: Points for development

More concise feedback required

Nevertheless, teachers provided constructive feedback to improve the usefulness of the student-focused feedback. Per Table 4, 44 percent of teachers felt that the length of the student-focused feedback was too lengthy. With this in mind, some teachers were of the opinion that the tool highlighted too many points for improvement which could be overwhelming and discouraging for students. Much like the teacher-focused feedback, future versions of the tool may want to include a feature where educators can choose whether they would prefer student feedback to be short, medium, or long in length.

Feedback elaboration required

While some teachers felt that the feedback was too long, it was also suggested that student-focused feedback required further elaboration in some areas. For instance, one teacher commented that it would be helpful for the tool to list more positive feedback regarding which curriculum areas had been met. In addition, teachers found that the errors flagged by the tool were not, in some instances, comprehensively explained to the student in the feedback. Essentially, teachers found this to be problematic as students may not properly understand why the error they made was wrong and may cause them to repeat such errors in the future.

Expanded scope of tool feedback required

Although out of scope, as the proof-of-concept tool focused specifically on grammar, punctuation and spelling, teachers were eager to see an expanded scope of the tool's student-feedback. Ultimately, teachers underscored the importance of the tool providing feedback on issues such as writing style and tone, as well as simply grammar, punctuation and spelling. Teachers were cautious that the focus was only on these errors with little reference to what might have been the main teaching aim of that unit e.g. formal language, use of paragraphing in a non-fiction information text, use of conjunctions to develop ideas in an argument, or use of comma after an adverbial opener to provide detail in a story.

Guidance documents suggested for feedback improvement

When asked which guidance documents could improve the tool's student-focused feedback, teachers listed the following:

- National curriculum
- EEF Teacher Feedback Guidance Report
- Internal school documents:
    - Progress documents
    - Medium Term plans
    - Writing assessment sheet
    - Writing target card
    - Year targeted statutory spelling list
    - First 200 high frequency word list
    - Outcome piece checklist used to mark English class work
    - Writing statements for each year group
    - Unit overview sheets

Results for the activity tasks

Activity Tasks: Positive insights

Success on teaching support

Teachers responded positively to the revision activity tasks that the tool generated. Per Table 6, 78 percent of teachers rated the task highly, with 56 percent scoring "5" and 22 percent scoring "4". Overall, teachers felt that the tasks would support student outcomes and facilitate student learning and development.

Success on task length

In general, teachers found that the revision activity task was appropriate in length. Per Table 7, 78 percent of teachers rated the task "just right".

Success on Personalization

Teachers liked how the tasks were specifically tailored to the student's work instead of general activities. They were impressed that the tasks responded to the errors students made and generated an activity which reflected necessary areas for improvement.

Table 6: Responses to "How likely would you be to use the above task to support your teaching of the student?"

Rating (5 - Extremely likely; 1 - Not likely)	Number of responses (out of 18 total)	Percentage
5/5	10	56%
4/5	4	22%
3/5	3	17%
2/5	-	-
1/5	1	6%

Table 7: Responses to "How appropriate is the length of the generated task?"

Rating	Number of responses (out of 18 total)	Percentage
Too long	4	22%
Just right	14	78%
Too short	-	-

Activity Tasks: Points for development

Revision Activity Suggestion

Teachers provided constructive feedback to improve the effectiveness of the revision activity task. Per Table 6, 6 percent of teachers said that it was 'not likely' that they would use the task to support their teaching of students. For example, one teacher mentioned that while they thought the activity to rewrite and correct their own sentence would benefit students, activities around identifying/correcting mistakes or rewriting another sentence would be better. In doing so, the teacher believed that correcting their own work first, and then correcting work that is not their own, would help solidify learning and show whether students have truly understood the errors that they made.

Time to complete task required

While teachers found that the revision activity task could be extremely useful, they mentioned that it would only be the case if the time to complete activities were factored into student timetables. Per Table 7, 22 percent of teachers found that the generated task was 'too long' in length. With this in mind, one teacher felt that some of the tasks could take students up to 20 minutes to complete, especially if they are having to research and correct spellings themselves or copy multiple spellings out. Considering that students will have other homework assignments, time to complete these tasks must be factored into class time or student homework time by teachers.

Guidance documents suggested for activity task improvement

When asked which guidance documents could improve the tool's revision activity task, teachers listed the following:

- National curriculum
- EEF Teacher Feedback Guidance Report
- Internal School documents:
    - Progress documents
    - Outcome piece checklist used to mark English
    - Writing statements for each year group
    - Unit overview sheets
    - Writing assessment sheet
    - Writing target card
    - English assessment documents and progression

Tool-focused user testing

Overview of the approach

During this separate user-testing phase, 5 super users were asked to review the tool and feedback generated for the bank of synthetic student work to evaluate the performance and usability of the tool. Super users were chosen from the group of teachers/educators who had attended, and taken part in, the Hackathon. This section specifically focuses on how super users interacted with general features of the tool to identify successes and areas of improvement.

General Tool: Positive insights

Success on Tool Navigation

Super users, during their independent exploration, found that the tool was easy to navigate. Super users expressed that each section was clearly formatted and produced informative and understandable content. In general, their user experiences were described as positive with super users stating that the tool was navigable, and that both pupils and teachers would be able to easily move between the required sections as necessary. The four main tabs of the tool were able to clearly guide users through the tool's intended workflow.

Success on Time-Saving

Super users were asked whether, in a situation where they were able to expose the tool to their own pieces of pupil work (handwritten or typed), they would envision it saving time and/or improving the quality of feedback. Overall, super users interviewed unanimously stated that the tool provided value and significant benefit in terms of time-saving and that, if developed further to include additional capability, the tool could prove to be invaluable to teachers. For example, the ability to not only easily identify errors made by pupils in literacy work, but to also quickly assess how those errors related to age-related expectations. Those interviewed Super users also highlighted that the feedback provided to both teachers and students was specific enough to enable identification of distinct learning gaps for teachers and students to target for improvement.

Success on Essay Upload Feature

Super users responded enthusiastically to the 'essay upload' feature which allows them to upload details of the specific task a pupil was set and the pupil's piece of work. Super users found that this feature made the tool more interactive and useful.

Success on Task Generation

The task generation functionality and structure were popular with those interviewed (with one super user referring to it as 'brilliant'). Specifically, the fact that tasks were directed towards particular types of errors within submitted essays (as opposed to every kind of error possible) was seen as fundamental for students to work towards any potential improvement.

General Tool: Points for development

More Concise Feedback Required

Nevertheless, other challenges remain in terms of improving the usability of the tool. Generally, the main constructive feedback provided was around the style in which teacher/student feedback was generated. While many super users were enthusiastic about the tool's ability to generate comprehensive and detailed feedback, some super users expressed that teacher feedback was often too lengthy or 'wordy' and could be more concisely and effectively summarised in a series of bullet points or categorised into 'strengths and weaknesses'. Moving forward, adding a customisation button so that educators can choose whether they would prefer feedback to be short, medium or long in length may be highly beneficial. Other customisations that could be beneficial include altering the tone, i.e. making it more encouraging, or format, i.e. bullet points.

Number of Errors Identified is Too Excessive

While many super users were impressed with the tool's ability to identify every single error in a student's work, some super users found that a high number of errors may be overwhelming, confusing, and discouraging for a primary school student. This is especially true when considering SEN students, for example dyslexic students. Consequently, some super users suggested that focusing on the fundamental errors (upper limit of 5) would be more beneficial for students' learning. Again, adding a customisation button so that educators can choose whether they would prefer all errors or fundamental errors to be highlighted may be beneficial. Alternatively, the tool's ability to identify errors was designed for the benefit of the educator to understand where the student is in his/her development. Future versions of the tool may instead focus on revision tasks which require the student to identify their own errors (as opposed to being shown where they went wrong).

Optical Character Recognition Required

While super users were impressed with the tool's 'essay upload' feature, it was highlighted several times that Year 4 pupils generally handwrite their work, rather than typing it out on a computer. Currently, the tool requires educators to manually input the student's work into a textbox. Super users felt that this approach would be time consuming, especially if student work is particularly lengthy. It would therefore be extremely helpful for educators if the tool was able to automatically transcribe students' handwritten work into a textbox via an image. This technology is known as Optical Character Recognition (OCR). Adding OCR capabilities to the upload feature, enabling a teacher to just take a picture of the work, may substantially increase the usability of the tool. Specifically, an educator would scan or take a photograph of the student's essay, the educator would then upload the scanned document/photograph into the tool application, and the OCR would process the image and extract the text. The text would then be displayed and processed to provide feedback in the same way as the tool works currently. It is worth noting this might be less of a requirement if the tool had been focused on secondary school pupils, who are more likely to type their work.

Feedback Readability Tweaks Required

The tool displays an assessment of the readability of the generated feedback, using the Flesch-Kincaid Reading Score. Despite this, some super users highlighted that feedback, at points, did not align with the student's reading level. For instance, the inclusion of certain technical terminology (i.e. fronted adverbials) was too advanced for a Year 4 student to understand. In other instances, some students may read at a much younger level due to various learning needs/difficulties. In addition, student feedback was described by some super users as ‘repetitive at times’ in terms of tone (i.e. always beginning with ‘Hey super star!’ or derivatives thereof). Overall, teachers highlighted the salience of appropriate readability levels for learning and development. As a next step, it may be highly beneficial to include a feature which gives educators the opportunity to input the reading-age of their students prior to feedback generation.

Error detection

How the tool detects errors in student work

At a high level, to detect errors in a piece of work the tool uses a combination of the open-source proof-reading software package LanguageTool, and the GPT variety of large language models (LLMs). LanguageTool uses a deterministic approach to identifying and correcting errors, allowing for greater consistency and explainability of its outputs, but means it is unable to deal with more nuanced and context-dependent mistakes. For example, LT can easily pick up a misspelling of “environment” to “enviroment”, however, struggles to pick out on words that are spelled correctly but used in the wrong context e.g. “Their going to the park” should be “They're going to the park”. LLMs are able to analyse the text as a whole, gaining that contextual understanding[3] needed to spot these more complex mistakes, but introduce uncertainty into the pipeline being probabilistic in nature. By using a combination of the two, the pipeline can capture as many obvious mistakes as possible deterministically, before using GPT to capture the more nuanced, context-dependent errors. Not only does this improve the robustness of the pipeline, but it has benefits for processing cost and time.

The error detection process combines the LT and GPT in a series of processing steps to identify and categorise errors. Below is a detailed technical description of each step in the pipeline, which are also outlined in Figure 7:

1. **LanguageTool error retrieval:** The pipeline leverages a self-hosted LanguageTool server, deployed on the AWS cloud service and accessed through an API[4]. The essay text is passed to the LT server, which performs a comprehensive check against numerous linguistic rules and returns error messages containing the details of each issue detected within the text. In addition to detecting the errors, LT also suggests a correction (see Step 4 below for more detail).
2. **Proper noun filtering:** Once LT's analysis is complete, the pipeline proceeds to filter out errors associated with proper nouns, stopping them from being erroneously flagged as mistakes. This is accomplished using a range of Named Entity Recognition (NER) techniques[5].
3. **LT error classification:** The errors detected by LT are then mapped to corresponding error types defined in the National Curriculum (NC). This is done in three steps:
    a. **LT lookup table:** Initially, a lookup table (LUT) maps the LanguageTool error codes directly to error types in the NC, ensuring a clear link between the

[3] It is well documented that LLMs can “understand” and generate text based on context.
[4] An API, or Application Programming Interface, is a set of rules, protocols, and tools for building software and applications. It allows different software programs to communicate with each other, in this case our app and LT/GPT.
[5] Found in the spaCy NER library.

detected errors and educational standards. The entries in the LUT can be seen in Table 8

b. **LT rules:** For errors not mapped by the LUT, a set of predefined rules is applied. These rules are based on the NC's class definitions and enable a deterministic approach to error classification. If the LUT does not contain a direct mapping, the LT rule ID associated with each error is matched against the custom set of rules, enabling the pipeline to further categorise them. For instance, the "CONFUSION_RULE" ID, triggers the tool to check for any of the known error subtypes involving homophones and classify accordingly.

c. **GPT classification of LT errors:** Errors that are not resolved by either the LUT or rule-based methods are then classified using GPT, accessed via the OpenAI API service. These errors are processed simultaneously by GPT, which attempts to categorise them based on context and the list of NC error definitions. The GPT models are only used to classify error types which LT accurately identify 100% of the time.

4. **Error correction:** The errors identified by LT are then corrected in the original essay text. If LT suggests only one possible correction, we use it directly. If more than one (or no) correction is suggested, we query GPT to choose the most appropriate one. E.g. In the sentence: “We baked lots of caks and ate them all!”, LT will detect the misspelling and might suggest, "cakes" and "cats" as potential corrections. In this case, we use GPT to select the more appropriate choice (“cakes”). This highlights the importance of pairing deterministic rule-based tools like LT with more context-sensitive LLMs.

5. **GPT error detection:** After the LT-detected errors are corrected, the GPT model is prompted to detect any additional errors. By performing this step after the previous round of corrections, the LLM only focuses on errors that LT missed. The model’s advanced understanding of language and context is therefore used in a complementary way to LT’s deterministic approach.

6. **Token information collection:** Throughout the error detection process, token information for all queries is collated. This data is crucial for evaluating the efficiency and effectiveness of the error detection operations.

7. **Aggregation of classified errors:** The final step in the process involves compiling all classified errors, from both LT and GPT sources, into a comprehensive list. This list represents the totality of errors detected in the student's essay and forms the basis for feedback and further instructional action.

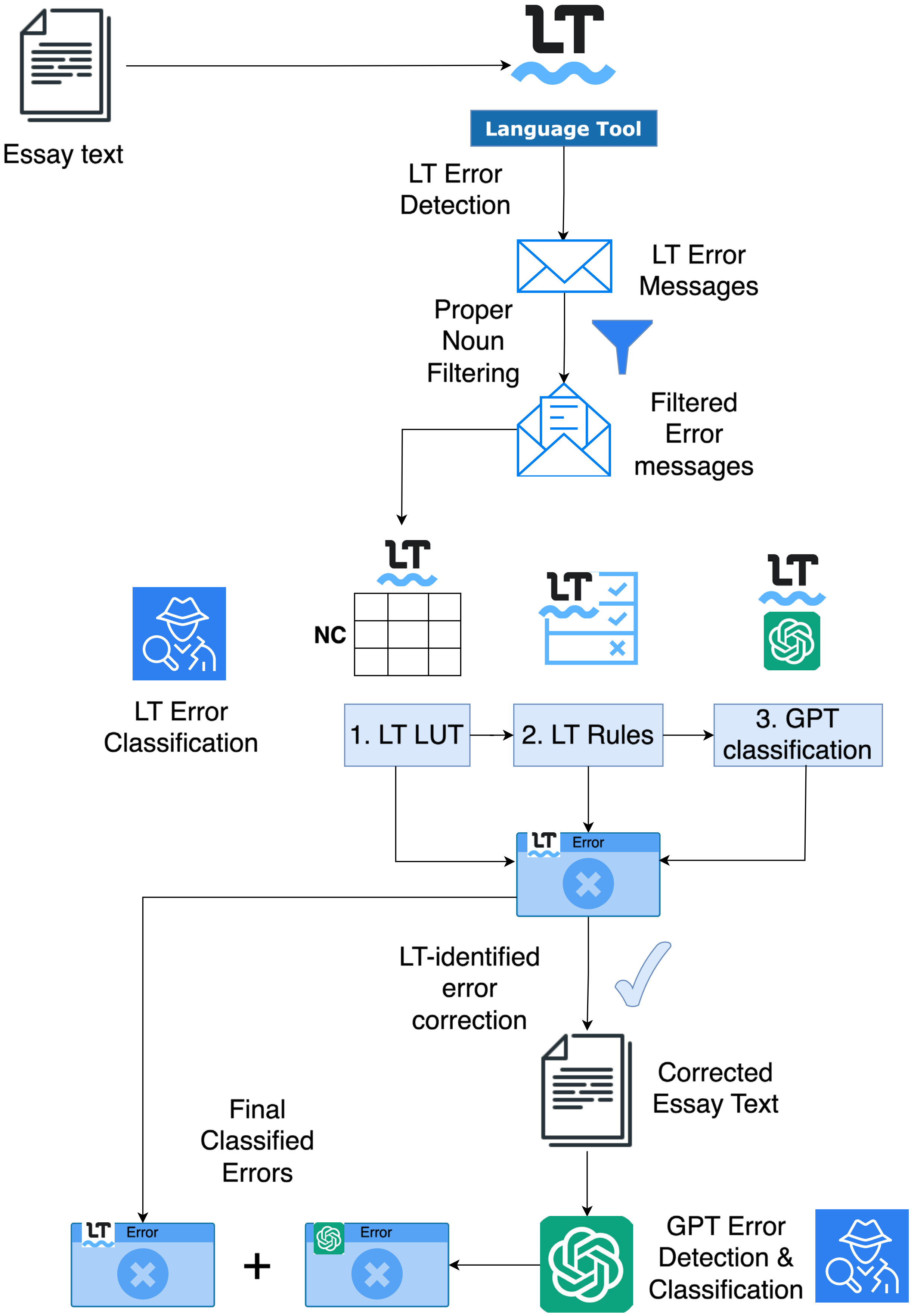


Figure 7: Detailed overview of the error detection pipeline

Table 8: Look-up-table to map the outputs of LT to NC-defined errors

LT Error Name	NC Error Definition
ONE_PLURAL	Plural
NO_KNOW	Homophones Year 4
THEIR_IS	Homophones Year 2
UPPERCASE_SENTENCE_START	Capital Letter
I_LOWERCASE	Capital Letter
SENT_START_CONJUNCTIVE_LINKING_ADVERB_COMMA	Fronted Adverbial Comma
COMMA_COMPOUND_SENTENCE	Fronted Adverbial Comma
EN_A_VS_AN	A Vs An
MANY_NN	Plural
MD_BASEFORM	Tense

The output of the error detection stage is a set of carefully structured information which details each of the detected errors. For example, a short essay taken from the synthetic pupil essay dataset can be seen below, with errors detected by the error detection pipeline underlined.

> *Sneaking Downstairs*
>
> *i lay under the cover, staring at the ceiling, my stomack empty. Slowly I got out of bed and crept towards my dore. the handle shimmered in the darkness, urging me to turn it. My hand quivered as the brass handle turned and made a 'click'. I jumped. Shadows crept across the landing while I nibbled at my nails. my parent's room's dore creaked, and I bolted down the stares, including the seventh one that makes an earsplitting thud when you step on it. I stared at the human-eating fridge, and my legs turned to jelly as I tiptoed towards it. I reached out...*

Below is an extract of the information created by the pipeline for the above essay, showing the output for the first four detected errors.


```
"error_type": "Punctuation",
"error_subtype": "Capital Letter",
```

"description": "Correctly using capital letters at the start of sentences and for proper nouns",
"year": 1,
"locator": "i",
"correction": "I",
"offset": 22,
"length": 1,
"error_detection_method": "LT_LUT"

"error_type": "Punctuation",
"error_subtype": "Fronted Adverbial Comma",
"description": "A comma is used after a fronted adverbial. Fronted adverbials are words or phrases placed at the beginning of a sentence which are used to describe the action that follows. A fronted adverbial only goes at the start of a sentence.",
"year": 4,
"locator": "staring",
"correction": "staring,",
"offset": 45,
"length": 7,
"error_detection_method": "GPT"

"error_type": "Spelling",
"error_subtype": "Common Misspelling Year 4",
"description": "A common word that is often misspelled for year 4 students.",
"year": 4,
"locator": "stomack",
"correction": "stomach",
"offset": 72,
"length": 7,
"error_detection_method": "LT_GPT"

"error_type": "Spelling",
"error_subtype": "Common Exception Year 2",
"description": "Words that are common, but break simple spelling rules.",
"year": 2,
"locator": "dore",
"correction": "door",
"offset": 132,
"length": 4,
"error_detection_method": "LT_RULE"

This information includes the error type (spelling, punctuation, or grammar) and subtype (the National Curriculum definition), as well as a description of the error subtype. The school year that the student is expected to learn that skill is also recorded. The "locator" shows the detected error and may include additional context from the essay; the corrected version is also given. The “offset” and “length” are numerical information used to situate the error within the essay; the offset is the number of characters into the essay before the error starts, and the length is the number of characters that comprises the error. Lastly, the method of error detection and classification is recorded, which can be one of “LT_LUT”, “LT_RULES”, “LT_GPT”, or “GPT”. The first three refer to errors detected with LT, and classified in one of the three ways described above, while “GPT” refers to error detection and classification performed by GPT.

Codification of the National Curriculum

In order to effectively map the LT- and GPT-detected errors back to the National Curriculum, it was necessary to programmatically encode the different requirements listed in the NC, a process known as ‘codification’. This approach was found to be much more performant than the alternatives of supplying the full free-form text via the prompt or using Retrieval-Augmented Generation (RAG). There were two key reasons this codification was necessary. When the full NC text was provided to the GPT models, they were very often not able to determine a mapping, and if they did, it was almost always wrong or inconsistent. This approach was also vastly more costly, and when compared to the codified approach, increased the cost by roughly 7x.

While a RAG approach did reduce the amount of the NC text supplied to GPT, and hence the cost, it was found to be inappropriate in this context. The text retrieved from the NC via this approach would often be unrelated to the detected errors. This is because it was unlikely the exact errors detected would be described in the NC text, and hence the retrieval would be based on the context of the essay, not the type of errors made.

The codification involves taking each of the statutory requirements listed in the NC, and create a Python object that includes:

- the type of error (Spelling, Punctuation or Grammar),
- the subtype of error e.g. Spelling error → Homophones (“accept/except”),
- the year at which this material is expected to be covered (Year 1, 2, 3 or 4),
- a brief description of the NC requirement e.g. (“Do students use the plural versions of words correctly?”), and
- examples of correct and incorrect use of the particular NC requirement.

It’s important to note that a selection of grammar, punctuation and spelling requirements for Years 1 to 4 were codified as, although the tool is ultimately targeted at Year 4 English students, the mistakes they make can also be more typical of Years 1-3. The requirements incorporated into the tool do not constitute an extensive codification of the entire NC, and it may be valuable for future work to conduct a more thorough codification of the NC. A complete list of all the codified requirements is available in Appendix A.1, while Table 9 and the code extract below demonstrate the process for one example error subtype (Spelling Contractions).

Table 9: Example requirement from the National Curriculum used for codification

Statutory requirements	Rules and guidance (non-statutory)	Example words (non-statutory)
Contractions	In contraction, the apostrophe shows where the letter or letters would be if the words were written in full (e.g. can't – cannot). It's means it is (e.g. It's raining) or sometimes it has (e.g. It's been raining), but it's is never used for the possessive.	can't, didn't, hasn't, couldn't, it's, I'll

```
contractions = SpellingError(
    Error_subclass=SpellingErrorSubClass.CONTRACTIONS,
    year=2,
    description="""Multiple words are combined into one word, with an
                apostrophe used to show where the words or letters
                would be be if the words were written in full.""",
    examples=["can't", "didn't", "hasn't", "couldn't", "it's", "I'll"],
    error_examples=[
      ErrorExample(
            error="I told my teacher that I willn't skip my homework",
            correction="I told my teacher that I won't skip my homework",
      ),
    ],
    gpt_detection=False,
)
```

Assessing the detection performance with synthetic data

To accurately evaluate the performance of the error detection logic as described above, it was necessary to obtain a ground-truth dataset containing detailed information on every error including its National Curriculum type, subtype, location in the text and correction. Such a dataset was not identified during the project. Faculty therefore partially synthesised a dataset meeting these requirements by programmatically inserting known errors into existing student work. This was achieved by first manually correcting exemplar essays, before known errors were then artificially incorporated into the error-free text. The approach meant the resulting essays still retained some level of realism, being originally written by Year 4 students, while also providing the detailed meta-data around the errors needed to perform the evaluation.

An alternative approach would have been to analyse the existing essays, with their inherent errors, and categorise these errors manually. This would ostensibly yield a dataset more representative of the common mistakes made by students but was determined to be too labour intensive and difficult to ensure both consistency in the corrections and coverage of a broad enough range of errors (both in type and context). By generating the errors, this strategy afforded the flexibility to simulate a diverse array of error types—across spelling, punctuation, and grammar—creating a more controlled and versatile testing environment.

The data generation process, shown in Figure 8, had three distinct steps:

1. Collate and manually correct KS2 example essays to create a database of error free essay text,
2. Insert errors defined using the National Curriculum into essays,
3. Create database of essays with errors, which for every error includes information about the:
    a. National Curriculum type & subtype
    b. Precise location of the error in the text
    c. Correction

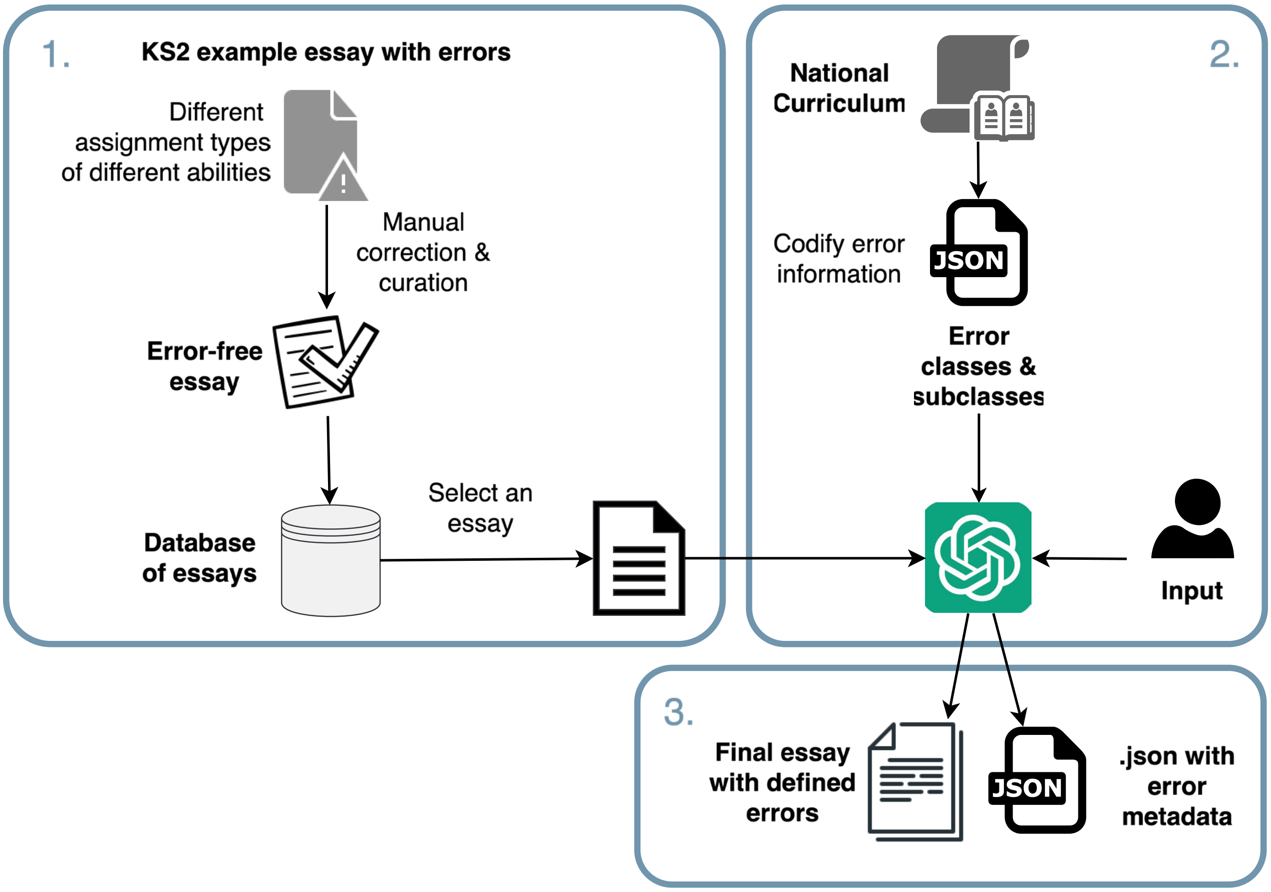


Figure 8: Pipeline for generating essays with synthetic errors

The base Key Stage 2 (KS2) essays were sourced from the Teacher assessment exemplification: KS2 English writing GOV.UK page. Once the texts were corrected manually, these two sources provided eleven essays from which to create the synthetic test dataset. It is worth noting that the process for inserting errors from the NC into the essays is essentially the error detection pipeline in reverse. Both pipelines were developed simultaneously, and the learnings from one often informed the development of the other.

While the bank of synthetic exemplar essays provided a good measure of overall performance of the tool, it did not allow for a more detailed analysis of incorrect or missed detections. This was primarily due to complexities in programmatically matching and aligning detected errors with the ground-truth error set, especially when multiple errors were located in the same sentence. It therefore became necessary to develop two additional datasets, enabling a more fine-grained assessment of the pipeline's performance.

The first of these consists of individual sentences, each containing a single error. This was primarily focused at enabling a more accurate assessment of the indexing of the errors, without the distraction of additional errors. GPT was used to generate the individual sentences, incorporating a single error from the codified set of National Curriculum requirements. The errors were chosen to give a complete coverage of codified requirements. The second dataset took 20 of these individual sentences and appended them to create a set of 20-sentence essays, with each sentence containing a single error. These essays could then be used to test how the indexing scaled across longer-texts, and check that the classification of errors was robust to changes in context and error-type. These two datasets provided a sufficient number of accurately measurable testing scenarios to rapidly test and improve the tool during development, leaving the bank of synthetic pupil essays to be used for validation and demonstration purposes.

The evaluation datasets used to assess this work, therefore consisted of:

1. **Individual sentence dataset:** 47 individual sentences, each containing a single, well-defined error,
2. **20-sentence essay dataset:** 40 essays comprising of 20 sentences with well-defined multiple errors,
3. **Synthetic pupil essay dataset:** 11 exemplar pupil essays with known errors artificially incorporated.

All three datasets were ensured to contain a mix of all three error types, giving complete coverage of the NC requirements codified for this tool.

Detection performance and improvements

To assess the tool's performance at detecting errors, for each error it is critical to know whether it has been both accurately located within the text and if it has been correctly classified. Three potential outcomes can therefore be defined for each individual error detection:

1. **Error found** – The error is correctly identified in terms of its position in the text and correctly classified as one of the NC-defined error classes.
2. **Error Misdetection** – The error is identified, but either the classification is wrong (mapped to the wrong part of the NC) or the location of the error is not quite right.
3. **No Error found** – The error is not found at all.

This categorisation of outcomes not only enabled a precise quantification of the detection pipeline performance but allowed for an easier identification of which component(s) of the pipeline were connected to the misclassification. The setup allowed for continuous evaluation and improvement of the pipeline, meaning components were introduced and modified over the period of development, guided by the evaluation results. These improvements can be grouped into four distinct stages of development, defined as:

1. **Initial pipeline (LT only)** – where only the LanguageTool package was used,
2. **Added GPT-based detection** – LT and the GPT models were used in combination,
3. **Improved NC mapping** – changes to process of mapping LT-detected errors to the NC, and
4. **Full pipeline improvements** – a range of nuanced improvements across the pipeline.

Each one of these stages is described in more detail below, while an overview of how these developments improved performance is summarised by both Figure 9, showing the breakdown of the three types of errors (Error found, Error Misdetection, No Error found) for each stage, and Table 10 which provides a timeline for these developments.

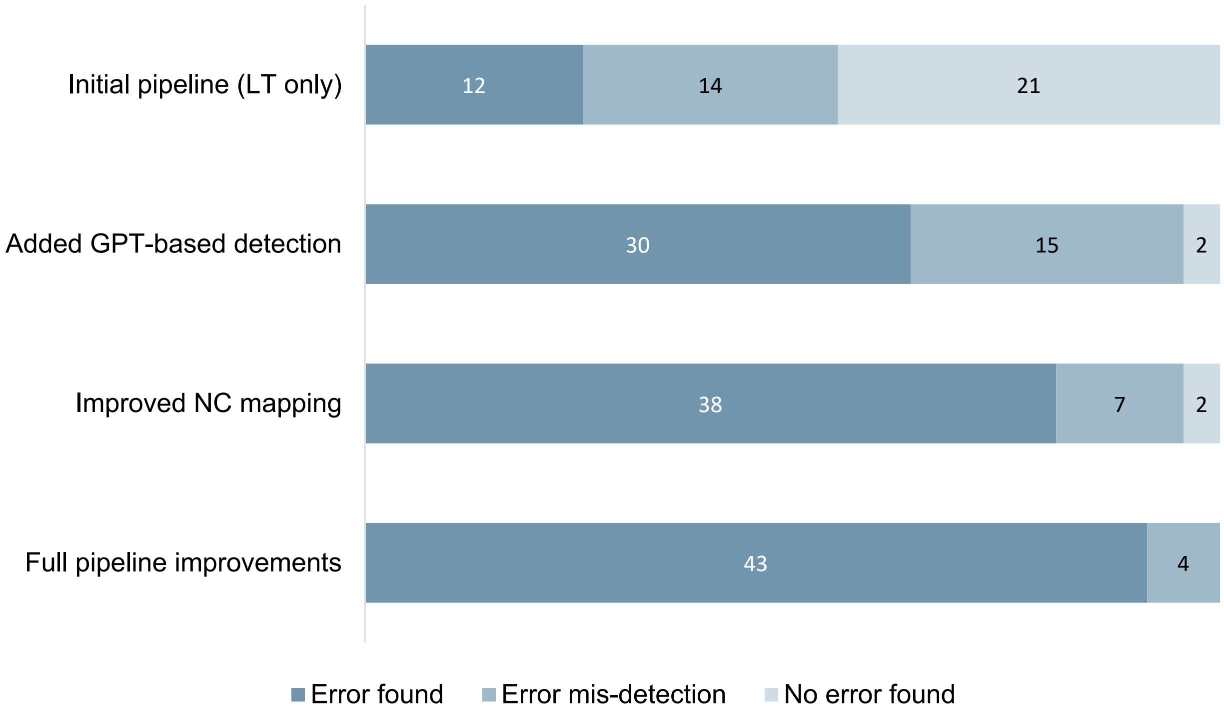


Figure 9: Performance breakdown by development stage, assessed against the individual sentence dataset

Table 10: Timeline of improvements to pipeline detection accuracy (% of errors classified as "Error Found") assessed against the individual sentence dataset

Pipeline improvement	Detection accuracy	Date implemented
Initial pipeline (LT only)	33%	22/12/23
Adding GPT-based detection	67%	08/01/24
Improved NC mapping	85%	18/01/24
Full pipeline improvements	92%	26/01/24

It should also be noted that all the results presented in this section are based on using the individual sentence dataset only. Results for the final detection pipeline on the 20-sentence essay and synthetic pupil essay datasets are given in the Appendix A.2. The performance on these more complex datasets is slightly degraded compared to the single error sentence dataset due to the presence of multiple, potentially overlapping errors and larger text length.

Initial detection pipeline (LT only)

Initially, we started by establishing a baseline of performance. This consisted of using only LT to detect the errors, without using the GPT models for any further error detection. Once detected by LT, the errors were then mapped to the NC using the three approaches referenced above: via a look-up-table using a set of custom rules and prompting GPT to classify the errors.

This method correctly classified 12/47 error sentences (~26%, see Figure 9). A larger number of errors were mis-detected, meaning that either the wrong error type was returned, or the location of the error in the sentence was incorrect. Just under half of errors, 21/47 (~45%), were not detected at all. A more detailed breakdown can be found in Figure 10, showing the contribution of the three mapping approaches. There are two ways a detected error can be classified as a misdetection; correctly located but incorrectly mapped (Incorrect Subtype), incorrectly located but correctly mapped (Incorrect Index). Clearly the deterministic rules resulted in the most misdetections, the majority of which were the result of issues with locating the errors in the text. The LUT incorrectly mapped errors in 50% of cases, mostly due to misclassifying the detected errors, while the GPT mapping was only used in two cases, and so the results don't provide much insight into its performance.

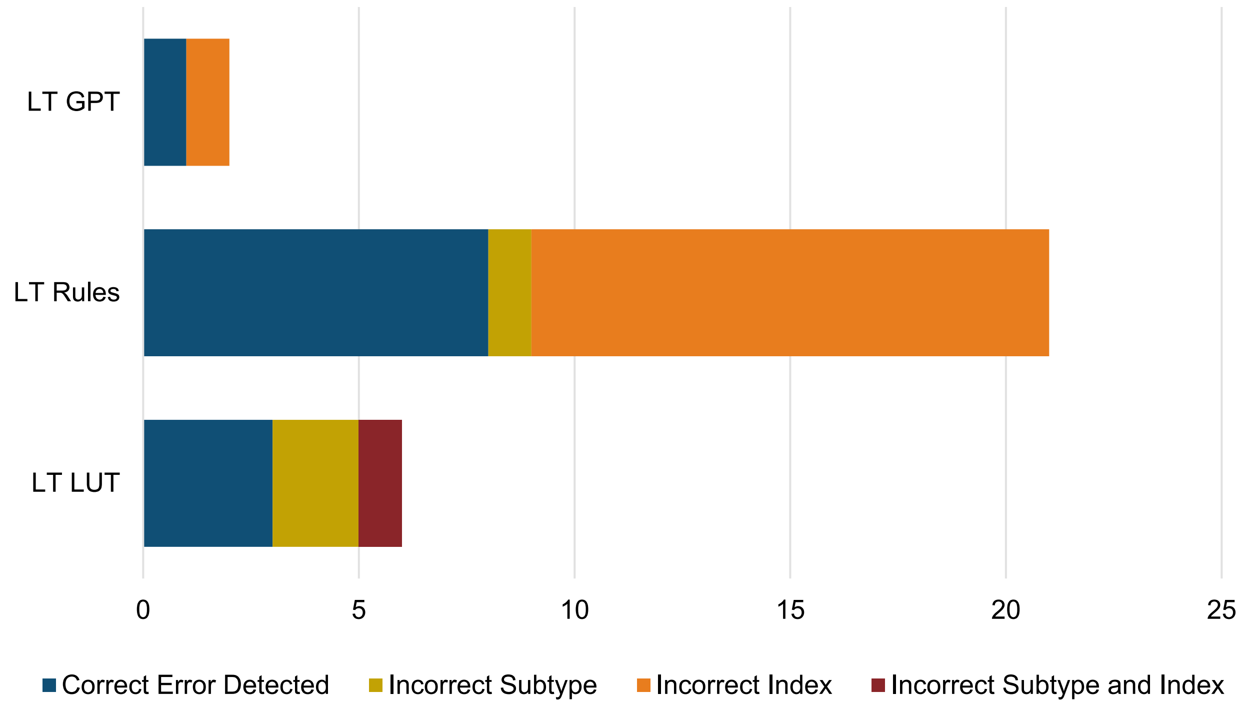


Figure 10: Count of errors correctly or incorrectly detected by the LT-only approach, assessed against the individual sentence dataset

The largest proportion of misdetections had issues with the indexing, the position of the error within the text. For example, in the sentence "The quick brown fox jumps over the lazy dog," the word fox starts at the 16th character position, it has a length of three characters, so its indexing information would be: index=16, length=3. To determine the index for a given error, it must be located within the essay text. This information is not easily determined, for example, in the sentence “I didn’t think it was fair that we couldn’t go to the fair because the fair was too expensive,” contains the word “fair” three times, but in the final instance it is a misspelling of the word “fare”. To determine the index information for the detected error: “fair” → “fare” we have three options, but no programmatic way to choose the correct one. A solution to this problem is discussed in Section 5.4.4.

Adding GPT-based detection

As 45% of errors were not even detected by LanguageTool, an obvious potential improvement to the pipeline was the addition of GPT for error detection; using both detection methods together. Errors that are identified by LT are firstly corrected and replaced in the original text, before the text is passed onto GPT. In addition to correcting the text, the GPT models were guided via the prompt to look for certain types of errors that LT was known to likely miss. Given assessing an essay with GPT costs more than with LT (see Section 5.5), this ensured that GPT was only utilised for errors most likely to increase the detection performance.

The addition of GPT as a detection method significantly improves performance as illustrated by Figure 9. Correct detections increased to 30/47 (~64%), while misclassifications only slightly rose to 15/47 (~32%). Almost no errors were missed out entirely, only 2/47 (~4%). This demonstrates the power of LLMs to analyse text in a context-dependent manner, however, Figure 11 highlights how many of errors GPT misclassified were accurately located but wrongly mapped to the NC.

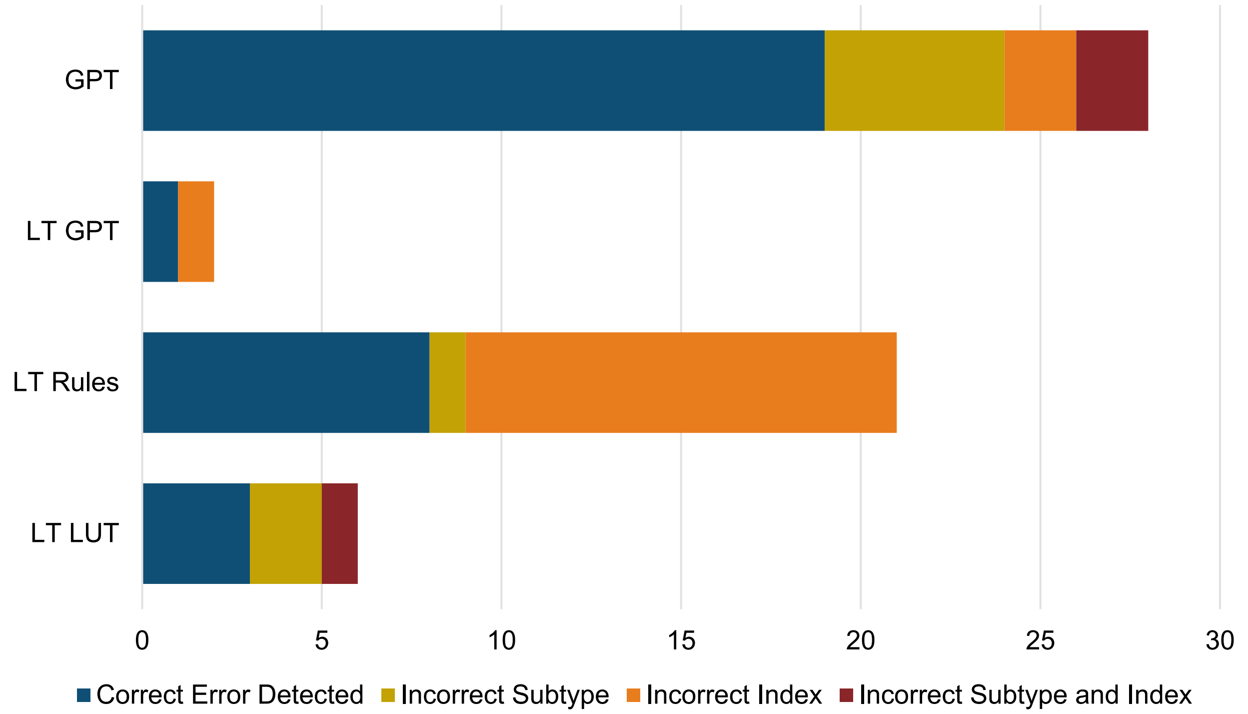


Figure 11: Count of errors correctly or incorrectly detected by the GPT-based approach, assessed against the individual sentence dataset

Improved LT to NC mapping & indexing

At this point, the largest source of errors from the detection pipeline were the result of the rules-based method of mapping detections from LanguageTool. The third stage of improvements focused on enhancing this deterministic mapping logic from the LT identified errors to the NC, including the indexing of the error within the text. Improvements to the rules took many forms, hence the below example of the logic for the use of prefixes can be used as an illustration.

> *If a prefix was used incorrectly, the suggested correction would be a change in prefix. Since a list of the prefixes pupils are expected to use exists, the tool can check whether the suggested change appears in this list of prefixes. If this is the case, then the detected error can be classified as a prefix-use error.*

Similar rules were developed for suffix, homophone, and capital letter errors. This makes the detection process for these kinds of errors more robust, deterministic (and hence simpler), and importantly, cheaper as we do not need to rely on GPT.

These changes lead to further increases in performance with the pipeline now correctly detecting 38/47 (~81%, see Figure 9) of all the errors, misclassifying 7/47 (~15%) and only missing 2/47 (~4%). It can be seen in Figure 12 that the main driver of this increase was from the rule-based mapping approach now correctly mapping 100% of errors to the relevant NC area. Despite this, some of the errors that were previously attempted to be mapped using these deterministic rules are now offloaded onto the GPT mapping method, or the GPT detection, increasing the number of mis-detected errors for both these methods.

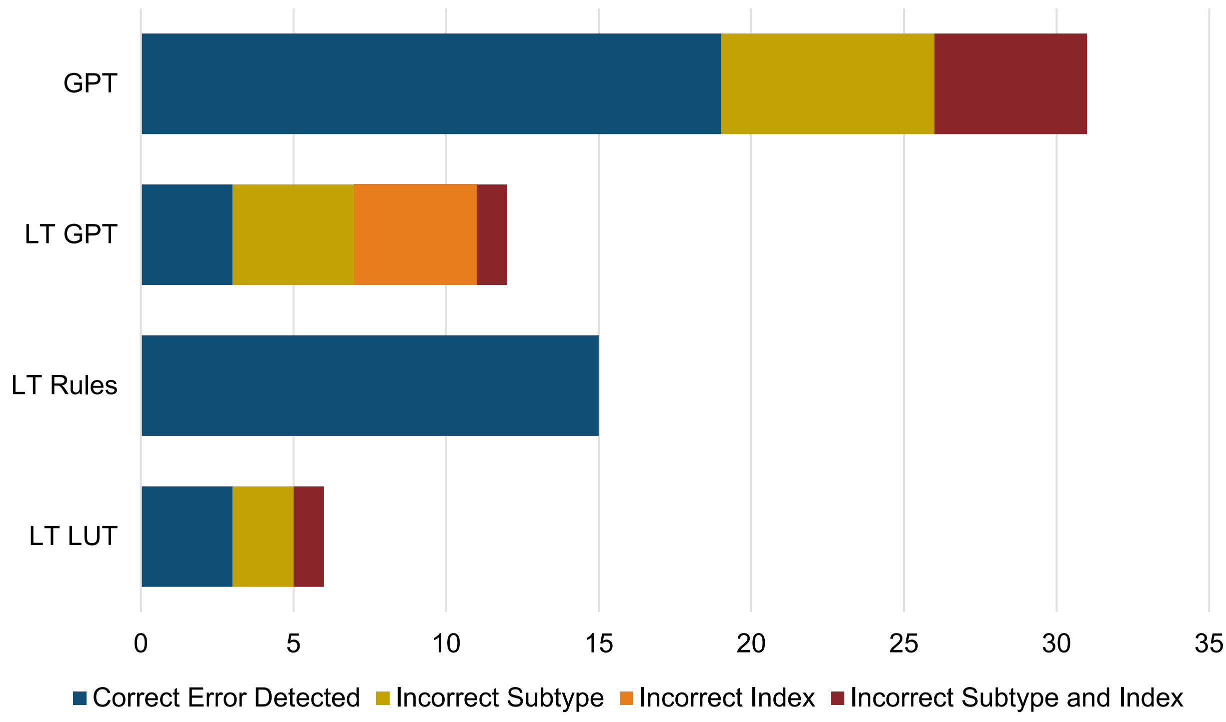


Figure 12: Count of errors correctly or incorrectly detected by the improved rule-based mapping approach, assessed against the individual sentence dataset

Full pipeline improvements

The final set of enhancements focused on error detection and classification using GPT, a pivotal aspect of which was the refinement of the error definitions to the National Curriculum (NC), alongside further indexing improvements.

It is well-documented that Large Language Models (LLMs) exhibit a heightened sensitivity to the formulation of prompts/instructions. Through a series of small-scale experiments, it became evident that elaborating on the NC definitions and enriching them with more comprehensive information and examples could yield further improvements in performance. This additional detail provides GPT with a more nuanced understanding necessary for accurately classifying complex errors.

An additional layer of refinements was introduced by implementing advanced logic for the indexing of error locations. As mentioned above, the indexing of the detected errors can be improved by providing a wider context to the error. Consider the example used previously; “I didn’t think it was fair that we couldn’t go to the fair because the fair was too expensive”. Instead of the correction being “fair” → “fare”, which is not unique in the sentence, by extending the context window around the error to include the surrounding words it is possible to make a unique pair. The error and correction pair now become “because the fair was too expensive” → “because the fare was too expensive”. Ensuring the error and its correction are unique within the sentence means calculating the indexing information becomes trivial, and effort can be shifted onto classifying the errors rather than locating them.

The resulting performance increase of the GPT classification and GPT detection methods meant the number of correctly detected errors increased to 43/47 (92%), while only 4/47 (8%) of errors were mis-detected and no errors were entirely missed (see Figure 9). Figure 13 shows that all the new correct classifications come from the GPT-based detection component, while misclassifications for both GPT-based classification and detection shift from misclassifications to indexing errors.

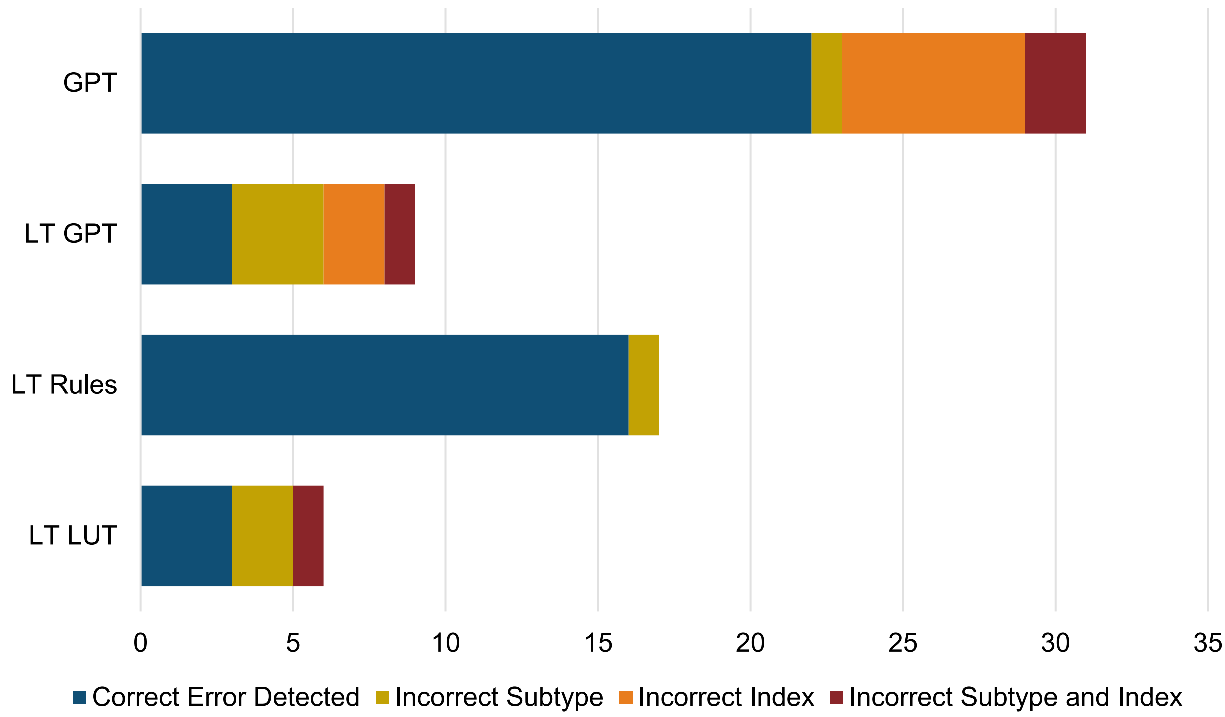


Figure 13: Count of errors correctly or incorrectly detected by the improved GPT-based classification approach, assessed against the individual sentence dataset

Cost and engineering considerations

LanguageTool hosting costs

The open-source version of LanguageTool was used for this project requiring it to be hosted on an AWS server accessible by the main components of the tool. The internal language model used by the tool comes in two different sizes, hence requiring differing amounts of resources to host the tool, each with different costs associated. The smaller model, referred to as LT Lite, costs £4.44 per month before tax to host, while a larger model, referred to as LT+, costs £80.67 per month before tax. Since the tool is set up on a dedicated server for the tool, the cost is fixed regardless of the number of times the tool is used[6].

GPT query costs

The direct cost to query GPT comes from the amount of text supplied as input to the generation process as well as the amount of text generated as output. Rather than charging per word or per sentence, OpenAI charges per token for the input and output. Tokens are used as a way to break words into smaller chunks, although some tokens can constitute entire words. The charges per token differ between different GPT models as well as if the token is an input or an output of the model[7].

The GPT query API provides a feature to return the actual input and output token count for each interaction. This information was captured within the tool pipeline to record token usage for the synthetic pupil essay dataset and was used with the above information on token prices to determine the cost of processing each one. These costs were then averaged across all the essays to provide an estimate for the cost to process a standard pupil essay; however, the true cost of any single essay will vary with both its length and the number of errors present.

Table 11 shows the average cost per essay for different parts of the error detection pipeline. Larger GPT models (GPT-4 vs GPT-3.5-turbo) are utilised for some parts of the pipeline due to the poor performance experienced when using smaller models. The costs of additional tasks and outputs performed by the tool are addressed in Section 6.1.3 (see Table 13).

Table 11: Breakdown of per essay error detection costs

GPT Use Description	GPT Model Used	Average Cost per Essay (£)
Choose the LT suggested error correction if there was more than one suggestion, or if there were no suggestions.	GPT-3.5-turbo	£0.0004
Classify LT detected errors using NC error definitions.	GPT-3.5-turbo or GPT-3.5-turbo-16k if context window was too small	£0.0098

[6] Within reasonable usage limits
[7] OpenAI's pricing: https://openai.com/pricing

GPT Use Description	GPT Model Used	Average Cost per Essay (£)
Detect errors in the essay that were not detected by LT.	GPT-4	£0.0535
Total Error Detection Cost		£0.0637

Model performance comparisons

To enable a choice of GPT model that ensured both high accuracy and cost efficiency, a comparative analysis of each model's performance was performed as part of this work. An overview of the findings is shown in Table 12, highlighting that error detection performance remains similar when comparing LT Lite and LT+, however, was observed to increase with the size of GPT model. Interestingly, the cost is more dependent on the choice of LT model, than GPT.

After extensive testing, we've found the combination of LT lite and GPT-4 to strike the best balance between accuracy and expense. This combination maintains a high accuracy rate of around 90% while being substantially more cost-effective: 20 times cheaper than alternative configurations given the expected usage rates of the tool. The choice between LT lite and LT+ models boiled down to the following cost analysis:

- **LT+ Model:** With a monthly cost of £80 and GPT-related error detection queries at £0.06 per essay, the GPT costs only equate to those of LT+ after processing more than 1300 essays.
- **LT Lite Model:** The breakeven point for LT Lite is much lower, at just 74 essays per month, making it a far more economical choice without significantly compromising on accuracy.

Table 12: Overall detection accuracy of different LT and GPT model combinations, assessed against the individual sentence dataset

GPT Model	LT Lite	LT +
No GPT	46%	52%
GPT 3.5-turbo	62%	63%
GPT 4-turbo	81%	88%

GPT Model	LT Lite	LT +
GPT 4	88%	92%

The error detection performance of the final error detection pipeline is provided in the Appendix A.2. While the absolute performance for these datasets is slightly degraded when compared to the single error sentences dataset, the general trend when comparing the performance of the different components was the same, as seen in Figure 19. This further supports the choice of LT Lite and GPT4 for the error detection.

Generation of feedback and activities

How the tool generates feedback and activities

Feedback generation

The tool generates two pieces of feedback: a piece of teacher focused feedback which is intended to be used by the teacher in a report-style text; and a piece of student focused feedback which is intended as a first draft for the teacher to assess and edit before being given to the student. In order to generate these pieces of feedback, the Open AI LLM, GPT-4-turbo, is used. The LLM is provided with the student's essay and the list of classified detected errors which are the output of the error detection pipeline described in Section 5. For example, the start of one of the synthetic pupil essays reads:

> *How Pointe Shoes Came To Be*
>
> *have you ever wondered why ballerinas look so beautiful and graceful on stage?*
>
> *…*

This would be provided to GPT, along with the following error information:


```
{
"error_type": "Punctuation",
"error_subtype": "Capital Letter",
"description": "Correctly using capital letters at the start of sentences
                and for proper nouns",
"year": 1,
"locator": "have",
"correction": "Have",
"index": 30,
"length": 4,
"error_detection_method": "LT_LUT"
}
```


The LLM is instructed to use these two pieces of information to generate both the teacher and student focused feedback respectively. The format of the provided information is described in the prompts so it can be understood and used effectively by the LLM, and specific instructions for the style and tone of the output are given depending on whether teacher or student focused feedback is requested. Continuing the above example, the part of the teacher focused feedback generated for the example error reads:

> *The most notable one is the incorrect use of capital letters at the start of sentences, a skill introduced in Year 1. For example, the student started a sentence with 'have' instead of 'Have'.*

Additional instructions were investigated for their usefulness, specifically instructions derived from guidance documents (documented in Appendix A.3), and the results of this investigation are provided below in Section 6.3.

The recently released OpenAI API feature JSON Mode, which forces the output of the LLM to be a valid JSON object, was used to increase the reliability of the output format, reducing variability that resulted in formatting errors in the tool. This necessitated the provision of additional instructions for the LLM to guide the format of the output.

After the feedback output is generated, the feedback text is extracted from the resulting JSON object and saved to a database of responses which the tool uses to display the generated outputs.

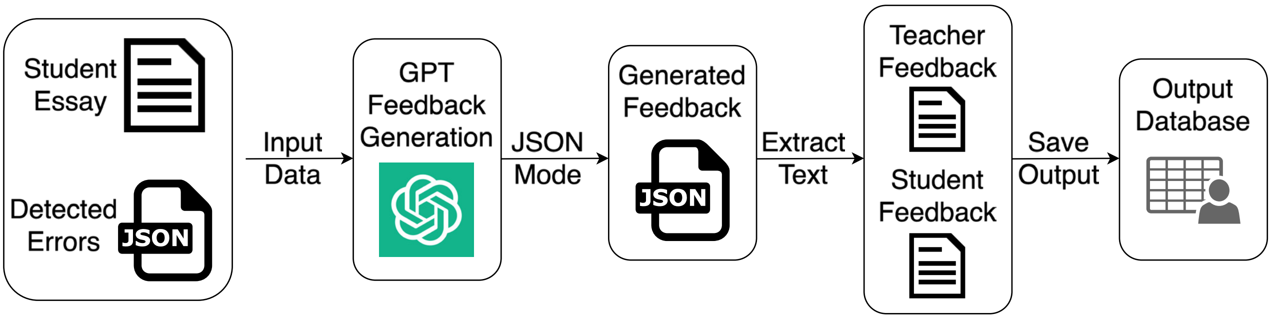


Figure 14: The feedback generation process

Task generation

The tool generates four formative worksheets consisting of a set of tasks based on the errors made by the student. Each one of the four has a different focus: one is a general set of tasks that cover all of the errors made by the student; while the other three focus on spelling, punctuation and grammar respectively.

The task generation process is very similar to the feedback generation process, with the exception that the teacher focused feedback is also provided as an input to the task generation. The LLM prompt additionally includes instructions to either generate a "General" set of tasks that should cover all the detected errors, or it is focussed onto a specific error category: "Spelling", "Punctuation", or "Grammar". The GPT-4-turbo model is used to generate the tasks, utilising the same JSON Mode mentioned above. The output formatting instructions are the same, as is the procedure for extracting the task text from the JSON object and saving the generated outputs to the database.

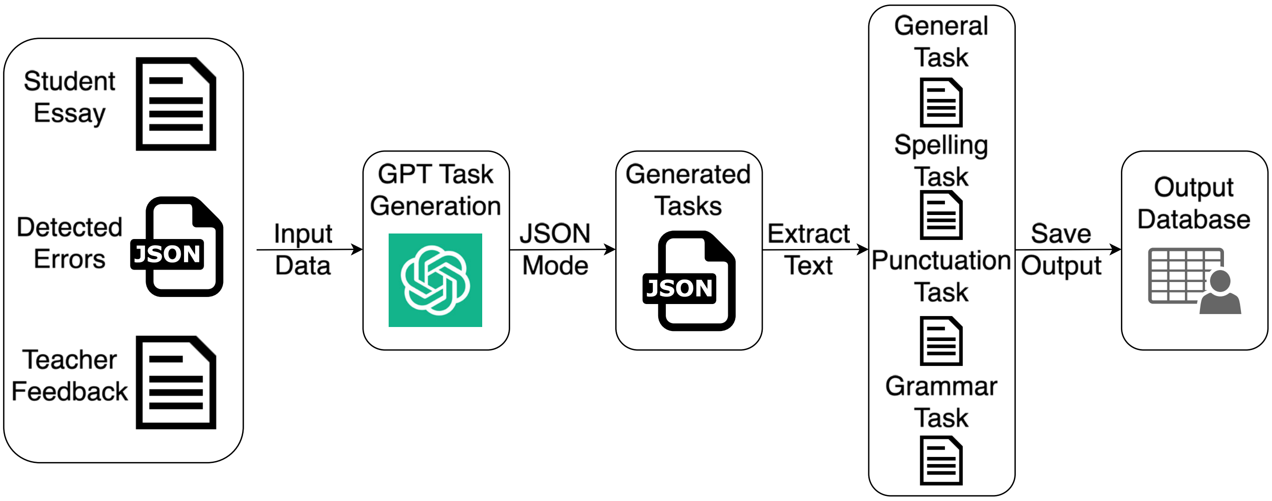


Figure 15: The task generation process

Below is an example of part of the ”General” worksheet was generated for the example essay used in the Section 6.1.1:

> *Capital Letters:*
> *Rewrite the following sentences with correct capitalization.*
> *a. have you ever wondered why ballerinas look so beautiful and graceful on stage?*
> *b. keep on reading, to find out what makes The Nutcracker you saw at christmas the magical story it is.*

Content generation costs and total tool costs

Table 13, below, shows the average cost per synthetic pupil essay to generate the two feedback texts, and the four worksheets. The previously shown costs of the error detection pipeline are also shown, so the total cost of the tool, averaged over the synthetic essays, can also be calculated.

Table 13: Breakdown of cost for an average essay per tool component

GPT Use Description	GPT Model Used	Average Cost per Essay (£)
Error detection pipeline	Multiple models	£0.0637
Generate teacher focused feedback.	GPT 4-turbo	£0.0850
Generate student focused feedback.	GPT 4-turbo	£0.0800

GPT Use Description	GPT Model Used	Average Cost per Essay (£)
Generate four worksheets: General, Spelling, Punctuation, Grammar	GPT 4-turbo	£0.1451
Total Cost		£0.3738

Assessment of performance

Evaluation process & criteria

Based on input from educators, to evaluate the quality of the generated feedback and tasks, two different methods were used. The first method used GPT to assess the content against a set of evaluation criteria. These criteria were derived from relevant sections of the guidance documents listed in Appendix A.8, as suggested by educators.

The technique of using LLMs to evaluate LLM generation is commonly used and multiple frameworks exist for this purpose. For example, GPTScore and G-Eval prompt LLMs to evaluate generated text on several metrics like coherence, relevance and conciseness. More generally, LLM-as-a-Judge has been used to evaluate output generated by LLaMA-based models by giving them an overall quality score.

The student essay, essay task, and generated output are provided to the LLM, along with the set of criteria and instructions on how to format the output. For each criterion, the LLM is instructed to score the generated output out of 100 and provide a short justification for the score it gave. These scores and justifications are returned in a JSON format, with GPT utilising the JSON Mode mentioned before. A validation of this method of evaluation is given in Section 6.2.2.

The criteria used to assess the generated output using GPT are as follows:

1. **Feedback:** Teacher and Student
    a. *Error Specific:* Does the feedback make references to specific errors made by the student in the essay?
    b. *Task Specific:* Does the feedback refer to the specifics of the essay task?
    c. *Improvement Specific:* Does the feedback provide specific and accurate guidance for how to improve?
    d. *Severe Errors:* Are more severe errors highlighted compared to more minor ones?
2. **Feedback:** Student Only

a. *Complexity*: Has complex material been broken down into smaller steps?
b. *Reflection*: Is the feedback not simply pointing out the errors but also encouraging understanding and reflection?

3. Revision tasks

a. *Error Specific*: Does the revision task focus on specific errors made by the student in the essay?
b. *Task Specific*: Does the revision task refer to the specifics of the essay task?
c. *Improvement Specific*: Does the revision task provide specific and accurate guidance for how to improve?
d. *Severe Errors*: Are more severe errors focussed on compared to more minor ones?
e. *Complexity*: Has complex material been broken down into smaller steps?
f. *Task Type Specific*: Do the tasks focus specifically on (Spelling/Punctuation/Grammar)?
g. *Task Variety*: Have a variety of tasks been generated? E.g. multiple-choice questions, fill in the blanks, rewrite sentences, etc.

The second method of evaluation uses quantitative metrics that can be directly measured using the generated output. The first of these calculated metrics is related to the readability of the generated output. The measure of readability is calculated via the Flesch reading ease score, which ranges from 0 (able to be read by a learner) to 100 (university graduate level). The score considers the average length of the sentences and the average number of syllables per word, where texts with shorter sentences and words using fewer syllables will score higher. The higher the score, the easier the text will be to read. This metric is widely used to assess the readability of graded readers or content.

Although the readability score of the generated output is a good measure of how understandable the text is, it alone is not an indicator of whether the text is at the right level for the reader, implying that the ideal score should always be 100. For example, as the teacher-focused feedback is aimed at a graduate level reader, it is acceptable for it to use technical language and have an extremely low score. Target readability scores were therefore defined. For the teacher focused feedback, this is a score of 60. For the student focused feedback, and all generated tasks, the target score is defined as the readability score of the student's essay. This means the generated outputs are dynamically assessed for the student the content is aimed at.

The readability evaluation score is calculated using this target readability score:

1. Calculate the absolute difference between the target score and the measured score.
2. If the measured score is below the target score, double the calculated distance.

    a. Meaning feedback is penalised twice as much for being of a higher-than-expected reading age.
3. Normalise the distances into scores between 0 – 1.
    a. Meaning scores that are further from the target are closer to one.
4. Subtract the score from 1.
    a. Meaning scores that are further from the target are now closer to zero.
5. Normalise from 0 – 100.

The resulting Readability evaluation score is a number between 0-100 where a score of 100 means the generated output had the exact same readability as the target. Lower readability evaluation scores are the result of generated output having readability scores that are further from the target.

The second calculated metric is a measure of the conciseness of the generated outputs. The Conciseness score is a number between 0-100 where a score of 100 means that the generated output used no more than 30 words per detected error type. As more words are used, the score falls lower and lower. By using the words per detected error type instead of the actual number of words, we can judge generated outputs that were generated using essays with differing numbers of detected errors. If an essay has a large number of detected errors, the generated outputs should not be penalised for providing more feedback or longer tasks compared to an essay with a small number of errors.

The steps for calculating this metric are as follows:

1. Calculate the number of words per detected error type in the generated output,
2. Apply the threshold after which the length of the output is penalised, in this case 30 words,
3. Apply a scaling that determines how harsh the penalisation is for words above the threshold,
4. Use an exponential function to apply the penalisation,
5. Normalise the score from 0-100.

The mathematical function encapsulating the above method can be described as:

$$100 \cdot e^{-\frac{words\ per\ error\ type - threshold}{scaling}}$$

Finally, a combined Total Score is calculated for the generated outputs. This score is calculated as the weighted average of the individual metric scores. The benefit to using a weighted average over a simple average is that some metrics can be treated as more important than others when calculating the total score. These weightings were used in the following way, in the feedback evaluation, the Readability evaluation score is weighted as twice as important than the other metrics, while all metrics are treated equally for the worksheet evaluation.

The following table, Table 14, shows the evaluation output for the teacher feedback for one of the demo essays. The output consists of a score for each metric and a justification for each score, as well as the weighted average of all scores for the feedback, as previously described in this report.

Table 14: Example evaluation output for a demo essay

Metric	Score	Justification
Error Specific	90	The feedback identifies several specific errors such as homophone confusion ('their' instead of 'there'), capitalisation at the beginning of sentences, missing full stops, incorrect use of commas, and grammatical errors ('a' vs 'an', pluralisation). However, it could have provided more examples to cover the full range of errors made.
Task Specific	70	The feedback acknowledges the essay's structure and the student's ability to present a balanced view, which aligns with the task's focus on structuring writing into paragraphs 'for' and 'against'. However, it does not specifically mention the use of words from different word classes to start sentences, a key part of the task.
Improveme nt Specific	80	The feedback provides clear guidance on areas for improvement, such as practising homophones, reinforcing basic punctuation rules, and reviewing fundamental grammar. It falls short of offering specific strategies or resources for improvement.
Severe Errors	60	While the feedback highlights both minor (homophones, capitalisation) and more severe errors (punctuation affecting sentence structure, grammatical errors impacting clarity), it does not explicitly prioritise the correction of severe errors over minor ones.
Concise	49.5	Readability score was 29.7, target was 60.0.
Readable	57.2	Concise score was 57.2. The number of words in the text was 431. 6 error subtypes were detected.
Total Score	66.5	

Validation of evaluation criteria

Consistency of evaluation scores

As the evaluation process involves the use of GPT to assess the outputs, this introduces some uncertainty into its scores. It becomes important therefore to check the scores assigned by GPT are consistent when evaluating the same piece of generated content repeatedly.

In order to measure the consistency of evaluation scores, the evaluation pipeline was run 10 times for each piece of feedback generated. These 10 separate evaluations can then be compared to determine consistency by calculating the spread across each measure. The average spread for each measure across all the teacher and student focused feedback examples, was then calculated, and can be found in Table 15.

These results show there is very little variation in the scores when the same piece of feedback is evaluated multiple times, demonstrating the consistency of the evaluation process. This suggests that the model is robust at evaluating feedback generated for different types of essays as well as essays with varying amounts of mistakes. Furthermore, low variability was observed across all the different measures, with Table 16 exemplifying this for one piece of teacher focused feedback. Evaluation results for all the teacher focused feedback can be seen in Appendix A.6.

Table 15: Average score spread for each evaluation metric

Evaluation Metric	Average Spread across all Essays
Error Specific	2.0
Task Specific	6.7
Improvement Specific	2.8
Severe Errors	4.6
Complexity	4.3
Reflection	6.1
Concise	0.0
Readable	0.0
Total Average Score	4.0

Table 16: Full set of evaluation metric scores for a single piece of teacher focused feedback

Evaluation Metric	Scores	Average	Spread
Error Specific	80, 80, 80, 80, 80, 80, 80, 80, 80, 80	80.0	0.0
Task Specific	50, 60, 50, 50, 50, 50, 50, 50, 60, 50	52.0	4.0
Improvement Specific	70, 70, 70, 70, 70, 70, 70, 70, 70, 70	70.0	0.0
Severe Errors	60, 50, 60, 60, 60, 60, 60, 60, 50, 60	58.0	4.0
Concise	63.2, 63.2, 63.2, 63.2, 63.2, 63.2, 63.2, 63.2, 63.2, 63.2	63.2	0.0
Readable	54.7, 54.7, 54.7, 54.7, 54.7, 54.7, 54.7, 54.7, 54.7, 54.7	54.7	0.0
Total Average Score	61.6, 61.6, 61.6, 61.6, 61.6, 61.6, 61.6, 61.6, 61.6, 61.6	61.6	0.0

Ability to generate a range of evaluation scores

The fact that the content evaluation scores are consistent is very important, however, in principle this consistency could be due to the evaluation always returning similar scores for any feedback, regardless of quality. The following investigation shows that the generated content evaluation is capable of generating a range of evaluation scores, and these scores align with a qualitative judgement of the content quality.

In order to investigate whether a range of evaluation scores could be generated, a single piece of teacher feedback (Appendix 9.3.1) was manually edited to reduce the quality of the feedback (Appendices 9.3.2-9.3.6) before being evaluated. Six edits were made, each of which were evaluated 10 times to monitor consistency. Table 17 shows the average score and spread for each edited version, with the decreasing evaluation scores clearly reflecting the induced decrease in the quality. These results confirm that the evaluation process is able to differentiate “bad” and “good” feedback with a continuum of scores being produced and can do so consistently.

Table 17: Average evaluation scores for incrementally degraded versions of feedback

Feedback Text Appendix Location	Average Score
A.3.1	67.2 ± 1.4
A.3.2	58.8 ± 0.8
A.3.3	58.6 ± 2.7
A.3.4	55.3 ± 2.0
A.3.5	44.7 ± 5.6
A.3.6	32.0 ± 1.2

Methods for improving performance and their impact

Description of the generation experiment pipeline

In order to explore different approaches to the feedback and task generation, an experiment pipeline was developed. This established a procedure where two versions of the generation process for feedback or tasks could be compared.

The experiment pipeline first generates a set of “baseline” outputs as defined in the experiment. Unless otherwise specified, this baseline generation is the default generation pipeline without modification. The evaluation described in Section 5.2 is performed on the baseline outputs, which are generated for all the synthetic pupil essays and each type of the generated output (teacher and student feedback, or all four sets of tasks).

Then the “experiment” outputs are generated. The process for generating these outputs is defined in Section 6.3.2 and differ from experiment to experiment. Frequently the difference between the baseline and experiment generation is a change in the LLM instructions used to generate the outputs. The same evaluation is performed for the experiment outputs for all synthetic essays.

The evaluation scores for the baseline and experiment outputs are then compared and averaged over all essays, including the total scores. The difference in the total scores averaged over all essays provides a single number with which the baseline and experiment outputs can be compared. Additional detail can be found by exploring the difference in individual metrics averaged over all essays, examining individual outputs and comparing them, or inspecting the justification for the evaluation scores for individual outputs.

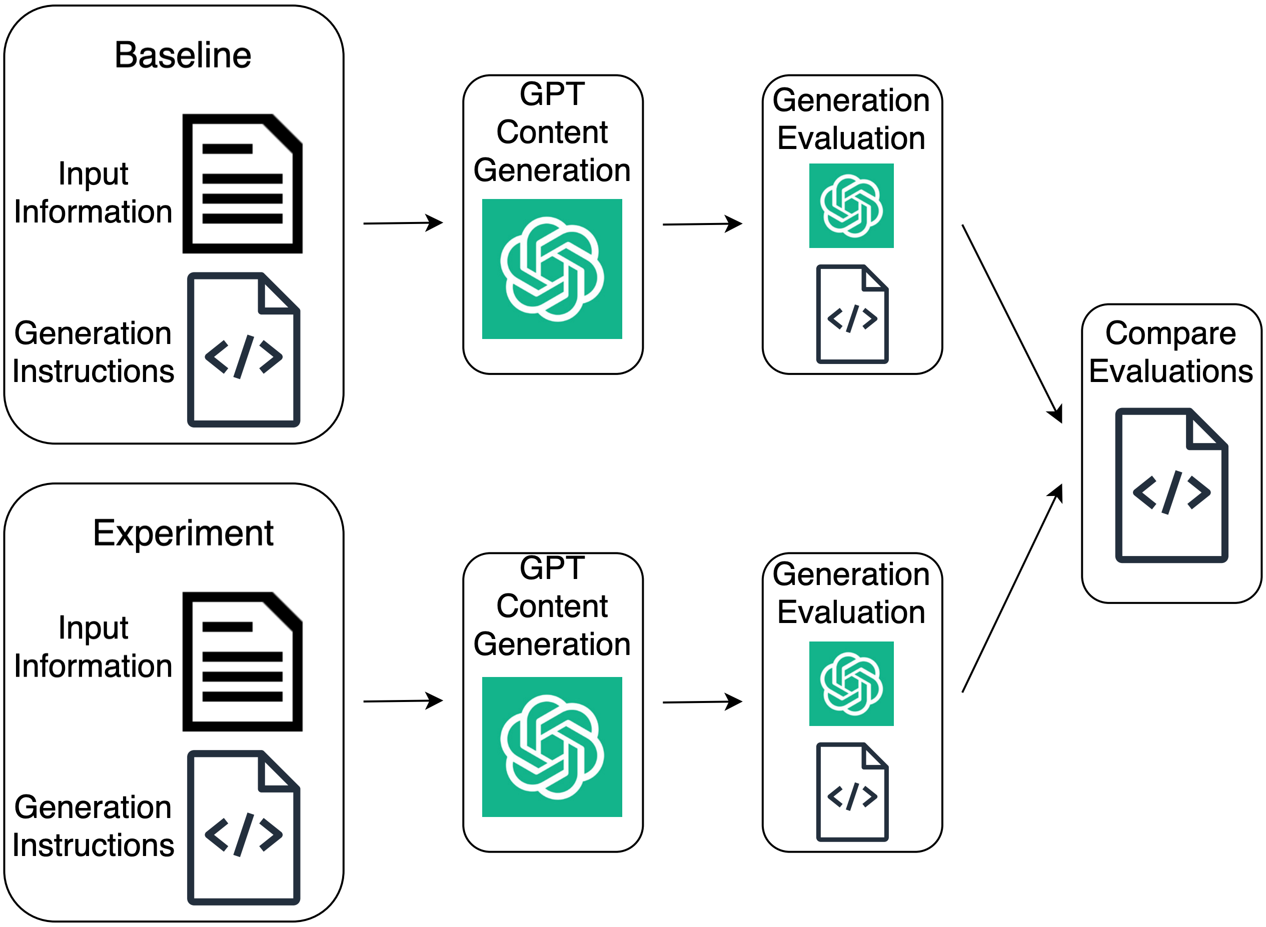


Figure 16: The experiment pipeline

List of generation experiments

Table 18: Details of the seven experiments

Experiment number(s)	Name	Description
1	Task generation with examples	How are generated tasks affected when including examples of multiple-choice questions in the task generation instructions?
2, 3	Generation with unstructured and structured prompts	How are generated feedback and tasks affected when changing the structure of the respective generation instructions?
4, 5, 6	Using guidance documents as prompt context	How is generated feedback affected if we provide additional context from guidance documents?
7	Evaluating the usefulness of parts of the feedback prompt	Which parts of the feedback generation prompt have the biggest impact on feedback generation?

Experiments details and results

Experiment 1: Tasks generation with examples

To assess the effectiveness of providing examples to the LLM when generating worksheets, six multiple choice questions were generated and included in the task generation prompt. The set of six contained two questions for each of the three error categories: spelling, punctuation, and grammar. For each, the question text had to be created by the Faculty team, before using the Oak National Academy Quiz Generator to provide the multiple choice answers. This generated four possible answers to the question, one correct answer and three incorrect. An example question can be found below:

> *Which is the correct plural possessive form of the word student in the sentence 'The students phones were confiscated'?*
> *1. The students' phones were confiscated.*
> *2. The student's phones were confiscated.*
> *3. The students's phones were confiscated.*
> *4. The student' phones were confiscated.*

The worksheets generated using the generated examples resulted in sets of tasks that only included multiple choice questions. This lowered task variety compared to the default task generation. A lower variety is believed to be an undesired effect since pupils are more likely to engage with tasks when there is a variety of types, assuming higher engagement should lead to improved educational outcomes.

Most of the evaluation measures showed changes that were smaller than the spread in those metric scores when averaged across all the essays. This was expected since the tasks generated only differ in format. The instructions for generating the tasks remained the same, where the example tasks only guided the type of questions generated. This meant that evaluation measures that focus on the content of the tasks (e.g. Do the tasks focus on specific errors made by the student?) were not altered significantly. Likewise, the worksheets have a similar overall length in terms of words, so scored similarly on the conciseness metric. The only large change was found in the Task Variety criteria, which fell from a baseline score of 54.2 ± 8.4 to an experiment score of 34.5 ± 7.5 resulting in a drop of -19.7 ± 11.2 points. This supports the qualitative assessment where a lack of task variety was observed. The total score for the tasks, averaged over all essays, fell by -3.4 ± 4.0 which is within the natural spread of the evaluation scores across the essays.

Experiments 2 and 3: Generation with unstructured and structured prompts

The way a prompt is structured may have an impact on the generation it produces. Two prompts containing the same information and instructions but in different formats may yield different results. As such, when optimising prompts, it is important to test different prompt strategies.

To test this, two experiments were conducted separately on both the worksheet and feedback generation prompts. The default worksheet generation prompt is an unstructured paragraph consisting of a series of independent instructions. The modified structured prompt adds section headers to differentiate the different parts of the prompt and modifies the instructions so they are in a list of tasks to follow format.

The worksheets generated using a free-form, unstructured prompt and those by a structured prompt were not significantly different. The task variety score did not vary between conditions, meaning that similar types of tasks were generated in both runs, and neither favoured a particular type of error more than another. All other measures remained consistent and showed no real change between the unstructured and structured prompts after accounting for the spread in scores. The length and style of the worksheets remained consistent, and scored similarly in conciseness and readability.

For the second experiment, focused on feedback, the prompts themselves are much longer and more structured than for the worksheet generation, and hence any effect should be more noticeable. The default prompt is a structured piece of text consisting of multiple sections and lists of instructions. The modified unstructured prompt converts this into a single sentence where the use of commas is avoided.

Again, the feedback generated by the default and modified prompts were not significantly different. Both feedback generations retained a consistent style and tone of voice. All scores, either averaged across both feedback types or assessing them individually, remained consistent and showed no change between the unstructured and structured prompts. Despite being a much longer prompt, no clear benefits of structuring the prompt were observed.

Overall, structuring the prompt does not lead to any improvements. However, a structured prompt is easier to maintain and debug than an unstructured prompt. As such, despite no improvements in metrics, structured prompts are preferred.

Experiments 4, 5 and 6: Using guidance documents as prompt context

While the baseline prompt to generate feedback was created using examples from the Teacher Assessment Exemplification, it was important to test whether the inclusion of guidance from additional documents may improve the output. Furthermore, given the length of the default and that it is broken in multiple sections, how this additional guidance is structured within the prompt was also a focus of these experiments.

These experiments consisted of:

- compare the generation of the default prompt with that of a prompt given additional instructions, derived from guidance documents.
- compare the generation of a prompt given additional instructions derived from guidance documents to a prompt that is given unedited excerpts from the guidance,
- compare the generation of the default prompt with that of a prompt that is given unedited excerpts from the guidance.

The feedback generated using the default prompt, additional context from the guidance documents as structured prompts, and additional context by passing quotes from the guidance documents did not vary significantly. All feedback generations had a similar structure and covered the same areas across the three conditions.

All the measures equally did not change between the baseline and modified prompts after accounting for the spread in scores. It should be noted, however, the additional guidance makes the prompts longer and therefore more costly to use. As such, it is clear there is a key level of information the model needs to perform a task week, and adding additional material does not improve its performance. Equally, when dealing with longer, unwieldy prompts, it may be possible to craft cheaper, more concise prompts without degrading performance.

Experiment 7: Evaluating the usefulness of parts of the feedback prompts

The final experiment aimed at understanding which parts of the feedback generation prompt have the biggest impact on feedback generation. The prompt itself is made up of two components: a ‘system prompt’ and a ‘user prompt’. The system prompt consists of information on the broad task that you're using the model for as well as how the model should respond, while the user prompt includes the specific example for the model to act on.

The experiment procedure first significantly pared back both the system and user prompts, before gradually adding additional instructions back and comparing the output to the baseline with each addition. This was first done for the system prompt, without adding anything to the user prompt. Then the system prompt was returned to the start and one additional component of the user prompt was added back. The series of system experiments was then repeated, and this process continued until the user prompt was fully restored.

In this way, the effectiveness of each part of both the system and user prompts could be evaluated in isolation. A diagram visualising the experiment flow can be seen in Figure 17, which was run on all 11 example essays.

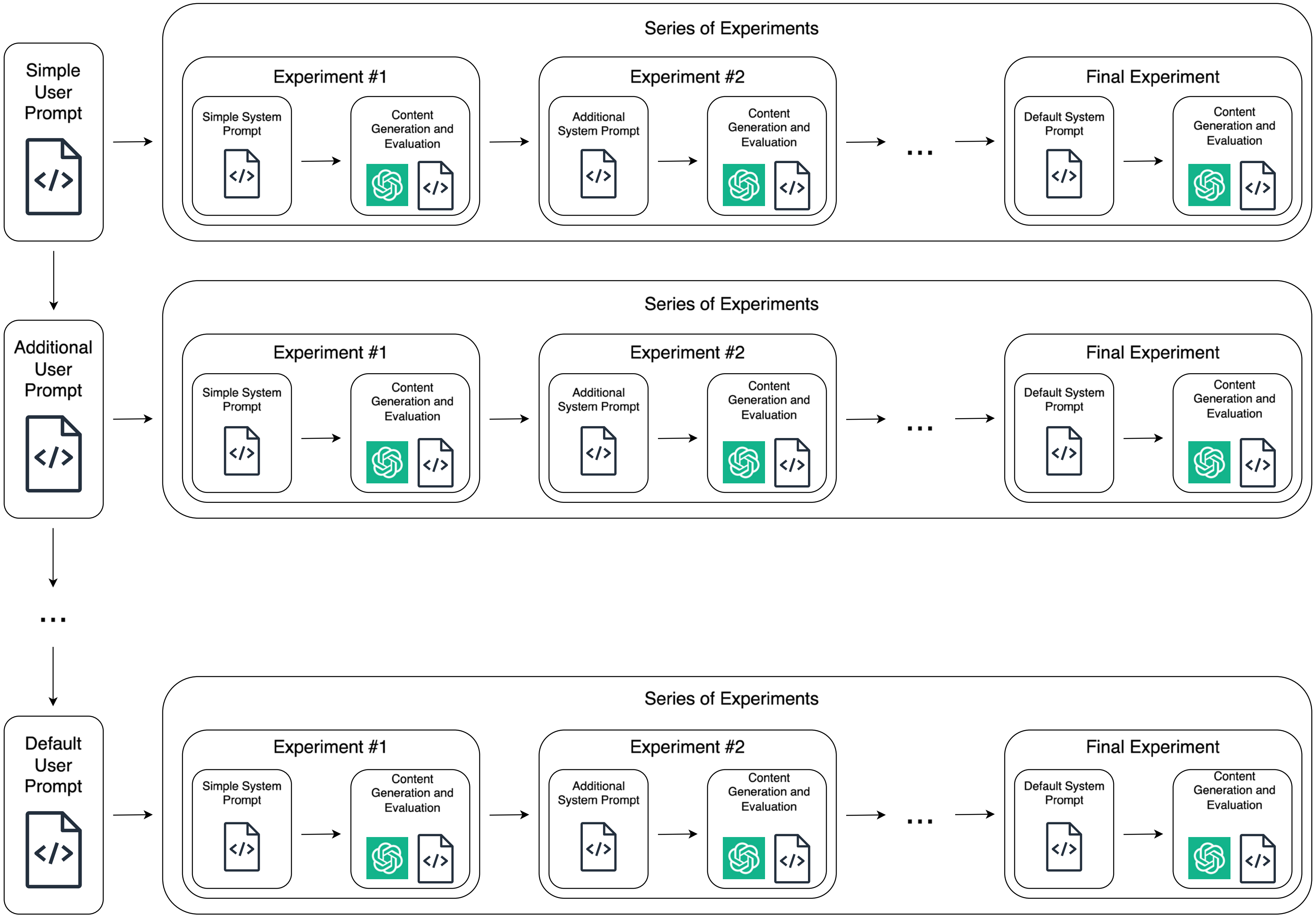


Figure 17: The prompt procedure for Experiment 7

This overarching experiment uses smaller, individual, experiments to compare the impact of including different system prompt parts, as seen in the "Series of Experiments" sections in Figure 17. On the left hand-side of the figure subsequent series of experiments are differentiated by the information included in the user prompt. A full list of the prompts used for each stage of the experiment can be found in Appendix A.5, but by way of example, the initial system and user prompts can be seen below.

Simple system prompt (as given in Appendix A.5.2):

```
CONTEXT
You are a tool used by UK Primary School teachers to generate
feedback on essays.
```

Simple user prompt (as given in Appendix A.5.1):

```
ESSAY
{A student's essay would be inserted here.}
TEACHER FEEDBACK = True
STUDENT FEEDBACK = False
```

The feedback generated using the simplest user and system prompts resulted in feedback that had evaluation scores that were similar to the default feedback generation. Inspecting the generated feedback shows that only providing GPT with the student's essay, and a general instruction to provide teacher feedback is sufficient to guide the LLM and generate good feedback. For example, the teacher feedback generated with the simple user and system prompts contained headings such as "Grammar and Punctuation", "Clarity and Structure", and "Vocabulary". The generated feedback also provided examples of errors that the student has made: *"For example, 'an long time ago' should be 'a long time ago', and 'technology mainly involves computer' should be 'technology mainly involves computers'."*

The structure of the generated feedback was not consistent between essays; however, some generated teacher feedback missing the headings mentioned above or more general feedback on the essay was given at the end of the output. Although these stylistic choices do not impact the evaluation measure particularly, it is believed that a uniform structure for the feedback is preferred, in part to aid use of the tool through consistency. Certain prompt parts do alter the structure of the generated feedback, however; predictably the prompt part that guides the LLM to include "General", "Specific", and "Summary" sections in the feedback.

These changes in structure have some impact on the individual evaluation measures. As seen in Appendix A.6, the feedback generated with all prompt parts was rated higher for referencing specific errors made by the student, due to the included guidance to this effect and the additional information on the detected errors. Conversely, the feedback generated with the simplest prompts scored higher on the Reflection measure, suggesting that the default LLM behaviour is to focus on this more than the specific errors made by the student. Almost all measures showed some level of change, but most of these changes are smaller than their spread in scores.

Comparing the initial experiment average evaluation score, 65.6 ± 2.6, which used the simple system and user prompts, to the final experiment average evaluation score, 66.6 ± 3.4, which used all system and user prompt parts defined in Appendix A.5, shows that the average scores did not change after accounting for the spread in the scores. This was a surprising result, as it was believed that the additional guidance provided in the system prompt, and the additional information in the user prompt, would result in higher quality feedback. However, GPT models are trained using extremely large datasets of text, which undoubtedly include guidance on writing good feedback, as well as examples of teacher feedback. Additionally, the data used for training extends beyond the UK education system and would encompass education systems and training materials from around the world, providing additional sources of information to train the model. It is evidenced that providing minimal guidance for the LLM is still sufficient for producing quality feedback.

Summary and insights of experiment results

These experiments show that the LLM is robust enough on its own to generate good feedback and worksheets without additional context. When given additional information about good feedback through guidance documents, the feedback structure and related measures did not change. This suggests that the model’s pre-training included enough information on how to generate feedback that meets the above criteria. As less text is therefore needed to be passed to the LLM, this can help lower the costs of the tool.

Moreover, the LLM is able to treat very unstructured instructions equally well as structured ones. Experiments 2 and 3 show that the output is not affected by the prompt structure, as long as the same instructions are included. As such, we can structure the prompts in a way comprehensible to developers in order to increase the maintainability of the tool without degrading performance.

The model’s output, however, is still very susceptible to specific prompt components. Experiment 1 shows that the model only generated 1 type of task after being given only 1 example of task type. Experiment 7 shows that certain instructions can heavily influence the structure of the generated feedback, but these mostly impact stylistic choices for the feedback, rather than improving the quality of the feedback as defined by our evaluation measures. As such, ensuring that any examples and instructions given to the LLM are representative enough of the desired output is fundamental.

Table 19: Summary of quantitative results for all experiments

No.	Name	Average overall score: Baseline	Average overall score: Experiment	Average overall score: Change	Notes
1	Tasks with Multiple Choice Question Examples	72.7 ± 3.3	69.3 ± 2.3	-3.4 ± 4.0	Change in score is largely due to decrease in task variety score (-19.7 ± 11.2) with small increases in other metrics that are smaller than the spread in the metric scores.
2	Task generation with unstructured and structured prompts	72.7 ± 3.3	72.0 ± 3.5	-0.6 ± 4.8	The LLM is robust enough to understand short unstructured information
3	Feedback generation with structured and unstructured prompts	67.0 ± 5.7	66.8 ± 3.8	-0.2 ± 6.9	The LLM is robust enough to understand long and very unstructured information. However, structured prompts are easier to maintain.
4	Using derived prompts from guideline documents to default	65.2 ± 4.3	67.3 ± 5.5	2.2 ± 7.0	The LLM pre-training produces good feedback with no need of additional information on how to generate it

No.	Name	Average overall score: Baseline	Average overall score: Experiment	Average overall score: Change	Notes
5	Using prompts derived from guidance documents to quotes from guidance documents	68.3 ± 5.7	66.8 ± 3.5	-2.4 ± 4.8	The lack of change may be caused by the LLM pre-training being robust enough to be able to generate good feedback with no additional information
6	Using quotes from guidance documents to default	67.0 ± 5.7	66.9 ± 3.5	-0.1 ± 6.7	The LLM pre-training produces good feedback with no need of additional information on how good feedback is like
7	Using prompt parts in a series of experiments	65.6 ± 2.6 (first in the series of experiments)	66.6 ± 3.4 (last in the series of experiments)	1.0 ± 4.3	Certain prompts can easily influence stylistic choices for feedback generation, but minimal instructions are needed to achieve quality feedback as defined by our evaluation criteria

For a full list of all results, see Appendix A.6.

Data privacy and IP

Objectives of the work on data privacy and IP

As outlined above, Faculty has built a proof-of-concept tool (PoC) to explore how Generative AI (GenAI) could be used to support teachers in providing feedback and suggesting associated revision activities. The tool takes a piece of Year 4 writing, assesses it against the National Curriculum, and produces personalised feedback and a revision activity based on the student's writing. To further develop the tool, Faculty aimed to explore the potential use of real-world examples of pupil work. Ultimately, the use of data to develop GenAI tools improves tool robustness, performance, and accuracy. Faculty's chief objective was to explore whether it would be possible to collect and obtain data, safely and appropriately, from willing schools/multi-academy trusts (MATs) on behalf of the Department for Education (DfE).

Considering this, Faculty explored three key principles:

1. **Intellectual Property (IP):** Student agreement was required for Faculty to obtain and use student's Intellectual Property. As primary school students are under the age of 18, the use of their student data was contingent upon parent/carer agreement. With this in mind, Faculty and DfE generated draft agreement forms which schools could disseminate to parents/carers (Annex 9).
2. **Safe transfer of student data:** Faculty explored how the safe and secure transfer of student data from schools to Faculty and the Department for Education could be facilitated.
    - **Removal of personal data:** Faculty explored how data used to develop the tool would be anonymised. Faculty explored how student anonymity could be maintained by removing student names and other personal information. The ICO offer guidance on how to achieve this.

While the project explored the feasibility of collecting pupil work with parental agreement, and a parental agreement form was developed for this purpose, no pupil work was ultimately collected via this route. The proposed collection process and draft collection form were published in order to share learnings from this process.

Key risks and parent/student/teacher perspectives

Faculty acknowledged the importance of reflecting teacher, parent, and student perspectives/concerns. When identifying schools/MATs who might be willing to share student work, Faculty explained the GenAI project comprehensively to ensure that schools understood and agreed with their involvement. Faculty explained the process and its various stages in detail (e.g. parent/carer agreement forms, anonymisation of personal information, secure transfer of data etc.) prior to schools' approval.

In addition, Faculty ensured that the GenAI project was appropriately and clearly explained to parents/carers, via the agreement wording document, to ensure that they understood what they were agreeing to. Specifically, the agreement wording document outlined the project itself, Faculty's role in obtaining data on behalf of DfE, and that all student work will be anonymised through the removal of personal information. The schools contacted by Faculty were enthusiastic about further involvement in the project – having previously taken part in Faculty's 'Hackathon' and user-testing research.

Summary of proposed approach

In order to meet requirements, Faculty and DfE engaged in planning efforts to test the feasibility of organising and managing the collection of student data to develop the GenAI tool. Fundamentally, this involved identifying schools, generating a document for parent/carer agreement, and ensuring appropriate and safe data processing and storage processes. No pupil work was ultimately collected via this route.

Identifying schools

Faculty reached out to four schools who had previously been involved in the GenAI project. To ensure that schools understood their potential further involvement, Faculty explained the GenAI project comprehensively and Faculty's reasoning behind collecting real-world examples of student work. In addition, Faculty explained the process to meet agreement, data processing, and data storage requirements. Three schools were enthusiastic about further involvement and agreed to share student work, parent/carer agreement permitting, though no pupil work was ultimately collected.

Parent/carer agreement

To explore how to obtain parent/carer agreement, Faculty and DfE generated an agreement wording document which requested parent/carer permission to share their child's schoolwork with DfE. This document was reviewed by Faculty's legal team. The Intellectual Property Office (IPO) were also consulted.

Overall, there were three main points included in the agreement wording document that would ensure that parents/carers understood what they were agreeing to. Firstly, the agreement wording document outlined the project clearly, providing parents/carers with a description of the tool. The explanation of the project, and why student data was beneficial for tool development, was designed to be accessible for parents/carers who may not be familiar with technical terminology related to Generative AI. Secondly, the agreement wording document explained DfE's partnership with Faculty. This was particularly important for parents/carers to be aware of, to ensure parents were comfortable with the agreement in the knowledge that the project was a legitimate and safe project supported by DfE. Thirdly, the agreement wording document distinctly outlined that all student work would be anonymised through the removal of student names and other identifying information. This was important to mention to ensure that parents were comfortable with the agreement in the knowledge that their child's personal information would not be used in any way. Data processing and storage

The legal approach for collecting student data involved a contract between DfE and Faculty. Faculty was responsible for collecting all agreement wording forms and student data on behalf of DfE. Faculty also was responsible for transferring all data from schools to Faculty for tool development and transferring all data to DfE, and removing student names and other identifying information from student work.

Faculty developed a step-by-step process to explore the feasibility of obtaining, processing and managing student data appropriately and safely. The following method of data collection and transferal was designed for a smaller-scale project and in some areas may require an adjusted or alternative approach for a larger-scale project.

- Schools/MATs would share agreement wording documents with parents/carers via email. Generally, email is the primary and most effective mode of communication which schools adopt to communicate with parents/carers. Considering this, schools/MATs would share the agreement wording document with parents/carers within an agreed deadline date. It is important to note that schools differ and may opt to seek agreement via other existing lines of communication (e.g. letters, texts, online school platform).
- Parents/carers would return agreement wording documents to schools/MATs. Following the deadline date, schools/MATs would receive agreement wording documents from parents/carers. Parents/carers would be made aware via the agreement form that they could retract their agreement at any point if desired. Subsequently, schools/MATs would identify which students have agreed and prepare for agreement forms and student work to be sent to DfE/Faculty.
- Schools/MATs would share agreement wording documents and student data with Faculty. There are several methods for small-scale and large-scale data collection. As a small-scale data collection project, each school/MAT could transfer the agreement wording documents and student work via email, through ZIP files. However, Faculty's preference would be to affect the transfer of data via a Secure File Transfer Protocol (SFTP) server. Once data is transferred, Faculty would upload the files securely to AWS.
- Faculty would be responsible for processing student data for tool development. Prior to data usage, Faculty would manually remove student names and any identifying information . Given the scale of the data in question, personal data would be anonymised manually by Faculty and not via software of any kind. While Faculty would seek to remove as much identifiable information as possible, it may be that the main body of student work (e.g. student answers to tasks) might include some limited personal (but not identifiable) information. For example, a student may list the first names of their parents, the city/street they grew up on etc.

- Faculty would share all agreement wording documents and student data with DfE. To ensure full transparency, and storage of data by DfE for future tool development, Faculty would share all data collected with DfE. To do so, Faculty would secure the transfer of data via a Secure File Transfer Protocol (SFTP) server.
- Faculty would delete all data collected at the end of the contracted period. Following the use of student data for tool development, Faculty would delete all student data from our system. Once student data files have been transferred to DfE, Faculty would confirm the deletion of the data with DfE.

As previously stated,while this process was designed to test feasibility of obtaining pupil work, no pupil work was ultimately collected via this route. If companies or other organisations, in partnership with schools and colleges, were to follow this pathway and are handling students' personal data, they must familiarise themselves with the ICO's guidance on anonymisation to ensure risks associated with the anonymisation process are properly managed. It is important that organisations keep up to date with technological changes to assure themselves that the risk of re-identification of data subjects are properly mitigated.

Learnings & implications

Performance against the aims of the project

As demonstrated in this report and the accompanying User Research Report, in collaboration with supporters and stakeholders from across DfE and other organisations, the project team have been able to make significant progress against the objectives of the Generative AI in Education project.

Testing the practicality of using Generative AI for predefined applications in educational settings

The User Research Report sets out in detail how the team were able to make use of the initial user engagement to identify a longlist of potential applications of GenAI in education settings, and through the Hackathons were able to deliver initial findings relating to the practicality of using GenAI to address each of these potential use cases. Once the POC had been identified, by designing and building the POC tool and then testing its functionality and outputs with users, the team were able to test how far this specific example of a GenAI tool could meet users' needs. However, as this project was limited to the development of a POC, the learnings specific to the POC tool are for the most part limited to this phase of development, and don't necessarily apply to further phases of tool development (e.g. to MVP phase or limited roll out). Specifically, testing the use of a live tool with a small group of test users inputting real data would provide invaluable further learning about how GenAI tools can be developed for and used by educational institutions.

Gathering more information about what works and the limitations of current GenAI models and approaches, and sharing information about the approaches to optimising a model

The User Research Report outlines the key findings of the Hackathons, including an account of the use cases, or applications of GenAI, that were tested during the Hackathons, the degree to which each use case was successfully addressed using GenAI, and the strengths and limitations of GenAI as applied to this use case. The in-depth user research carried out in the latter half of the project also sought to understand teachers, school leaders and students' perspectives on the potential use of GenAI tools in education, and the findings are set out in section four of the User Research Report.

Through the development of the POC tool, and specifically through the iterative process of experimentation and evaluation with error detection, feedback generation and activity generation, the team have been able to test a number of hypotheses about approaches to optimising performance of GenAI models. The findings of the experimentation work are outlined in sections five and six above, and the most widely applicable conclusions have been laid out in the following section on 'Key Learnings'. The publication of this report ensures that these findings are available for reference by the EdTech sector, but it may also be valuable to consider methods for further dissemination of these findings, as well as exploring methods to ensure that future research and learning from the development of other tools are shared with the sector and key stakeholders on an ongoing basis.

Key learnings

Learnings from building an AI-powered educational tool

Engaging educators throughout the development cycle

Crucial to the successes of the tool was the regular involvement of educators from the start of the process. Adopting an iterative approach to improving the tool through frequent feedback from a variety of user groups meant that tool could better meet the needs of teachers and students. This collaborative approach enables the AI's capabilities to be enriched with expert insights, while also building trust with users. This was also important in the evaluation of the tool, with educators inputting into the development of the evaluation criteria, ensuring the assessment was reflective of educational standards and expectations.

The importance of highlighting what a pupil has demonstrated and not just mistakes

While identifying and providing feedback on errors is useful for both educators and students, it is also important to highlight what the pupil has done well. Such information enables teachers to perform assessments of how well the child is performing against year group expectations (for example demonstrating Year 4 statutory spelling requirements). Not only this, but displaying specific positives back to the pupil is a critical part of encouraging them and reinforcing learning. Focusing only on the negatives puts this tool, and others that follow suit, at a disadvantage.

Assessing pupil work in isolation decreases the effectiveness of insights

A clear signal from educator feedback highlighted it is important for work to be assessed in both the context of other work from the pupil and, in comparison to the rest of the class or year group. Consequently, this would identify recurring errors, enabling the distinction between simple mistakes and areas for improvement. This distinction is not only important for supporting the educator in focusing on fundamental knowledge gaps, but in this case, prevents students from being overwhelmed, confused or even discouraged if too many errors are presented to them at once. Furthermore, collating these insights over a longer period allows teachers to more effectively monitor progress at both an individual and class level.

Enabling tools to be customised by and to the specific educator

Recognizing the diversity in teaching methods and educational objectives, it will be crucial to provide educators with the ability to customise tools, potentially even the generative AI prompts. This could empower teachers to tailor the AI's outputs to better match their unique teaching styles and the specific needs of their students (e.g. students with dyslexia which may need different formatting/phrasing of the feedback). While offering customisation is important, it is essential to provide educators with the guidance and support necessary to ensure it can be used safely and effectively. This includes offering templates, best practices, and examples of successful custom prompts. Additionally, implementing safeguards to maintain content quality and appropriateness is paramount, ensuring that all custom prompts align with educational standards and objectives.

Considering the wider workflow is crucial for ensuring usability

Despite teachers being able to see the inherent value in a tool, unless considerations are made for how it will integrate into their current workflow it will not gain traction with users. Specifically for this tool, the lack of OCR capability to transcribe handwritten pupil work into the tool could become a major blocker for users as it significantly decreases usability. Year 4 pupils are unlikely to complete work via a computer, and so manually entering the work into the tool would become a burden for the teacher, negating the benefits of the tool. Considering how a tool will fit in with the processes that come before and after the task the tool is designed to support can generate key learnings for the tools design, ensuring its usability.

Learnings from assessing pupil work against the National Curriculum

The most useful applications will blend deterministic and AI-based approaches

It is clear that traditional deterministic language modelling and generative AI based approaches are complementary for the task of identifying mistakes in pupils' work. Not only does this improve the accuracy of the location and classification of errors, but it acts to reduce both the time and cost of the process. Key to making the combination work, however, is understanding the strengths and weaknesses of each, and ensuring the interface(s) between them are as smooth as possible. For example, in the tool, GPT was only used to detect error types that LT couldn't. Furthermore, by using GPT to bolster the classification of LT detected errors, LT could cover a larger range of NC requirements, leaving GPT to focus on the more nuanced cases.

Model selection should consider costs as well as performance

It is possible to use models cost-effectively without compromising on performance. As shown during the project, by starting with GPT-3.5-turbo for its speed and cost-effectiveness, and escalating to more capable models like GPT-4 only when and where necessary, can maintain high accuracy while controlling costs. But the total cost of the pipeline should be considered alongside the cost per essay. When deciding on the best combination of LT and GPT versions, it becomes clear that while GPT had the highest impact on cost per essay, however, when taking into account the predicted usage of the tool, LT was actually the deciding factor due to the proportionally high hosting costs of the LT-plus server. That said, it is also important to note that this analysis would be different if the tool was to be deployed for a much wider set of users, more representative of a real usage pattern for an ed-tech tool. When designing these sorts of tools, combining cost and performance into the analysis is critical.

Generative AI is more efficient and accurate when given highly structured requirements

Although an educator is able to understand and apply detailed guidance documents, such as the National Curriculum, LLMs currently struggle with the task. Education guidance documents are often quite abstract in nature, with recommendations dependent on the situation they are applied in. This makes the document seem somewhat homogenous to a model, meaning it is difficult for it to select and comprehend the right section for the task it has been asked to complete. GPT4 was not able to determine a mapping for errors when provided with the full, unmodified version of the NC text. It was, however, able to complete the task when provided with the highly-structured codified version developed for this tool, highlighting that the issue is not in the complexity of the task but the presentation of information. Moreover, the codified version decreased processing costs by roughly 85%. High quality frameworks are therefore critical to enable LLMs in education-specific contexts, providing a bridge between their internal representations and the nuanced requirements of educational standards.

Synthetic data can provide benefits over just being a proxy for real pupil work

There is no substitute for using real examples of pupil work for both the development and testing of educational tooling, however, this is not always possible. While creating synthetic data that resembles its true counterpart is often seen as a lesser alternative, it is possible to utilise it in a way that enables insights not always possible with real data. A good example of this was the construction of the three synthetic datasets for testing the detection pipeline. By using real pupil essays as a base for these datasets, it was possible to construct test scenarios that would have been incredibly difficult to do with real data without collecting and manually assessing large quantities of data. The inserting of artificial errors allowed for the distributions of errors to be carefully controlled to ensure complete coverage. It also means detection rates in different contexts, styles or quality of language can be tested but subtly changing the errors inserted into specific sentences or essays. On top of this, another benefit is having a consistent definition and application of the corrections. If work assessed by multiple educators were used, each would apply the NC requirement slightly differently and perhaps miss (intentionally or not) errors in the text. The synthetic datasets give a much more rigorous approach to measuring performance.

Learnings from generating education-specific content with generative AI

Few-shot learning does not necessarily improve performance

Few-shot learning is an approach to designing prompts for LLMs where examples of the task being successfully completed are provided to model alongside the instructions for the specific case. When asking GPT4 to generate both feedback and worksheets, providing examples of good outputs didn't significantly alter the quality of the output generated by the model. In particular for the worksheets, the examples decreased the variety of question types, an undesired effect. While this might be the case for the Year 4 English essays, it is possible that examples become important when dealing with more senior year groups or other subjects.

While the structuring of instructions is not essential for prompts, it is useful for maintaining them

The experiments above identified that structure within a prompt is not a defining factor on the quality of the feedback or worksheets generated. Instructions can be provided in long-form prose and meander between topics; written as though they are a train of thought. While this doesn't have an impact on the model's performance, anecdotally it does make it much more difficult to develop and maintain the prompts. Much like with style guides for code[8], the use of conventions for structuring prompts improves their readability and accessibility to the human developers who have to work on them. This structuring and readability are expected to be even more important in later stages of product development (e.g. development to MVP), as the complexity of a tool grows and the likelihood of code being handed over between different developers increases.

LLMs are inherently good at providing feedback

The quality of feedback generated was equally high irrespective of the level of instruction the model was given. This demonstrates that the base performance level of LLMs on this particular task could be enough to support educators. It is believed this inherent level of quality derives from the inclusion of education-specific feedback guidance within the training data of these models, possibly sourced from across the globe. It should be caveated that this can only be said confidently for feedback on Year 4 English essays, and quality is highly likely to decrease for older year groups and other subjects. Despite the good performance out of the box, the use of clear instruction sets in the model prompts does increase the robustness of GPT at completing the task. The format and structure of the generated feedback varies greatly with limited guidance in the prompt, hence making it more difficult for an educator to consume and compare feedback for multiple pupils and may reduce their trust in the tool if the outputs are highly variable. The lack of consistency also limits the potential further processing or analysis that is then done on the generated content.

[8] For example, the PEP-8 standard for Python

Using generative AI to "mark its own homework" is an effective evaluation technique

Using the same model to both generate and assess educational content initially seems counterintuitive, however, as demonstrated through the experiments run on the tool, it is not only possible but highly effective. Developing such a technique was crucial for enabling both the iterative development of the tool and the experiment pipeline, without which the content would have had to have been assessed manually. This would have introduced inconsistencies in the assessment and delays to the work. Key to making this approach work was the creation of carefully crafted evaluation criteria covering aspects like coherence, alignment with curriculum standards, pedagogical value, and the encouragement of critical thinking and reflection. AI-generated assessments can also be paired with quantitative metrics, offering a nuanced view of the content's effectiveness.

Learnings from collecting and using real-world examples of pupil work

From the work completed as part of the Data Privacy and IP-focused activities outlined in section 7 above, we are able to draw a number of important conclusions that would be applicable to any similar work to be conducted in the future, that is, any future projects where DfE works with schools to use student or school data to inform the development of GenAI or other tools.

Teachers' perspectives

In our user research work throughout the rest of the project, there were numerous instances of students and teachers raising concerns about the way that data, and specifically student data, was used in the development of GenAI tools. Despite their views on the risks around the use of student data more generally, teachers raised no concerns about the process or the use of student data in this specific case. From this we could infer that although teachers are concerned about the broader principles and risks surrounding the use of student data, they were accepting of the use of student data for a specific GenAI tool where the benefits are clear.

Communications with parents

In working with schools to communicate with parents, it was clear that the most effective way to seek agreement from parents would be to make use of the school's existing lines of communication, using email where this is the school's primary existing mode of communication. However, this may not be the case for all schools, and it will be important to design a method of seeking agreement which is robust to other forms of communication (texts, letters, and other platforms) for a larger-scale project. In addition, significant attention was given to the drafting of the agreement form to ensure that parents, who will not always be familiar with the concepts of GenAI, data privacy, IP etc., were given enough information in a format that was clear and non-technical. For a larger-scale project, it may be valuable to test the drafting of agreement forms with a group of parents to ensure that the language used is sufficiently clear and accessible, and so ensures that parents are fully informed when they give agreement. Similarly, for collection of data from students over 18, testing the agreement form with a group of students would help to provide assurance in the same way.

Process for sharing data

For a small-scale project, the process of sharing data between schools/MATs and Faculty was relatively straightforward. Once obtaining agreement from parents/carers, schools could share student work with Faculty via an SFTP server. The same method could be used when transferring student data and agreement wording documents from Faculty to DfE. This method can be replicated for large-scale data collection. However, it is important to note that the data sharing process for a larger-scale project (e.g. involving a higher number of student work) may require different methods of data collection depending on the volume of student work.

Removal of Personal information

Given the scale of this project, it would be sufficient to remove personal data by hand and to check manually that there was no student or other personal data included in the processed data. For a larger project, this would not be feasible, and an open-source or commercial off the shelf (COTS) model would need to be used to remove the data and remove personal data, hosted securely so that none of the data is shared either before or after removal of personal data. This is a standard approach to removal of personal data but it is important to note that this does not guarantee removal of all personal data, and a small amount of personal data may remain in the data once processed. In drafting the agreement forms for parents and the information provided to schools, it is important to be clear that the removal of all personal data is not guaranteed. Wording to this effect was included in the content drafted for this project, and could be used or adapted for future projects.

Implications for future work

Implications for developing similar tools

Approaches to extending into other subjects and year groups

The design of this PoC could be used as the basis for similar products to be developed for the education sector. If so, it would be valuable for it to also cover other subjects and year groups. Covering other year groups would require a wider codification of the National Curriculum and could mostly use the architecture as it is described here. The PoC uses requirements from Years 1-4 to highlight the severity of errors, demonstrating how this would work. Additional prompts would have to be developed to incorporate the assessment of additional aspects of language, such as tone and style, however, it is believed these could follow much of the same structure as those for spelling, punctuation and grammar.

Extending to other subjects could be more complex, depending on the extent of the extension. If the goal is still to assess the use of language against a set of requirements (such as the National Curriculum), then the extension is simpler. For English language subjects, it will follow the above, while for foreign languages the LanguageTool package would need to be swapped for an equivalent in that language, and the equivalent of the National Curriculum would need to be codified. If the goal is to also check if the student has correctly completed the task or answered the question, then this approach is less well suited. In that situation, it is recommended to use Retrieval Augmented Generation, to search the curriculum for the correct answer and then have an LLM assess if the pupil's response matches this.

Considerations when integrating tooling into a teacher's workflow

If a tool like this were to be developed, to ensure usability for teachers it would likely need to process images of student work, rather than just plain text. As such, the capability to upload and process images of student work to the tool, either via a scan or photo, would need to be incorporated into the pipeline proposed here. Considerations will therefore have to be made for how to practically manage:

- the process of parents giving their agreement, and being able to withdraw this if necessary;
- the storage, retention and deletion of pupil data; and
- the control of pupil privacy when processing pupil work via the OpenAI API.

Conducting real-world trials with educators to test a tool's impact

In order to truly understand and measure the potential benefit(s) of any tool similar to this, it is not enough to only test it with the limited number of educators who have seen and used this PoC. To specifically test if a tool supports the goal of reducing workload and improving educational outcomes by automating routine tasks for educators, it will be necessary for it to be used by a much larger group of teachers from a variety of schools. It would also be necessary for such a tool to be used and tested within teachers' everyday workflows, rather than in isolation as done here.

Implications for Generative AI in education

Guidance documents should incorporate AI-friendly structures

When designing educational content and guidance documents, including explicit structures and markers that AI tools can recognize and interpret is critical to enabling them to be more effectively used by AI-powered tools. This could mean tagging learning objectives, error types, and correction suggestions in a standardised format that AI applications can process. The codification approach used in this work is one example of how this could be done, but it will be important for the sector to align on a standard that best enables the ed-tech sector.

Fostering customisation and innovation through crowdsourced prompts

By enabling educators to modify and share their customised prompts, an ecosystem of collaborative innovation could be fostered. This crowdsourcing approach could not only enhance the repository of effective AI prompts but also leverage the collective expertise of educators to identify and disseminate the most impactful practices. Such an ecosystem would encourage continuous improvement in AI-generated educational content by the users, making it increasingly responsive to the evolving landscape of teaching and learning. There are already positive examples of this, e.g. Oak National Academy has made example prompts freely available and they have been helpful in the development of this current tool.

Appendixfe

A.1 Error detection: National curriculum error tables

Below we provide tables that show our approach of converting the National Curriculum into Python Code. For each of the 3 types of English language elements (Spelling, Punctuation and Grammar) we define the following columns:

- Year group: The year in the NC where this knowledge is expected to be acquired.
- Error subtype: The specific error subtype as defined by the NC.
- Description: Brief description of the correct use of this language element according to the NC.
- Error examples: Examples of incorrect use and corresponding correction.
- Detection method: Whether we used Language Tool (LT) or GPT to perform the detection of this particular error type.

Table 20: Grammar error table

Year Group	Error Subtype	Description	Error Examples	Detection Method
Year 3	A vs. An	Use of 'a' or 'an' based on the next word beginning with a consonant or vowel sound.	"I found an rock." -> "I found a rock.", "A acorn." -> "An acorn."	LT only
Year 2	Tense	The forms of words can change when they are used in the past, present, or future tense.	"Yesterday, I find a fossil on the beach." -> "Yesterday, I found a fossil on the beach."	LT and GPT
Year 1	Plural	Correct use of plural versions of words.	"I have been to a lot of churchs" -> "I have been to a lot of churches"	LT and GPT

Table 21: Punctuation error table

Year Group	Error Subtype	Description	Error Examples	Detection Method
Year 4	Fronted Adverbial Comma	Use of a comma after a fronted adverbial.	"In the morning we had breakfast." -> "In the morning, we had breakfast."	LT and GPT
Year 3	Speech Inverted Comma	Inverted commas are used to indicate direct speech.	"Watch out for that dog!" -> "'Watch out for that dog!'"	LT and GPT
Year 2	Full Stop	Full stops being used to end sentences.	"Today I went to the park Tomorrow I will go to the zoo" -> "Today I went to the park. Tomorrow I will go to the zoo."	LT and GPT
Year 1	Capital Letter	Correctly using capital letters at the start of sentences and for proper nouns.	"i live in England" -> "I live in England"	LT and GPT

Table 22: Spelling error table

Year Group	Error Subtype	Description	Error Examples	Detection Method
Year 4	Prefix Use	Correct use of prefixes to change the meaning of a word.	"His explanation was unplausible." -> "His explanation was implausible."	LT only

Year Group	Error Subtype	Description	Error Examples	Detection Method
Year 4	Suffix Use	Correct use of suffixes to transform words.	"I had an examinasion" -> "I had an examination"	LT only
Year 4	Homophones	Words that sound similar but are spelt differently.	"I didn't here the teacher talk." -> "I didn't hear the teacher talk."	LT and GPT
Year 4	Common Misspelling	Common words that are often misspelled by Year 4 students.	"I don't like my neybor." -> "I don't like my neighbour."	LT only
Year 2	Common Exception Words	Words that are common but break simple spelling rules.	"What are you going to do for Crismas?" -> "What are you going to do for Christmas?"	LT only
Year 2	Contractions	Multiple words combined into one with an apostrophe showing where letters were removed.	"I willn't skip my homework." -> "I won't skip my homework."	LT only
Year 1	Common Exception Words	Words that are common but break simple spelling rules for Year 1.	"'I want to go and play outside.' I sed." -> "'I want to go and play outside.' I said."	LT only

A.2 Performance of error detection pipeline on additional datasets

A.2.1 Performance on 20-sentence essay dataset

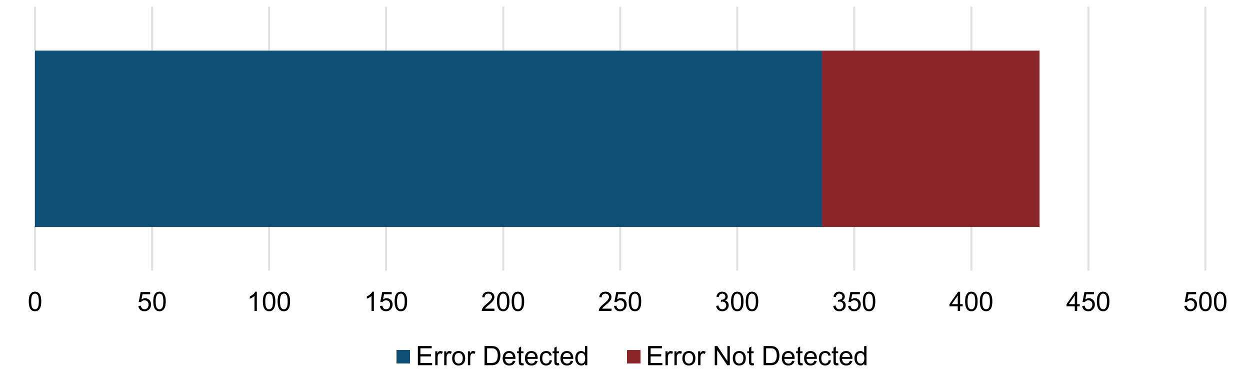


Figure 18: Error detection performance on the 20-sentence essay dataset

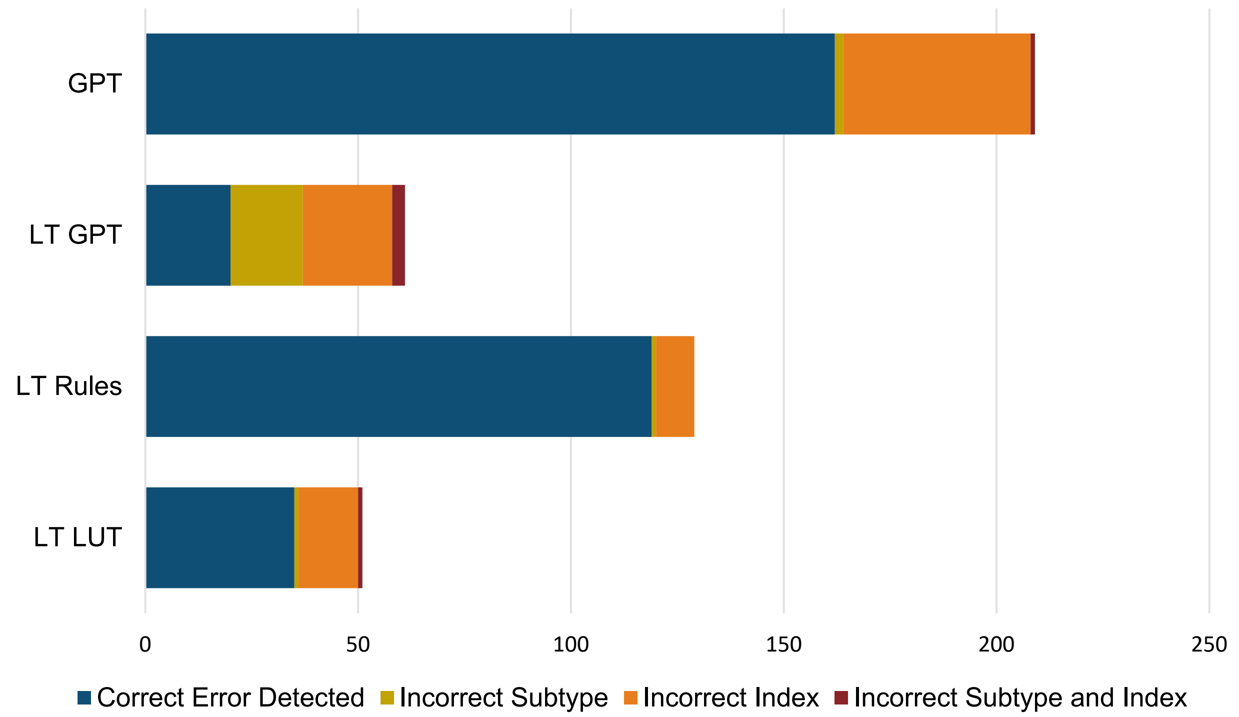


Figure 19: Count of errors correctly or incorrectly detected by each component of the detection pipeline, assessed against the 20-sentence essay dataset

There is a slight performance degradation when detecting errors in longer pieces of texts when compared to the single error sentence dataset, as described in Section 5.4.4. 25% of errors were incorrectly detected, which is a similar performance compared to misdetection in single error sentences. However, there is a higher rate of missing errors, at 22% compared to 12%. This is due to the longer text being passed to the error detection pipeline, which results in less controlled input and output and an increase in likelihood of missing errors.

A.2.2 Performance on synthetic pupil essay dataset

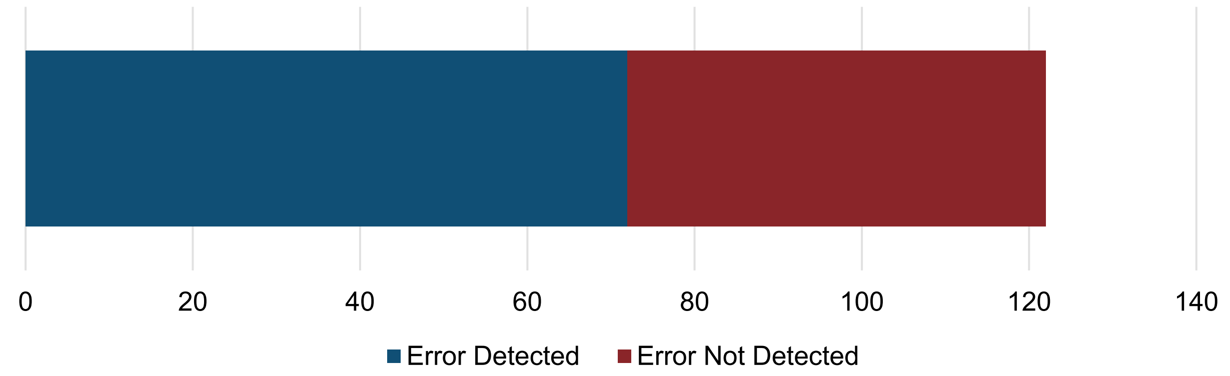


Figure 20: Error detection performance on the synthetic pupil essay dataset

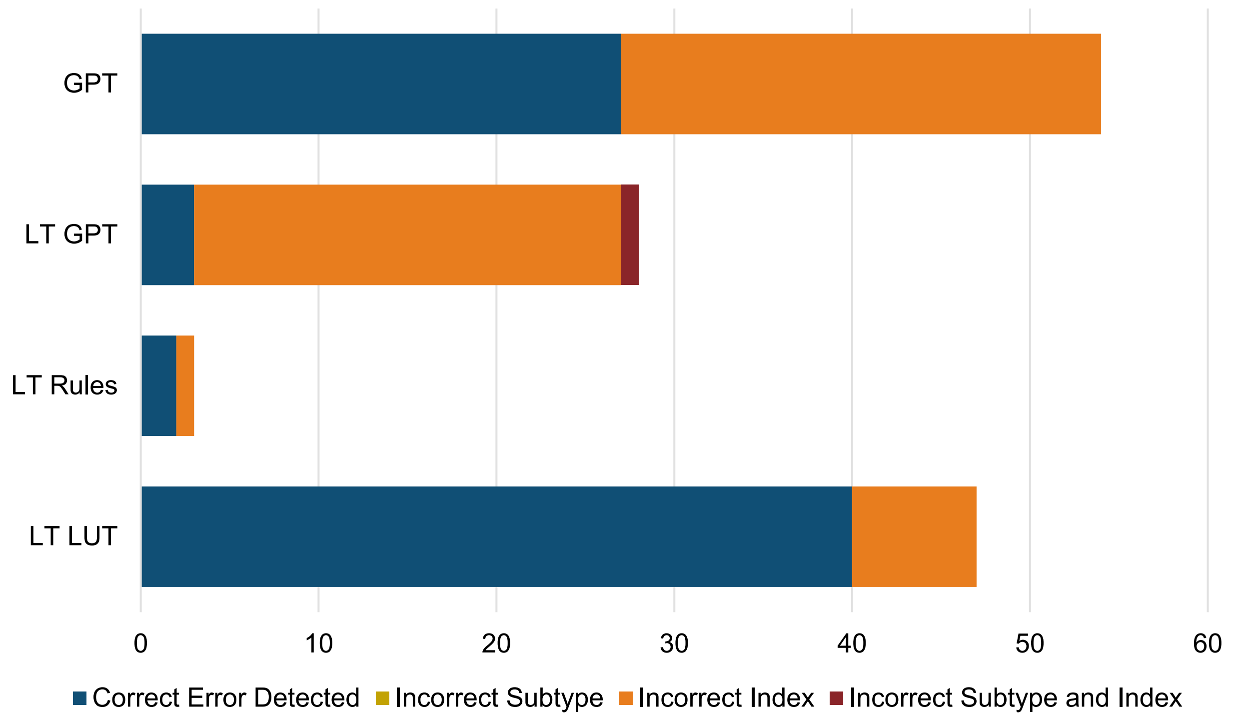


Figure 21: Count of errors correctly or incorrectly detected by each component of the detection pipeline, assessed against the synthetic pupil essay dataset

Further degradation of the error detection performance is observed when analysing the synthetic pupil essay dataset. The rate of undetected errors increases to 41%. There is also a higher misdetection or false discovery rate of 46% due to the classification of misdetections. However, the rate of misdetections being detected as the incorrect subtype decreased, with no error being classed as the right index but incorrect subtype.

A.3 Guidance document usage and derived LLM instructions

A.3.1 Feedback generation

Table 23: Details of existing guidance on feedback and how it was incorporated

Guidance Document	Quote	Derived Instruction
EEF Guidance Report: Teacher Feedback to Improve Pupil Learning	Feedback should focus on moving learning forward, targeting the specific learning gaps that pupils exhibit. Specifically, high quality feedback may focus on the task, subject, and self-regulation strategies.	Provide specific feedback on the types of errors the student has made, based on the list of errors provided.
EEF Guidance Report: Teacher Feedback to Improve Pupil Learning	Feedback that focuses on a pupil's personal characteristics is less likely to be effective. This may be because feedback about a person (rather than about the specifics of a task, their understanding of a subject, or their use of self-regulation) may not provide enough information to close a learning gap and move learning forward.	Provide general feedback on the essay, highlighting what the student has done well. Break this down into bullet points that are easy to read.
	Using feedback to inform future teaching and learning may not only be confined to a teacher's current class. Indeed, all teachers interviewed in the review of practice explained that they use the feedback they provide to inform how they teach the topic next year. By identifying the feedback that they regularly need to give, this informs them of the learning gaps and misconceptions that often arise. In turn, this can be used to adapt the initial instruction provided to pupils next time the teacher teaches this topic, improving the quality of initial teaching.	Carefully explain the most critical errors that the student has made and give examples of that mistake.

Guidance Document	Quote	Derived Instruction
	Providing clear, concise, and focused feedback. Sometimes less is more. Providing clear and concise feedback (which still features task, subject, and / or self-regulation advice) may support teachers in offering feedback that does not 'overload' pupils.	Make the feedback clear and concise, without repeating yourself.
	Breaking complex material into smaller steps (e.g. using partially completed examples to focus pupils on the specific steps).	Break down the errors the student has made into multiple steps. Show the student how to correct their mistakes with an explanation
	High-quality feedback can be written or verbal; it is likely to be accurate and clear, encourage further effort, and provide specific guidance on how to improve.	Provide specific guidance on how the student can improve

A.3.2 Task generation

Table 24: Existing guidance on task generation and how it was incorporated

Guidance Document	Quote	Derived Instruction
EEF Guidance Report: Teacher Feedback to Improve Pupil Learning	'What not to write': discuss with the class a list of 'what not to write'. This could follow a presentation to the class of an incorrect pupil response to a task; these are sometimes referred to as 'non-examples'.	Task Type: Non-Example Generate three examples of the error and ask the student to correct the sentences.

Guidance Document	Quote	Derived Instruction
EEF Guidance Report: Teacher Feedback to Improve Pupil Learning	Make feedback into detective work. For example, rather than saying to students, "If you swap these two paragraphs around the story would be better", you would say, "I think it would be better if two of these paragraphs were reversed. Find out which two you think I'm talking about".'	Task Type: Detective Work. Generate text with the error included and ask the student to identify which errors have been made.
EEF Guidance Report: Teacher Feedback to Improve Pupil Learning	In this strategy, a teacher poses three focused questions at the end of a written piece of work. The pupils then respond to these. Teachers should ensure that questions are meaningful and focused and they will be different for different students.	Task Type: Three Questions Generate three questions about the student's essay. Ask them (possible question types to ask).
EEF Guidance Report: Teacher Feedback to Improve Pupil Learning	Teachers may ask pupils to make specific corrections and edits to previous work. A checklist of common errors, with appropriate modelling of use by the teacher, may helpfully steer this approach.	Task Type: Correction Show the student 3 sections of their essay where they made the worst mistakes, ask them to correct these mistakes. Do not give them the answer.
ITT Core Content Framework	Breaking complex material into smaller steps (e.g. using partially completed examples to focus pupils on the specific steps).	Task Type: Partial Correction Generate an example of the error. Partially correct the error and ask the student to finish the correction.
ITT Core Content Framework	Receiving clear, consistent and effective mentoring in how to structure tasks and questions to enable the identification of knowledge gaps and misconceptions (e.g. by using common misconceptions within multiple-choice questions)	Task Type: Multiple Choice Generate a multiple-choice question related to the error, include plausible, but incorrect options as well as the correct choice.

Guidance Document	Quote	Derived Instruction
Oak Lesson Slide Decks	In general, content begins with a recap of the definition of the concept being taught (verb, subordinate clauses, apostrophes, etc.) followed by some examples, then some questions to test the students.	Task Meta Structure: You will follow the steps below to generate a task for the student: You will identify the error that the student is making. You will provide the student with the definition of the concept that are getting wrong. You will provide an example where the concept is being used. You will generate a [Task Type] task for the student to complete.
Key Stage 2 Tests	Specific examples of questions	Multiple choice: Which sentence has the correct punctuation Which sentence is a command Which pair of words are antonyms Which sentence is most formal In which sentence is X a verb/noun Which sentence has the correct use of tense Tick all sentences that have correct punctuation

Guidance Document	Quote	Derived Instruction
	Specific examples of questions	Classification: Given a list of sentences, tick if it is an exclamation or question Tick if apostrophe is for contraction or possession Tick if the underlined word is a verb/noun/adjective
	Specific examples of questions	Editing: Insert a comma/full stop/inverted commas/apostrophe in the correct place Rewrite the verbs to be in the past tense Chose the correct word out of the options in a sentence. E.g. We done / did a great job in picking the juiciest fruit for our pie.
	Specific examples of questions	Labelling: Circle the two words that are synonyms Circle all the adjectives/adverb in the sentence

A.3.3 Feedback evaluation

Table 25: Existing guidance on evaluating feedback and how it was incorporated

Guidance Document	Quote	Derived Instruction	Student or Teacher
EEF Guidance Report: Teacher Feedback to Improve Pupil Learning	Feedback should focus on moving learning forward, targeting the specific learning gaps that pupils exhibit. Specifically, high quality feedback may focus on the task, subject, and self-regulation strategies.	Does the feedback make references to specific errors made by the student?	Both
EEF Guidance Report: Teacher Feedback to Improve Pupil Learning	Providing clear, concise, and focused feedback. Sometimes less is more. Providing clear and concise feedback (which still features task, subject, and/ or self-regulation advice) may support teachers in offering feedback that does not 'overload' pupils.	Is the feedback clear and concise?	Both
ITT Core Content Framework	Breaking complex material into smaller steps (e.g. using partially completed examples to focus pupils on the specific steps).	Has complex material been broken down into smaller steps?	Student
EEF Guidance Report: Teacher Feedback to Improve Pupil Learning	Feedback that focuses on a pupil's personal characteristics is less likely to be effective. This may be because feedback about a person (rather than about the specifics of a task, their understanding of a subject, or their use of self-regulation) may not provide enough information to close a learning gap and move learning forward.	Does the feedback refer to the specifics of the task?	Both

Guidance Document	Quote	Derived Instruction	Student or Teacher
ITT Core Content Framework	High-quality feedback can be written or verbal; it is likely to be accurate and clear, encourage further effort, and provide specific guidance on how to improve.	Is the feedback understandable for a teacher?	Teacher
ITT Core Content Framework	High-quality feedback can be written or verbal; it is likely to be accurate and clear, encourage further effort, and provide specific guidance on how to improve.	Is the feedback understandable for a 9-year-old child?	Student
ITT Core Content Framework	High-quality feedback can be written or verbal; it is likely to be accurate and clear, encourage further effort, and provide specific guidance on how to improve.	Does the feedback provide specific guidance for how to improve?	Both
EEF Review of the Evidence	Identify patterns that suggest either a simple mistake or a deeper misunderstanding, and tailor feedback accordingly.	Are more severe mistakes highlighted compared to more minor ones?	Both
EEF Review of the Evidence	Feedback should go beyond correction, encouraging deeper understanding and dialogue. Feedback to focus more on formative comments and less on grades to promote learning.	Is the feedback not simply stating/pointing out the errors but also encouraging understanding and reflection?	Student

A.3.4 Task evaluation

Table 26: Existing guidance on task evaluation and how it was incorporated

Guidance Document	Quote	Derived Instruction
EEF Guidance Report: Teacher Feedback to Improve Pupil Learning	Feedback should focus on moving learning forward, targeting the specific learning gaps that pupils exhibit. Specifically, high quality feedback may focus on the task, subject, and self-regulation strategies.	Does the revision task make references to specific errors made by the student?
EEF Guidance Report: Teacher Feedback to Improve Pupil Learning	Providing clear, concise, and focused feedback. Sometimes less is more. Providing clear and concise feedback (which still features task, subject, and/ or self-regulation advice) may support teachers in offering feedback that does not ‘overload’ pupils.	Is the revision task clear and concise?
ITT Core Content Framework	Breaking complex material into smaller steps (e.g. using partially completed examples to focus pupils on the specific steps).	Has complex material been broken down into smaller steps?
EEF Guidance Report: Teacher Feedback to Improve Pupil Learning	Feedback that focuses on a pupil’s personal characteristics is less likely to be effective. This may be because feedback about a person (rather than about the specifics of a task, their understanding of a subject, or their use of self-regulation) may not provide enough information to close a learning gap and move learning forward.	Does the revision task refer to the specifics of the task?
ITT Core Content Framework	High-quality feedback can be written or verbal; it is likely to be accurate and clear, encourage further effort, and provide specific guidance on how to improve.	Is the revision task understandable for a 9-year-old child?

Guidance Document	Quote	Derived Instruction
ITT Core Content Framework	High-quality feedback can be written or verbal; it is likely to be accurate and clear, encourage further effort, and provide specific guidance on how to improve.	Does the revision task provide specific guidance for how to improve?
EEF Review of the Evidence	Identify patterns that suggest either a simple mistake or a deeper misunderstanding, and tailor feedback accordingly.	Are more severe mistakes highlighted compared to more minor ones?
EEF Review of the Evidence	Feedback should go beyond correction, encouraging deeper understanding and dialogue. Feedback to focus more on formative comments and less on grades to promote learning.	Is the revision task not simply stating/pointing out the errors but also encouraging understanding and reflection?

A.4 Feedback used to test generated content evaluation

A.4.1 Original feedback generated by the tool

General feedback

The student's essay is a vivid and imaginative depiction of the Titanic, demonstrating a strong ability to create a sense of place and atmosphere. The student's use of descriptive language is commendable, and they have a clear understanding of how to use adjectives to enhance their writing. However, there are several areas where the student could improve, particularly in spelling, punctuation, and grammar.

Specific feedback

Spelling:
The student has made several spelling errors, particularly with common words for their year group. For example, the word 'blue-sapphire' was misspelled as 'blue-saphire'. This is a Year 4 error, indicating that the student may need to revisit their spelling lists from this year. Other misspelled words include 'meals' and 'country'.

Punctuation:
The student has made a number of punctuation errors, particularly with capital letters and speech marks. For instance, the student failed to capitalise 'the Titanic', which is a Year 1 error. This suggests that the student needs to revisit the rules for capitalisation. The student also failed to use inverted commas correctly in direct speech, which is a Year 3 error.

Grammar:
The student has made several grammar errors, including incorrect use of 'a' and 'an', and incorrect tense. For example, the student used 'an' instead of 'a' before a word starting with a consonant, which is a Year 3 error. The student also used the present tense 'smiles' instead of the past tense 'smiled', which is a Year 2 error. In addition, the student has made errors with plurals, using 'passenger' instead of 'passengers', which is a Year 1 error. This suggests that the student needs to revisit the rules for forming plurals.

SUMMARY

The student has demonstrated a strong ability to create a vivid and engaging narrative. However, they need to work on their spelling, punctuation, and grammar. In particular, they need to revisit the rules for capitalisation, the use of 'a' and 'an', and the formation of plurals. They also need to work on their spelling, particularly of common words for their year group. With focused practice in these areas, the student has the potential to significantly improve their writing.

A.4.2 Degraded feedback #1

General feedback

The student's essay is a vivid and imaginative depiction of the Titanic, demonstrating a strong ability to create a sense of place and atmosphere. The student's use of descriptive language is commendable, and they have a clear understanding of how to use adjectives to enhance their writing. However, there are several areas where the student could improve, particularly in spelling, punctuation, and grammar.

Spelling:
The student has made several spelling errors, particularly with common words for their year group. For example, the word 'blue-sapphire' was misspelled as 'blue-saphire'.

Punctuation:
The student has made a number of punctuation errors, particularly with capital letters and speech marks. For instance, the student failed to capitalise 'the Titanic', which is a Year 1 error. In addition, the student has made errors with plurals, using 'passenger' instead of 'passengers', which is a Year 1 error. This suggests that the student needs to revisit the rules for forming plurals.

Summary

The student has demonstrated a strong ability to create a vivid and engaging narrative. However, they need to work on their spelling, punctuation, and grammar. In particular, they need to revisit the rules for capitalisation, the use of 'a' and 'an', and the formation of plurals. They also need to work on their spelling, particularly of common words for their year group. With focused practice in these areas, the student has the potential to significantly improve their writing.

A.4.3 Degraded feedback #2

General feedback

The student's essay is a vivid and imaginative depiction of the Titanic, demonstrating a strong ability to create a sense of place and atmosphere. The student's use of descriptive language is commendable, and they have a clear understanding of how to use adjectives to enhance their writing. However, there are several areas where the student could improve, particularly in punctuation.

Specific feedback

Punctuation:
The student has made a number of punctuation errors, particularly with capital letters and speech marks. For instance, the student failed to capitalise 'the Titanic', which is a Year 1 error. This suggests that the student needs to revisit the rules for capitalisation. The student also used the present tense 'smiles' instead of the past tense 'smiled', which is a

Year 2 error. In addition, the student has made errors with plurals, using 'passenger' instead of 'passengers', which is a Year 1 error. This suggests that the student needs to revisit the rules for forming plurals.

A.4.4 Degraded feedback #3

Punctuation:
The student has made a number of punctuation errors, particularly with capital letters and speech marks. For instance, the student failed to capitalise 'the Titanic', which is a Year 1 error. This suggests that the student needs to revisit the rules for capitalisation. The student also failed to use inverted commas correctly in direct speech, which is a Year 3 error.

Grammar:
The student has made several grammar errors, including incorrect use of 'a' and 'an', and incorrect tense. For example, the student used 'an' instead of 'a' before a word starting with a consonant, which is a Year 3 error. The student also used the present tense 'smiles' instead of the past tense 'smiled', which is a Year 2 error. In addition, the student has made errors with plurals, using 'passenger' instead of 'passengers', which is a Year 1 error. This suggests that the student needs to revisit the rules for forming plurals.

A.4.5 Degraded feedback #4

The student's use of descriptive language is commendable, and they have a clear understanding of how to use adjectives to enhance their writing. The student has made several spelling errors. The student has also made a number of punctuation errors, such as capital letters and speech marks. In addition, the student has made errors with plurals, this suggests that the student needs to revisit the rules for forming plurals. The student has demonstrated a strong ability to create a vivid and engaging narrative. However, they need to work on their spelling, punctuation, and grammar. With focused practice in these areas, the student has the potential to significantly improve their writing.

A.4.6 Degraded feedback #5

Spelling:
The student has made several spelling errors, particularly with common words for their year group. For example, the word 'blue-sapphire' was misspelled as 'blue-saphire'. This is a Year 4 error, indicating that the student may need to revisit their spelling lists from this year. Other misspelled words include 'meals' and 'country'.

A.5 Feedback system and user prompt parts

A.5.1 User prompt parts

Student essay and teacher/student feedback request

```
ESSAY
A student's essay would be inserted here.
TEACHER FEEDBACK = True
STUDENT FEEDBACK = False
```

Student essay detected and classified errors

```
ERRORS
The detected and classified errors for the student's essay would be
inserted here.
```

Feedback task instruction

```
TASK
One of:
Provide feedback, appropriate for a TEACHER, on the essay, given the
essay and the list of errors.
Provide feedback, appropriate for a STUDENT, on the essay, given the
essay and the list of errors.
```

A.5.2 System prompt parts

LLM output guidance and simple task context

```
OUTPUT FORMAT
Provide the output in **.json** format with a **single key** called either:
    - "teacher" if providing TEACHER feedback
    - "student" if providing STUDENT feedback.
Remember, JSON only requires you to escape double quotes, not single
quotes.
Do not escape single quotes (\') instead just use single quotes (') in
order to make a valid JSON.
Make sure to escape newline characters such as "\n": you should use "\\n".
Here is an example for the TEACHER feedback:
{"teacher": "" #Your generated teacher feedback}
Here is an example for the STUDENT feedback:
{"student": "" #Your generated student feedback}
CONTEXT
You are a tool used by UK Primary School teachers to generate feedback on
essays.
```

Additional task context

```
You are an expert in UK spelling, punctuation, and grammar.
You will be asked to provide one of two types of feedback on the essay:
1. TEACHER: You will provide feedback to UK Primary School teachers based
on student essays and the errors that have been detected.
      - The feedback will be used by the teacher to report on the
      performance of the student.
2. STUDENT: You will provide feedback to UK Primary School Students based
on their essay and the errors that have been detected.
      - The feedback for the student should be understandable for a 9
      year-old.
```

Provided information context

```
TASK
You will receive the following information:
  - "Essay": The essay the student has written.
  - "Errors": A list of the errors that the student has made in the essay.
You will receive the following information for each error:
  - "Type": The general category of the error which can be "spelling",
  "punctuation", or "grammar".
  - "Subtype": The name of the specific subtype of the error and a brief
  description of it.
  - "Error": The piece of text where the error is located.
  - "Year": The school year that the student is introduced to this kind of
  technique.
  - Who the feedback is intended for: TEACHER or STUDENT
You will generate feedback based on the essay and the list of errors.
```

Teacher feedback task context

```
TEACHER FEEDBACK
The feedback will be in the style of a report and should not be a letter
to the teacher.
Your writing should not be too formulaic, make the feedback engaging to
read and ready to be used in a report on the student.
```

Additional teacher feedback task context

```
You will provide the following information:
  1. "General Feedback": Provide general feedback on the essay.
    - You do not need to suggest "there are several areas that need
    improvement, particularly in spelling, punctuation, and grammar."
    - Be creative in your general feedback and avoid robotic language.
  2. "Specific Feedback": Provide specific feedback on the types of errors
  the student has made, based on the list of errors provided.
```

- You will only focus on error subtypes that have a lower "Year" since these are the foundation the student will build their skills off of.
- Group the errors of those subtypes and give one example of the error.
- Do NOT just list every error that the student has made but include examples of errors in your feedback.
- Mention the "Year" for the error subtype during your feedback for that subtype.
- Then give a brief overview of the other, less severe, subtype errors the student has made.
- If the student has made errors from Years 1 or 2, use more concerned language to indicate that these are errors the student is not expected to make.
- You will structure the feedback into Spelling, Punctuation, and Grammar sections.
- Do NOT start these sections with sentences like:
  "The student has made several grammar errors",
  "The student has made a number of punctuation errors"

3. "Summary": Summarise the specific feedback into a very brief paragraph.

Student feedback task context

STUDENT FEEDBACK
The feedback will be in the style of an encouraging, short piece of text.
Your writing for the student feedback should be understandable for a 9 year old.
Use simple language and non-technical terms.
Be encouraging to the student while giving feedback on what errors they made.

Additional student feedback task context

You will include the following:

1. Provide general feedback on the essay, highlighting what the student has done well.
2. Carefully explain the most critical errors that the student has made and give examples of that mistake.
   - Break this down into bullet points that are easy to read.

Additional guidance on creating feedback

GUIDANCE FOR CREATING FEEDBACK
This is guidance that you must follow when creating and structuring the feedback.
You will follow these points:
- Provide specific feedback on the types of errors the student has made, based on the list of errors provided

- Provide general feedback on the essay, highlighting what the student has done well.
- Carefully explain the most critical errors that the student has made and give examples of that mistake.
- Break this down into bullet points that are easy to read.
- Make the feedback clear and concise, without repeating yourself.
- Break down the errors the student has made into multiple steps. Show the student how to correct their mistakes with an explanation.
- Provide specific guidance on how the student can improve.

A.6 Detailed generation experiment evaluation scores

Table 27: Experiment 1 evaluation scores

Evaluation Metric	Baseline	Experiment	Change
Error Specific	81.8 ± 5.9	85.0 ± 2.4	+3.2 ± 6.4
Task Specific	58.0 ± 7.6	55.0 ± 6.3	-3.0 ± 9.9
Improvement Specific	79.0 ± 5.2	77.8 ± 2.4	-1.2 ± 5.8
Severe Errors	61.8 ± 4.9	63.2 ± 3.4	+1.4 ± 5.9
Complexity	61.3 ± 5.0	56.6 ± 5.1	-4.7 ± 7.1
Task Type Specific	97.7 ± 4.8	97.7 ± 4.8	+0.0 ± 6.8
Task Variety	54.2 ± 8.4	34.5 ± 7.5	-19.7 ± 11.2
Concise	97.4 ± 2.5	100.0 ± 0.0	+2.6 ± 2.5
Readable	91.7 ± 3.1	91.3 ± 2.6	-0.4 ± 4.1
Total Average Score	72.7 ± 3.3	69.3 ± 2.3	-3.4 ± 4.0

Table 28: Experiment 2 evaluation scores

Evaluation Metric	Baseline	Experiment	Change
Error Specific	81.8 ± 5.9	78.0 ± 8.3	-3.9 ± 10.2
Task Specific	58.0 ± 7.6	56.4 ± 7.8	-1.6 ± 10.9
Improvement Specific	79.1 ± 5.2	76.7 ± 6.1	-2.4 ± 8.1
Severe Errors	61.8 ± 4.9	60.0 ± 4.9	-1.8 ± 6.9
Complexity	61.3 ± 5.0	63.6 ± 5.1	2.4 ± 7.2
Task Type Specific	97.7 ± 4.8	95.9 ± 6.0	-1.8 ± 7.7
Task Variety	54.2 ± 8.4	54.3 ± 7.9	0.1 ± 11.5

Evaluation Metric	Baseline	Experiment	Change
Concise	97.4 ± 2.5	96.2 ± 2.6	-1.2 ± 3.7
Readable	91.7 ± 3.1	92.3 ± 3.6	0.6 ± 4.8
Total Average Score	72.7 ± 3.3	72.0 ± 3.4	-0.6 ± 4.8

Table 29: Experiment 3 evaluation scores

Evaluation Metric	Baseline	Experiment	Change
Error Specific	81.8 ± 4.5	84.1 ± 4.1	2.3 ± 6.1
Task Specific	44.2 ± 9.4	46.8 ± 7.0	2.6 ± 11.7
Improvement Specific	72.9 ± 4.5	75.9 ± 4.5	3.0 ± 6.4
Severe Errors	58.3 ± 5.7	61.8 ± 6.0	3.5 ± 8.3
Complexity	40.0 ± 11.5	45.5 ± 11.6	5.5 ± 16.3
Reflection	52.7 ± 12.3	53.6 ± 13.7	0.9 ± 18.4
Concise	67.7 ± 9.4	64.8 ± 7.9	-2.9 ± 12.3
Readable	75.7 ± 4.1	75.6 ± 5.0	-0.1 ± 6.5
Total Average Score	67.0 ± 5.7	66.8 ± 3.8	-0.2 ± 6.9

Table 30: Experiment 4 evaluation scores

Evaluation Metric	Baseline	Experiment	Change
Error Specific	82.3 ± 4.1	82.7 ± 4.7	0.5 ± 6.2
Task Specific	46.4 ± 9.2	50.5 ± 8.4	4.1 ± 12.4
Improvement Specific	71.8 ± 5.5	74.0 ± 4.6	2.2 ± 7.0
Severe Errors	58.6 ± 5.7	57.5 ± 6.4	-1.1 ± 8.5
Complexity	45.5 ± 13.7	47.0 ± 12.7	1.5 ± 18.7
Reflection	50.0 ± 14.1	45.0 ± 12.0	-5.0 ± 18.6
Readability	75.0 ± 5.1	76.1 ± 4.8	1.1 ± 7.0
Concise	67.2 ± 9.7	66.5 ± 7.8	-0.7 ± 12.5
Total Average Score	65.2 ± 4.3	67.3 ± 5.5	2.2 ± 7.0

Table 31: Experiment 5 evaluation scores

Evaluation Metric	Baseline	Experiment	Change
Error Specific	83.6 ± 4.0	83.2 ± 4.2	-0.5 ± 5.7
Task Specific	51.0 ± 7.8	47.7 ± 7.4	-3.2 ± 10.8
Improvement Specific	74.0 ± 4.0	73.2 ± 4.2	-0.8 ± 5.8
Severe Errors	60.0 ± 6.4	60.5 ± 6.1	0.5 ± 8.8
Complexity	49.0 ± 12.2	40.9 ± 12.4	-8.1 ± 17.4
Reflection	55.0 ± 6.7	50.0 ± 9.5	-1.6 ± 6.5
Readability	76.8 ± 4.5	75.2 ± 4.6	-1.6 ± 6.5
Concise	65.2 ± 8.7	68.3 ± 9.9	3.1 ± 13.2
Total Average Score	68.3 ± 3.2	65.9 ± 3.5	-2.4 ± 4.8

Table 32: Experiment 6 evaluation scores

Evaluation Metric	Baseline	Experiment	Change
Error Specific	81.8 ± 4.5	83.6 ± 4.1	1.8 ± 6.1
Task Specific	44.2 ± 9.4	48.6 ± 6.8	4.4 ± 11.6
Improvement Specific	72.7 ± 4.5	75.0 ± 4.5	2.1 ± 6.4
Severe Errors	58.3 ± 5.7	59.3 ± 6.0	1.0 ± 8.3
Complexity	40.0 ± 11.5	45.5 ± 13.7	5.5 ± 17.9
Reflection	52.7 ± 12.3	53.2 ± 11.3	0.4 ± 16.7
Readability	75.7 ± 4.1	76.0 ± 4.4	0.3 ± 6.0
Concise	67.7 ± 9.4	67.8 ± 10.3	0.1 ± 14.0
Total Average Score	67.0 ± 5.7	66.9 ± 3.5	-0.1 ± 6.7

Table 33: Experiment 7 evaluation scores

Evaluation Metric	First Simple Experiment	Final Experiment	Change
Error Specific	70.9 ± 5.0	82.7 ± 4.1	11.8 ± 6.5
Task Specific	57.3 ± 7.1	48.2 ± 8.7	-9.1 ± 11.2
Improvement Specific	66.1 ± 5.4	73.9 ± 4.0	7.8 ± 6.7
Severe Errors	52.7 ± 7.7	60.7 ± 5.5	8.0 ± 9.5
Complexity	44.1 ± 8.7	46.4 ± 11.5	2.3 ± 14.4
Reflection	62.7 ± 8.9	50.5 ± 12.1	-12.2 ± 15.0
Readability	79.0 ± 6.0	75.8 ± 3.3	-3.2 ± 6.8
Concise	78.5 ± 8.5	67.5 ± 8.7	-11.0 ± 12.2
Total Average Score	65.6 ± 2.6	66.6 ± 3.4	1.0 ± 4.3

A.7 Repeated per-essay evaluation generation scores

Table 34: Essay 1 evaluation scores

Evaluation Metric	Average	Spread
Error Specific	78.8	3.3
Task Specific	25.0	8.7
Improvement Specific	68.8	3.3
Severe Errors	50.0	0.0
Complex	30.0	0.0
Reflection	42.5	6.6
Readability	93.4	0.0
Concise	99.2	0.0
Average Score	60.2	1.7

Table 35: Essay 2 evaluation scores

Evaluation Metric	Average	Spread
Error Specific	80.0	0.0
Task Specific	35.0	12.8
Improvement Specific	70.0	0.0
Severe Errors	53.0	4.6
Complex	33.0	4.6
Reflection	50.0	8.9
Readability	98.1	0.0
Concise	85.8	0.0

Evaluation Metric	Average	Spread
Average Score	64.6	3.4

Table 36: Essay 3 evaluation scores

Evaluation Metric	Average	Spread
Error Specific	75.0	5.0
Task Specific	22.0	6.0
Improvement Specific	65.0	5.0
Severe Errors	48.0	4.0
Complex	30.0	0.0
Reflection	43.0	4.6
Readability	81.4	0.0
Concise	90.3	0.0
Total Average Score	55.7	1.4

Table 37: Essay 4 evaluation scores

Evaluation Metric	Average	Spread
Error Specific	86.0	4.9
Task Specific	32.0	9.8
Improvement Specific	76.0	4.9
Severe Errors	63.0	9.0
Complex	52.0	6.0
Reflection	49.0	10.4
Readability	96.1	0.0
Concise	49.2	0.0
Total Average Score	68.8	4.5

Table 38: Essay 5 evaluation scores

Evaluation Metric	Average	Spread
Error Specific	90.0	0.0
Task Specific	41.0	8.3
Improvement Specific	80.0	0.0
Severe Errors	64.0	9.2
Complex	54.0	10.2
Reflection	56.0	6.6
Readability	91.6	0.0
Concise	58.0	0.0
Total Average Score	71.0	2.6

Table 39: Essay 6 evaluation scores

Evaluation Metric	Average	Spread
Error Specific	90.0	0.0
Task Specific	52.9	7.0
Improvement Specific	80.0	0.0
Severe Errors	70.0	0.0
Complex	54.3	4.9
Reflection	52.9	7.0
Readability	97.9	0.0
Concise	76.5	0.0
Total Average Score	74.0	1.0

Table 40: Essay 7 evaluation scores

Evaluation Metric	Average	Spread
Error Specific	71.0	3.0
Task Specific	26.0	9.2
Improvement Specific	61.0	3.0
Severe Errors	50.0	0.0
Complex	30.0	0.0
Reflection	43.0	4.6
Readability	97.1	0.0
Concise	92.5	0.0
Total Average Score	59.4	1.7

Table 41: Essay 8 evaluation scores

Evaluation Metric	Average	Spread
Error Specific	70.0	0.0
Task Specific	47.5	4.3
Improvement Specific	60.0	0.0
Severe Errors	43.8	4.8
Complex	31.3	3.3
Reflection	48.8	3.3
Readability	95.2	0.0
Concise	91.3	0.0
Total Average Score	61.5	1.0

Table 42: Essay 9 evaluation scores

Evaluation Metric	Average	Spread
Error Specific	80.0	0.0
Task Specific	22.2	6.3
Improvement Specific	70.0	0.0
Severe Errors	50.0	0.0
Complex	32.2	4.2
Reflection	40.0	8.2
Readability	91.3	0.0
Concise	81.3	0.0
Total Average Score	59.6	1.6

Table 43: Essay 10 evaluation scores

Evaluation Metric	Average	Spread
Error Specific	90.0	0.0
Task Specific	67.8	6.3
Improvement Specific	85.6	5.0
Severe Errors	72.2	9.2
Complex	53.3	4.7
Reflection	69.4	6.8
Readability	99.2	0.0
Concise	56.9	0.0
Total Average Score	81.2	5.3

Table 44: Essay 11 evaluation scores

Evaluation Metric	Average	Spread
Error Specific	80.0	0.0
Task Specific	38.0	9.8
Improvement Specific	70.0	0.0
Severe Errors	55.0	5.0
Complex	40.0	7.7
Reflection	49.0	10.4
Readability	98.1	0.0
Concise	84.8	0.0
Total Average Score	66.0	2.8

A.8 Assessed guidance document

Table 45: Details of the guidance documents assessed and how they were used

Document	Description	Used for the PoC
ITT Core Content Framework	The initial teacher training (ITT) core content framework defines in detail the minimum entitlement of all trainee teachers.	Yes, in crafting prompts to evaluate generated feedback and tasks. Also used in guidance prompts experiments.
Early Career Framework	The early career framework (ECF) sets out what career teachers are entitled to learn about and learn how to do when they start their careers.	No, content is already covered by the ITT Core Content Framework
Guidance Report: Teacher Feedback to Improve Pupil Learning	A guidance report by the Education Endowment Foundation focusing the principles of good feedback rather than the written or verbal methods of feedback delivery.	Yes, in crafting prompts to evaluate generated feedback and tasks. Also used in guidance prompts experiments.
Improving Literacy in Key Stage 2	The report offers seven evidence-based recommendations for improving literacy, particularly in struggling pupils aged 7-11. Includes examples and resources to aid implementation, and is useful for both advanced younger pupils and older pupils who are behind.	No, focuses more on long-term support than on individual feedback and task generation.
English Grammar lesson units for Year 4 students - Oak National Academy	Series of lessons on Grammar, divided into sentence, word and punctuation level objectives. Includes video lessons, with each introducing a topic, giving examples, providing exercises and recapping the lesson.	Yes, used to get ideas for structuring output content and defining error subclasses.
KS2 English Grammar Teaching Resources for Lesson Planning - Oak National Academy	A collection of slides for literacy lessons ranging from year 3 to 6.	No, content is more focused on lessons rather than feedback and tasks.

Document	Description	Used for the PoC
Key stage 2 tests: 2023 English grammar, punctuation and spelling test materials	Example of a punctuation and spelling test.	Yes, used to create guidance-derived task generation prompts
National curriculum assessments: practice materials	Collection of links to sets of tests for English and other subjects	Yes, used to discover Key Stage 2 tests in the above row
Phonics: Teaching and assessing phonics, phonics performance data	Resources and guidelines on phonics education, strategies for teaching phonics, and possible assessment tools and examples	No, phonics was not incorporated into the tool
Embedding Formative Assessment - Evaluation report	Evaluates a program aimed at integrating formative assessment strategies in schools, analysing the program's impact on teaching practices and student learning outcomes.	Yes, used to refine tool outputs to give better feedback
EEF Review of the Evidence on Written Marking	Comprehensive analysis of marking practices in education. It discusses various aspects of written feedback, including effectiveness, impact on student learning, teacher workload, and best practices for delivering constructive feedback.	Yes, used to identify how the tool's feedback should be like.
EEF Evaluation Report: Improving Writing Quality	Paper on research into a technique for improving quality in primary and secondary school students	No, focuses more on long-term programs to improve writing rather than individual feedback and task exercises
English programmes of study: key stages 1 and 2 | National curriculum in England, September 2013	The national curriculum with appendices, defining how English should be taught for key stage 1 and 2	Yes, this was heavily used in order to create the National Curriculum codification, as described in Section 5.2

A.9 Agreement form

Below we provide the agreement form generated by Faculty and DfE for schools to disseminate to parents/carers.

The agreement wording form explains DfE's partnership with Faculty, outlines the project, provides parents/carers with a description of the tool, and explains that all student work would be anonymised through the removal of student names and other personal data.

Department for Education

Dear [Parent/Carer],

The Department for Education (DfE) is currently working with a specialist artificial intelligence (AI) company and schools across the country on an innovative and ground-breaking project to help deliver the education system of tomorrow.

We aim to develop Artificial Intelligence (AI) tools to:

1. assist teachers in reviewing and providing feedback on children's schoolwork.
2. establish an Education Content Store to drive quality and innovation in generative AI education tools

We would like to ask if you are willing to participate in this amazing project. To do this we would like to use your child's schoolwork to help develop the tool. The schoolwork is protected by copyright in the UK. Copyright protects original works, such as literary works, and stops others from using it without the owner's permission. The copyright in the schoolwork will most likely be owned by your child and you, as their parent, carer, or guardian, will therefore have the right to grant permission for DfE and its suppliers to use the schoolwork.

We will need you to agree that up to 5 pieces of your child's schoolwork can be shared with our supplier, Faculty, who are working with DfE under contract, and who will use the schoolwork to help develop the tool and also then share the schoolwork with DfE.

We understand your child's privacy is important and we will protect this by removing their name from the schoolwork before it is used to develop the AI tools.

This is an exciting initiative by the DfE, and we hope that you will be willing to take part. If you are happy to take part, please complete the required information in the form below and return it via email to [school email address to be included].

Kind regards,

[...]

I, ____[NAME OF PARENT/CARER]

at ____[ADDRESS]

I agree to

- up to 5 pieces of my child's schoolwork to be shared with the DfE for the following purposes
    - to help develop an AI tool to assist teachers in reviewing and providing feedback on children's schoolwork: Yes/No [please indicate by selecting Yes/No]
    - to establish an Education Content Store to drive quality and innovation in generative AI education tools: Yes/No [please indicate by selecting Yes/No]

I understand that this means DfE, its suppliers, representatives and agents can use schoolwork created by
____ [NAME OF CHILD]

for the development of artificial intelligence tools by the Department for Education.

This signed agreement form when completed needs to be sent via email to [school email address to be included].

More information about how the DfE handles personal information is published here:

[Personal Information Charter]

https://www.gov.uk/government/publications/privacy-information-artificial-intelligence-ai-tools

Department for Education

Reference: RR1442

ISBN: 978-1-83870-565-7

For any enquiries regarding this publication, contact www.education.gov.uk/contactus.

This document is available for download at www.gov.uk/government/publications.