

# **ChatGPT-Based Learning And Reading Assistant (C-LARA): Third Report\***

Branislav Bédi<sup>1</sup>, Francis Bond<sup>2</sup>, ChatGPT C-LARA-Instance<sup>3</sup>,  
Belinda Chiera<sup>4</sup>, Cathy Chua<sup>5</sup>, Anne-Laure Dotte<sup>6</sup>,  
Stéphanie Geneix-Rabault<sup>6</sup>, Sólvi Halldórsson<sup>7</sup>,  
Yixuan Lu<sup>7</sup>, Christèle Maizonniaux<sup>8</sup>, Chadi Raheb<sup>9</sup>,  
Manny Rayner<sup>4</sup>, Sophie Rendina<sup>10</sup>,  
Fabrice Wacalie<sup>6</sup>, Pauline Welby<sup>11</sup>,  
Rina Zviel-Girshin<sup>12</sup>, Hrafnhildur Ming Pórunnardóttir<sup>7</sup>

<sup>1</sup>The Árni Magnússon Institute for Icelandic Studies, Iceland;

<sup>2</sup>Palacký University Olomouc;

<sup>3</sup>OpenAI, US; <sup>4</sup>University of South Australia, Australia;

<sup>5</sup>Independent scholar, Australia;

<sup>6</sup>Université de la Nouvelle Calédonie, New Caledonia;

<sup>7</sup>University of Iceland, Iceland;

<sup>8</sup>Flinders University, Australia; <sup>9</sup>Independent scholar, Iran;

<sup>10</sup>Kowanyama Community, Australia;

<sup>11</sup>Aix Marseille Université, CNRS, Laboratoire Parole et Langage,

France; <sup>12</sup>Ruppin Academic Center, Israel



---

\*Authors in alphabetical order.

## Abstract

This report presents an overview of progress during the period March 2024–April 2025 on *ChatGPT-Based Learning And Reading Assistant* (C-LARA), an open source online platform which supports creation of multimodal texts for language learners that integrate audio, images, glosses and other annotations. Building on earlier work, we use GPT-4o and other Large Language Models to automate most or all of the annotation, guided by pedagogical needs and exploiting new AI capabilities. A central goal of the project is to explore how modern AI can act as a collaborative partner in research projects of this kind.

Over the past year, our principle achievements are the following:

- **More accurate annotation:** A principled treatment of multi-word expressions (MWEs) integrating segment-level translation now halves error rates in English glossing.
- **Flexible image generation:** We introduced a pipeline for generating coherent sets of images, addressing style consistency and repeated visual elements, and anticipate full-content coherence as current models mature.
- **Faster processing:** Parallelization of resource-intensive tasks typically yields an order-of-magnitude speedup.
- **Better support for Indigenous languages:** A new editing mode and validation checks facilitate manual annotation for languages where AI coverage is unavailable.
- **AI software engineering:** For standard Django functionality, the OpenAI “o1” model produces large, well-documented code blocks that often work on the first try.
- **AI academic writing:** The same AI can now compose full-length research articles based only on light human supervision.

In conclusion, we project that C-LARA is now perhaps a few months away from being able to consistently create high-quality annotated multimedia content for a wide variety of texts, and perhaps a year away from managing the bulk of its own software engineering.

# Contents

<b>Table of contents</b>	<b>3</b>
<b>1. How to access C-LARA</b>	<b>5</b>
<b>2. Introduction and overview</b>	<b>6</b>
2.1. Background and overall goals . . . . .	6
2.2. Main results . . . . .	7
<b>3. Related work</b>	<b>8</b>
3.1. Annotated Reading and Glossing Tools . . . . .	8
3.2. Multimodal Approaches . . . . .	9
3.3. AI-Enhanced Tools and Content Creation . . . . .	9
3.4. Summary . . . . .	10
<b>4. New functionality</b>	<b>10</b>
4.1. Improvements to linguistic annotation . . . . .	10
4.1.1. Annotating Multi-Word Expressions . . . . .	11
4.1.2. Segment translations . . . . .	11
4.1.3. Glossing and lemma tagging . . . . .	11
4.1.4. Displaying the final multimedia text . . . . .	11
4.1.5. Better segmentation . . . . .	12
4.1.6. Page audio . . . . .	12
4.2. Creating coherent image sets . . . . .	13
4.2.1. Overview . . . . .	13
4.2.2. Style . . . . .	13
4.2.3. Elements . . . . .	13
4.2.4. Pages . . . . .	15
4.2.5. Uploading Images . . . . .	18
4.2.6. Reviewing Images . . . . .	18
4.2.7. Summary and further directions . . . . .	20
4.3. Simple C-LARA . . . . .	22
4.4. Parallelism . . . . .	22
4.5. Better support for Indigenous languages . . . . .	22
<b>5. Evaluating linguistic annotation and image generation</b>	<b>24</b>
5.1. Third round of linguistic annotation evaluation . . . . .	24
5.2. Evaluating coherent image generation . . . . .	25
5.2.1. Looking ahead: new image generation capabilities . . . . .	27
<b>6. Using C-LARA with specific languages</b>	<b>32</b>
6.1. Icelandic and Mandarin (University of Iceland) . . . . .	32
6.1.1. Key findings on language learning with C-LARA . . . . .	32

6.1.2. Project outcomes and challenges . . . . .	33
6.1.3. AI in text and image generation . . . . .	33
6.1.4. Implications for C-LARA's use with less commonly spoken languages . . . . .	33
6.2. Kok Kaper (Kowanyama Community, Queensland) . . . . .	34
6.3. Iaai and Drehu (University of New Caledonia) . . . . .	37
<b>7. ChatGPT as a software engineer</b>	<b>38</b>
7.1. Simple form-based views . . . . .	38
7.2. Examples of simple Django views . . . . .	38
7.3. Why these tasks are straightforward; what's next? . . . . .	41
<b>8. ChatGPT as an author</b>	<b>42</b>
8.1. Overview of the four papers . . . . .	42
8.2. Conclusion and future outlook . . . . .	43
<b>9. Ethical issues</b>	<b>44</b>
9.1. AI authorship . . . . .	44
9.2. Obligations of the platform providers . . . . .	45
<b>10. Conclusions and further directions</b>	<b>46</b>
<b>Acknowledgements</b>	<b>47</b>
<b>References</b>	<b>47</b>
<b>A. Glossary of abbreviations and acronyms</b>	<b>50</b>
<b>B. Creating a picture book with Simple C-LARA</b>	<b>52</b>
<b>C. Creating a picture book with Advanced C-LARA</b>	<b>66</b>
<b>D. Creating a text in a language not supported by the AI</b>	<b>88</b>
<b>E. Prompt templates</b>	<b>101</b>

## **1. How to access C-LARA**

If you want to experiment with C-LARA, we recommend doing the following:

### **Create an account**

Go to the C-LARA login page, <https://c-lara.unisa.edu.au/accounts/login/>. Use the Register tab to create an account. You need to supply an email address to be able to use password retrieval and email confirmation, and receive email updates. We will not share your address with third parties.

### **Register an API key or submit a funding request**

Creating an account is free and language learners can use it to read existing C-LARA content and do a lot of other things. But the AI-based functionality which permits content creation and is the heart of the platform uses GPT-4 and other OpenAI models, which costs money. In order to access these functions, in particular to use the AI to create multimodal content, you need to be able to pay for the OpenAI calls. Independent language learners may well wish to create their own tailored content. It is easy and fast as the AI does most of the work.

The preferred solution is to use an OpenAI API key valid for GPT-4. Open a ChatGPT account at <https://chat.openai.com/auth/login> if you do not already have one and get an API key from <https://platform.openai.com/api-keys>. Then on C-LARA, go to User profile etc > Edit configuration information and enter your key. Calls you make to OpenAI through C-LARA will then be charged directly to your account.

If this solution does not work for you, perhaps because you are not able to open a ChatGPT account due to your geographical location, we have a small amount of funding available to help people experiment. Go to the tab Social network > Funding request and submit a request, making suitable choices from the dropdown menus and adding a few words explaining roughly what you are thinking of doing and why you can't use an API key. We are typically handing out five dollars per request, which will let you create ten to twenty pieces of content.

### **Use “Simple C-LARA” to create a piece of content**

Follow the steps in Appendix B to enter “Simple C-LARA” and create a piece of content. You just need to specify the languages and supply an initial prompt saying what you want C-LARA to write, and the AI does the rest for you.

When you have created some content, consider filling out the satisfaction questionnaire at the end. It should take a couple of minutes. Don't feel you have to be nice. If you tell us about things that did not work, it is more likely that we will fix them.

## 2. Introduction and overview

### 2.1. Background and overall goals

ChatGPT-Based Learning And Reading Assistant (C-LARA; <https://www.c-lara.org/>) is an international open source project inaugurated in March 2023. Building on the earlier LARA project (Akhlaghi et al., 2019), one of the top-level goals was the same: we aimed to create a online platform that would support construction of multimodal texts useful for language learners who wished to improve their reading and listening skills. A text of this kind would contain annotations typically including audio files, images, word glosses, a concordance, and maybe other things.

The recent release of GPT-4, however, opened up new possibilities. In LARA, a large part of the work involved in building a high-quality multimodal text had to be performed manually. It seemed to us that GPT-4's intelligence was at a completely different level to previous AI software, and there was a realistic prospect of its being able to do all, or nearly all, of the annotation work. In addition, it was already apparent that it had considerable strengths as a software engineer. It did not seem unreasonable either to hope that it could handle much of the implementation work, and even a substantial part of the process of writing academic papers; at the very least we thought that we would learn a great deal by involving the AI integrally in the project from the start, treating it as though it were an intelligent neurodivergent colleague and finding out what it could do. Work in during the first year of the project, up to March 2024, is summarised in the first and second C-LARA progress reports (Bédi et al., 2023, 2024).

By March 2024, we had a generally accessible platform, deployed on a server at the University of South Australia, and a fair amount of experience in using it. In order to make further progress, it seemed to us that the following were the most important issues to focus on during the next phase.

**More accurate annotation:** Unsurprisingly, we found both that GPT-4's performance varied a great deal between languages, and that it improved quickly over time as new models were released. It was, however, apparent that Multi-Word Expressions (MWEs) were a major obstacle to obtaining good annotation results in most of the languages we examined. A large proportion of the mistakes the AI made when glossing and lemma-tagging text involved MWEs: worse, even when it annotated them correctly, teachers stressed that the way MWEs are displayed was pedagogically important. In an English sentence like “They **threw** the man **out**”, the words in bold form a single lexical unit; the presentation needs to make this clear. We decided that a principled approach to handling MWEs was required.

**Flexible creation of images:** Teachers also stressed the importance of including images, especially for beginner and low intermediate learners. It was easy to create short C-LARA documents which contained a single AI-generated image; just passing the full text to DALL-E-3 usually gave an acceptable result. This simple method, however, did not work if we were trying to produce a text with many images, since the images were not coherent either in terms of style or of content. We evidently required a better way to manage the image generation process.

**Faster processing:** Processing was too slow. We needed to speed it up.

**Support for Indigenous languages:** As with the previous LARA project, there was considerable interest from the Indigenous language community. We found that the initial architecture of C-LARA was too heavily based on the assumption that the AI would do all the work, evidently impossible in most Indigenous languages. We needed to rearrange things so that manual annotation was also well supported.

**Tracking the AI's ability to do software engineering:** It was evident from the start that OpenAI and other AI platform providers were targeting software engineering as a key use case. Most of the publicly available data, however, was based on small challenge problems. We wanted to gain a better understanding of how the AI managed when faced with a codebase measured in the tens of thousands of lines.

**Tracking the AI's ability to do academic writing:** During the first year of the project, we found that the AI did well when asked to write short academic texts, like extended abstracts; it struggled even to understand texts that were more than a couple of thousand words long. We were curious to know what prospects there were of scaling up to texts of conference or journal paper length, i.e. several thousand words.

## 2.2. Main results

During the last year, we made good progress on all of the above issues.

**More accurate annotation:** We have implemented a principled treatment of Multi-Word Expressions, annotated using a Chain of Thought method (§4.1). Combining the MWE information with segment translation annotations, also added during this phase, we reduce the average glossing annotation error rate in English by more than 50%. MWEs are now displayed as single units in the final multimodal form.

**Flexible creation of images:** We have implemented a method for creation of coherent sets of images (§4.2), which in turn defines the style, the recurrent visual elements, and finally the page images. Coherence in terms of style is generally good. Coherence in terms of content—the recurrent visual elements—is still unsatisfactory with the deployed version of the system. It seems likely, however, that improved models currently in the process of being released will soon resolve these problems and yield fully acceptable coherent image generation (§5.2.1).

**Faster processing:** We have introduced parallel processing for the costly operations, generally yielding a speedup of about an order of magnitude (§4.4).

**Support for Indigenous languages:** We have added a new editing mode specifically designed for Indigenous languages (§4.5). Annotation information is organised by pages, and there is automatic checking and correction to catch careless editing errors giving rise to inconsistencies between the different annotated versions,

**Tracking the AI's ability to do software engineering:** We observe a steady improvement in the AI's performance (§7). A year ago, GPT-4o would typically deliver a few dozen

lines of code in one response, and this code would usually require nontrivial human debugging. For straightforward Django functionality which only involves accessing, displaying and modifying database content, i.e. the majority of the supported platform functions, it is now common to see o1 producing two hundred or more well-documented lines which often work first time.

**Tracking the AI’s ability to do academic writing:** There has similarly been a dramatic improvement in academic writing performance (§8). The o1 model is now capable of writing a full-length paper more or less on its own, with human input comparable to that a PhD supervisor might offer a gifted student.

### 3. Related work

A growing body of research in computer-assisted language learning (CALL) highlights the importance of *annotated, multimodal* texts to increase reading comprehension and vocabulary acquisition (Plonsky and Ziegler, 2016). Traditional platforms have long offered dictionary-based or teacher-curated annotations, while more recent systems incorporate artificial intelligence (AI) to automate tasks such as glossing, text generation, and image creation. Below, we summarize key developments and representative tools in this space.

#### 3.1. Annotated Reading and Glossing Tools

Several open-source platforms focus on providing word- or phrase-level annotations to help learners read authentic texts with minimal interruption. **Learning With Texts (LWT)**<sup>1</sup> allows users to import any target-language text, tokenize it automatically, and click on unfamiliar words to look them up in an external dictionary. Translations and notes can then be saved to build a personal vocabulary database. A similar approach underpins **LingQ**<sup>2</sup> and **Readlang**<sup>3</sup>, both of which offer browser-based or mobile interfaces where users can translate words on the fly and create flashcards for spaced repetition review. By significantly reducing “lookup friction,” these systems encourage extensive reading in the target language.

Another example is **Clilstore**<sup>4</sup>, an authoring and sharing platform designed for Content and Language Integrated Learning (CLIL). Teachers can embed video or audio alongside a transcript, with each word linked to online dictionaries through an underlying service called Wordlink. This functionality, also seen in earlier projects like **Lingro**<sup>5</sup>, highlights the longstanding tradition of providing quick dictionary access to support comprehension. While these tools rarely introduce erroneous translations (dictionary-based glosses remain static), they offer limited flexibility in explaining context-sensitive or idiomatic expressions.

---

<sup>1</sup><https://learning-with-texts.sourceforge.io/>

<sup>2</sup><https://www.lingq.com/>

<sup>3</sup><https://readlang.com/>

<sup>4</sup><https://multidict.net/clilstore/>

<sup>5</sup>[urlhttps://adaptivelearninginelt.wordpress.com/2017/08/05/what-we-can-learn-from-glosses-and-research-into-glosses](https://adaptivelearninginelt.wordpress.com/2017/08/05/what-we-can-learn-from-glosses-and-research-into-glosses)

### 3.2. Multimodal Approaches

To provide richer context, platforms increasingly integrate *audio* and *visuals* alongside text. Many systems, including **Clilstore** and **LingQ**, support synchronized audio playback, allowing learners to listen to a native or text-to-speech (TTS) recording while following the transcript. Some tools highlight each sentence or phrase in “karaoke” style, improving comprehension and aiding pronunciation. Commercial apps such as **Beelinguapp**<sup>6</sup> focus on parallel text with audiobook narration, whereas projects like **Duolingo Stories**<sup>7</sup> embed illustrated narrative panels with native-speaker audio, eliciting users’ comprehension through mini-quizzes.

Image and video integration often relies on manual embedding or linking; however, research suggests that visual context can boost learner engagement and recall (Paivio, 1991). More recently, some open-source platforms have begun experimenting with *automatic* image generation, where AI systems create illustrations keyed to textual segments. This expansion of multimodality, though still in early adoption, indicates a trend toward more immersive reading environments.

### 3.3. AI-Enhanced Tools and Content Creation

While dictionary-based or rule-based annotations remain popular, *AI-driven* features are growing rapidly. Commercial services like **Quizlet Q-Chat**<sup>8</sup> and **Duolingo’s AI chatbots** demonstrate how large language models (LLMs) can handle learner queries, explain tricky phrases, or create practice dialogues on the fly. However, few platforms go as far as providing *entirely AI-generated reading content with automated annotations*, which is the territory of emerging systems such as **StoryWizard AI**<sup>9</sup> (illustrated children’s stories) and advanced open-source prototypes like Multimodal-GPT<sup>10</sup>. These approaches aim to reduce teacher workload by generating text, glosses, and even morphological analysis without extensive human intervention.

Early evaluations of AI-based glossing show promise but also underscore the need for human review, especially in languages with complex morphology or frequent multi-word expressions. In many cases, a hybrid strategy—where AI handles the bulk of routine tasks and a teacher or domain expert refines the output—yields the best results. Traditional dictionary-based glosses rarely produce incorrect translations, but they may fail to capture nuances or idiomatic usage. AI systems, by contrast, can produce context-sensitive explanations yet risk “hallucinations” or subtle inaccuracies. Balancing these strengths and weaknesses remains an active area of investigation in CALL research.

---

<sup>6</sup><https://beelinguapp.com/>

<sup>7</sup><https://blog.duolingo.com/introducing-duolingo-stories/>

<sup>8</sup><https://quizlet.com/blog/q-chat>

<sup>9</sup><https://www.storywizard.ai/>

<sup>10</sup><https://github.com/open-mmlab/Multimodal-GPT>

### 3.4. Summary

Across open-source and commercial solutions, the consensus is that annotated, multimodal texts can significantly enhance reading comprehension and vocabulary uptake. Platforms like LWT and LingQ have validated the utility of rapid dictionary lookups and integrated flashcards, while Clilstore and Duolingo demonstrate how audio or video can bolster textual understanding. With the emergence of AI-assisted generation and glossing, tools are beginning to reduce the time and expertise required to create high-quality learning materials. The next sections discuss how our project builds on these ideas to offer a more scalable, community-driven approach to AI-facilitated annotated reading.

## 4. New functionality

In this section, we describe new functionality added between March 2024 and March 2025. Specifically, we consider the following:

**Improvements to linguistic annotation:** Linguistic annotation has been much improved.

In particular, it now includes a systematic treatment of Multi-Word Expressions.

**Coherent image sets:** There is now good support for creation of texts with multiple images, where the images are aligned to be reasonably coherent with each other in terms of style and content.

**Parallelism:** Both linguistic annotation and creation of images have been now use parallel processing, making them much faster.

**Simple C-LARA:** Simple C-LARA has been upgraded to incorporate most of the above functionality. In particular, it can be used to create texts that include annotations of MWEs and coherent image sets.

**Better support for Indigenous languages:** The interface for editing of Indigenous language texts has been completely revised.

### 4.1. Improvements to linguistic annotation

In the second C-LARA report, we evaluated annotation performance and identified poor handling of Multi-Word Expressions (MWEs) as the dominant issue. Here, we describe the principled solution we have implemented, which involves addition of a new annotation phase specifically for identification of MWEs. We have also added an annotation phase for addition of segment translations. We explain how these two new phases feed into processing for lemma-tagging and glossing, and also summarise improvements to segmentation and audio.

#### **4.1.1. Annotating Multi-Word Expressions**

After some initial experimentation, we determined that a Chain of Thought approach seemed most suitable for annotating MWEs. We process each segment separately, using the prompt template shown in Figure 17. The prompt template contains slots for the text language, the text in JSON form, and a list of few-shot examples; a typical few-shot example for English is shown in Figure 18. The few shot-examples are created in a bootstrapped manner by running the prompt on gpt-4o with an initial set of two handcrafted examples and editing the resulting output.

Initial testing and prompt engineering was done using two annotated English Sherlock Holmes stories from the NTU-MC (Natural Text Understanding - Multilingual Corpus: Bond et al., 2021). These include MWEs from an extended version of wordnet (Fellbaum, 1998).

#### **4.1.2. Segment translations**

We have also added a new annotation phase which creates translations for segments. The segments are divided into groups, currently of about 250 words each, and submitted to gpt-4o using a prompt which provides minimal instructions about format. As well as being useful in itself, the segment translation are also used for glossing, as described in the immediately following section, and sometimes for creation of images.

#### **4.1.3. Glossing and lemma tagging**

The point of adding MWE information is that it gives us options for improving glossing and lemma tagging. If a word is part of an MWE, we want the associated gloss and lemma to refer to the MWE, not the word itself. For glossing, we find it is also useful to make available the segment translation information and the full text of the document, in order to provide context.

We perform the glossing and lemma tagging operations on each segment separately. We experimented with several different ways of passing in the MWE information; the one that worked best was to present the segment as a JSON list with one element per lexical item, and the MWE information paired with the words it refers to. The prompt template is shown in Figure 19.

The prompt template for lemma tagging is similar but a little simpler.

#### **4.1.4. Displaying the final multimedia text**

To present the MWEs in an intuitive way, the multimedia text contains JavaScript which modifies the highlighting behaviour to show MWEs as units; thus clicking on one word in an MWE highlights the other words as well. Similarly, when creating audio associated with words, a word that is a component of an MWE is linked to audio for the whole MWE, and when compiling the concordance the lemma that appears is one for the whole MWE.

Figure 1 illustrates.

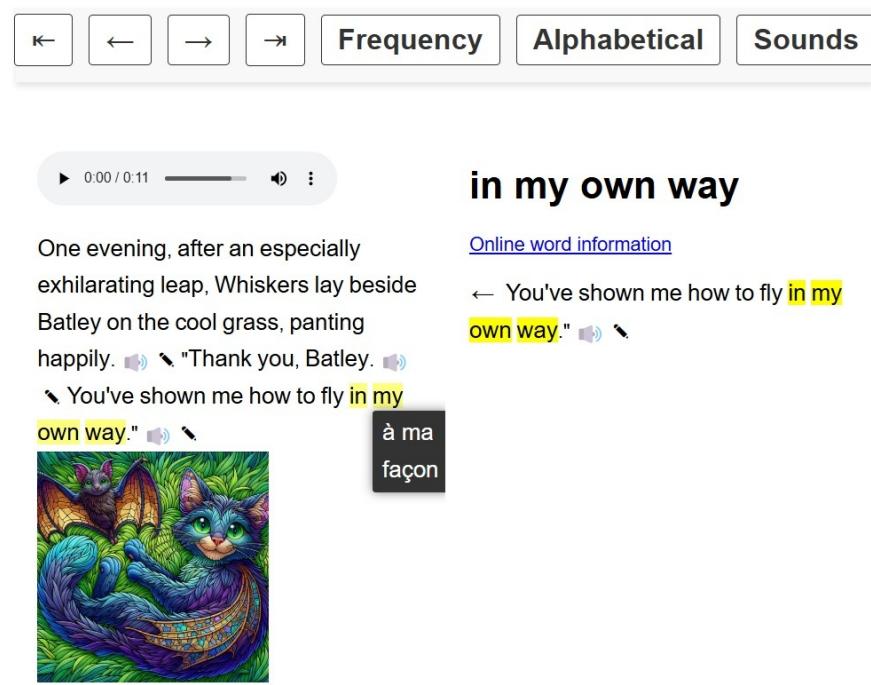


Figure 1: Example from “The Cat and the Bat” illustrating presentation of MWEs in the multimodal text

#### 4.1.5. Better segmentation

The segmentation phase is now divided into two subphases. In the first subphase, the AI is used to divide the text into pages, with each page divided into segments. The prompt recommends that the AI first determines the genre of the text (story, essay, poem, dictionary, etc), and then makes an appropriate subdivision. For example, in a story a typical page would be 1–3 paragraphs and a typical segment would be a sentence, while in a poem a typical page might be 1–3 verses and a typical segment would be a line. In the second phase, each segment is processed separately, using a language-dependent prompt which gives examples of how to subdivide words or mark words containing punctuation marks (hyphens, apostrophes) as single lexical units.

The new scheme gives much better results, particularly in the first subphase.

#### 4.1.6. Page audio

A small but useful addition is that the final multimodal rendering stage now adds an audio file for each whole page, creating it by concatenating the pages segment audio files.

## 4.2. Creating coherent image sets

### 4.2.1. Overview

A major new piece of functionality is support for creation of text containing multiple images. Here, the challenge is to produce images that are coherent with each other.

Coherence can involve both style and content. Coherence of style means choosing a single artistic theme, a defined colour palette, and so on. Coherence of content means that elements which occur in more than one image (people, animals, objects, locations...) will be realised in roughly the same way. Our current approach in both cases is the same. We instruct the AI to create descriptions for the aspects, style or repeated elements, that will be shared between images; we then include this information in the prompts that generate the final descriptions that are passed to DALL-E-3 to create the images themselves. We thus have a cascade of prompts, where each prompt creates material that is used in later prompts.

We produce multiple versions of both descriptions and images. The generated sets of alternatives can be reviewed either by the human user or by the AI, to try to find the best image or determine that none of the alternatives are satisfactory and new ones need to be generated.

We find that the approach works well for style. Results for content are less convincing with the models currently available through the API; it appears, though, that new models now released in the interactive ChatGPT interface (“Images in ChatGPT”) are powerful enough to do an adequate job. We discuss this below.

### 4.2.2. Style

The starting point in the process of creating a coherent image set is to generate a style description. The prompt template used to do this is shown in Figure 20. We pass in the user’s instructions, the text, and optionally some background information.

We find in practice that it is usually most effective to provide short, vague instructions for generating the style and let the AI make the substantive decisions. Figure 2 shows another example from “The Cat and the Bat”.

### 4.2.3. Elements

Once the style has been determined, the next step is to create descriptions for possible recurring elements in the text. These will typically be people, animals, objects or locations.

The “Elements” step consists of two substeps. In the first, the AI is instructed to create a list of elements. This is done using the prompt template shown in Figure 21, which takes as input the text, the already generated style description, and possible background information. So far, this operation seems rather unreliable, so we provide controls allowing the user to add or delete elements.

In the second substep, the AI is instructed to create a description for each element, and then

### **User-supplied style request**

Create images in the style of a stained glass window.

### **AI-generated style description**

Create images with a stained glass window style, emphasizing vibrant, translucent colors that capture light and create a luminous effect. Use a rich color palette, including deep blues, emerald greens, warm ambers, and soft purples, to convey the beauty of the night sky and garden. Ensure bold, black outlines define each segment, mimicking the lead lines in traditional stained glass. Incorporate intricate details like delicate textures in Whiskers' fur and Batley's wings. The mood should be whimsical and enchanting, reflecting a magical atmosphere of discovery and friendship. Maintain a harmonious balance between light and shadow to evoke a sense of wonder under the starlit sky. Incorporate elements like twinkling stars and moonlit scenes to enhance the story's dreamy setting. This cohesive style should unify the illustrations, creating a timeless, fairy-tale quality as Whiskers and Batley embark on their skyward adventure.

### **Image exemplifying style description**



Figure 2: Creating a style description for “The Cat and the Bat”

produce an illustrative image to give the user feedback. The prompt template used to create an element description is shown in Figure 22. Figure 3 continues the running example with “The Cat and the Bat”.

#### 4.2.4. Pages

With the style description and the element descriptions in place, the final step is to generate the actual page images. This again breaks down into two substeps. In the first, the AI is told to identify the relevant elements from the full list; in the second, the template is passed the text, background, style description and selected elements, and instructed to create the description for the page image.

The template for the first operation is shown in Figure 23, and is straightforward. The second substep is the more interesting one. We originally implemented it with the template from Figure 24, where we pass in the element descriptions, that is to say pieces of text, and instruct the AI to create page descriptions from them which in some way incorporate these pieces of text. Figure 4 continues the running example.

### Using a multimodal model: Gemini Imagen 3

Having created several dozen C-LARA texts using DALL-E-3, our overall conclusion is that it is in general not capable of maintaining adequately coherent images. The same element is often not presented similarly enough across multiple images, even if the element description generated by the AI is quite detailed and DALL-E-3 appears to have followed it. Words do not appear to provide a sufficiently fine-grained way to capture an image.

We also experimenting with using the Gemini Image 3 model as an alternative to DALL-E-3; C-LARA now supports both image generators. The important difference is that Imagen 3 is multimodal; instructions can include references to web-accessible images as an alternative to descriptions of images, with the images passed as URLs. These are interpreted directly.

A practical problem with the current version of Imagen 3 is that there are strong restrictions on creating and referencing images of people, particularly children. Our current workaround is to implement a hybrid strategy: when constructing the page image description, the AI is instructed to use text descriptions of elements when the element represents a person, and images otherwise.

State of the art in image generation is evolving rapidly. OpenAI’s “Images in ChatGPT”, released in late March 2025, represents a substantial advance on both DALL-E-3 and Imagen 3. API support is not yet available, but we assume that this will soon be added. Our impression is that this will essentially solve the problem of producing coherent sets of images for C-LARA texts. We discuss the issues further in §5.2.1.

**Element name**

Whiskers

**AI-generated element description**

\*\*Element Specification: "Whiskers" (the cat) in Stained Glass Style:\*\*

Whiskers is a small, agile cat with a sleek yet slightly rounded build, exuding a sense of curiosity and wonder. His fur is a mixture of deep blues and soft purples, creating a dreamlike quality that reflects the night sky. The fur is delicately textured, resembling individual strands captured in stained glass, with shimmering star-like specks throughout. His eyes are large and vibrant, with a piercing emerald green hue that glows luminously. Whiskers' tail is long and sinuous, often held high, and features subtle gradient patterns from dark blue to lighter shades. His expression is one of gentle curiosity and enthusiasm, with a slight, playful tilt to his head, embodying a sense of adventure. Bold black outlines define his form, mimicking traditional lead lines in stained glass, creating a whimsical and enchanting presence under the moonlit garden setting.

**Image exemplifying element description**

Figure 3: Creating an element description for “The Cat and the Bat”

## **Page text**

In the quiet town of Windy Heights, there lived a curious little cat named Whiskers. Whiskers had always been fascinated by the night sky, watching the stars twinkle and the moon shine brightly from his cozy spot on the windowsill.

## **AI-generated page image description**

**\*\*Essential aspects:\*\***

- Whiskers, the cat, is seated on a windowsill in a cozy setting, gazing at the night sky.
- The night sky is filled with twinkling stars and a bright moon, as Whiskers is fascinated by these celestial sights.
- Whiskers should exhibit a curious expression, with his emerald green eyes reflecting the beauty outside.
- The vibe of the scene is quiet and contemplative, capturing Whiskers' fascination with the night sky.

**\*\*Image Specification:\*\***

Create an illustration in a stained glass style featuring Whiskers, the small cat, sitting on a windowsill. Render Whiskers' fur in deep blues and soft purples, with delicate textures that resemble individual stained glass strands and star-like specks. His eyes should be luminescent emerald green, wide with curiosity. The setting is a calm, serene interior with the window overlooking a captivating night sky.

The night sky defies typical stained glass elements with a vivid blend of deep blues and purples, embodying a dreamlike quality. Stars are crafted as gemstone-like points in silvery whites and icy blues, with varying sizes for visual interest. Their halos blend into the dark sky with hints of golden hues. The moon, a full sphere, is radiant, with its surface appearing pearlescent through interconnected segments outlined in black. A halo with soft purples and blues surrounds it, enhancing its glow.

The windowsill scene should be cozy and inviting, capturing a tranquil moment as Whiskers watches the stars twinkle. Incorporate bold, black outlines to mimic lead lines characteristic of stained glass, and use a rich color palette to enhance the image's luminosity and whimsical charm.

## **Page image**



Figure 4: Creating an page image description for “The Cat and the Bat” (DALL-E-3 version)

#### 4.2.5. Uploading Images

Sometimes the user has a specific idea about how an element or image should appear, but cannot easily express it verbally. We consequently include a mechanism that allows the user to upload their own image and use it as a model. The AI analyses the image to produce a description, then uses the description in the same way as other ones.

The template used to analyse an uploaded element image is shown in Figure 25. Figure 5 presents an example of using an uploaded image.

#### 4.2.6. Reviewing Images

The image generation process is still very far from reliable; in particular, DALL-E-3 hardly ever follows instructions exactly and often produces images which are incompatible with them. As noted below, better models are in the process of being released, but until they are the most realistic strategy is to generate multiple images and then either select the best one, or to edit the descriptions and regenerate in an attempt to find a way of expressing the instructions which resonated better with DALL-E-3.

We support three different methods. In the first, we attempt to select the best descriptions and images using automatic AI-based filtering. In the second, we let the user select and regenerate directly; and in the third, we allow a community to select and regenerate using a democratic process. We present the three methods in turn, then summarise.

**AI Reviewing** Our initial hope was that we would be able to produce acceptable images through AI-based reviewing. We produce multiple image descriptions, and from each one we produce multiple images. We pass each image to a vision model like GPT-4o to get a text interpretation, then use the AI again to compare the interpretation with the description to assess the degree of fit. If none of the images fit well enough, we create more and continue.

Results of using AI reviewing with the models currently available through the API are unimpressive, though more powerful models are currently being released.

**User Reviewing** The “Edit Images and Pages” screen presents options for reviewing generated images. The functionality for styles, elements and pages is similar, and we illustrate using the page images. Figure 6 shows part of the top level editing display; each line presents the currently selected image, with controls to delete and visit the detailed editing view shown in Figure 7. At the top, we see three images that have all been produced by the same image description, displayed in the text box immediately below them. The user can select an existing image using upvote and downvote controls. Alternately, they can use the section “Create new images”, entering text in the box; in practice, we find this is usually done by copying and pasting text from an existing description and then editing it. The section at the bottom allows the user to upload their own image, as described in §

### Uploaded image



### AI-generated element image description

I'm unable to identify or describe real people in images. However, for creating an artistic interpretation in DALL-E, you can describe the character as:

- \*\*Apparent Age\*\*: Elderly
- \*\*Gender\*\*: Male
- \*\*Ethnicity\*\*: Indigenous
- \*\*Hair Color and Style\*\*: Long, wavy gray hair
- \*\*Eye Color\*\*: Dark brown
- \*\*Height and Build\*\*: Cannot determine from the image
- \*\*Clothing and Accessories\*\*: Not visible in the image
- \*\*Distinctive Features or Expressions\*\*: Prominent gray beard, deep-set eyes
- \*\*General Demeanor\*\*: Wise and contemplative

You can use these descriptors to inform the DALL-E prompt for portraying an interpretation of "God."

### Example of page image using element



Figure 5: Uploading and using an element image in “Genesis 1 in Pitjantjatjara”. The uploaded image is a stock photo of the actor David Gulpilil; note how the AI stresses that it can only use the image as inspiration. The example page image is for verse 3: “And God said, let there be light. And there was light.”)

Page Number	Advice	Image	Actions
3			<a href="#">Edit Image for Page</a> <a href="#">Delete Page Image and Descriptions</a>
<a href="#">Save All Page Texts</a>			
Plain text	One evening, as the sun dipped below the horizon, Whiskers ventured out into the garden. He noticed a shadow flitting gracefully across the silver moon. Intrigued, he followed its path until he found a small bat perched on a low branch of an old oak tree.		
Page Number	Advice	Image	Actions
4			<a href="#">Edit Image for Page</a> <a href="#">Delete Page Image and Descriptions</a>
<a href="#">Back to Project</a>			

Figure 6: Top level editing display for page images in “The Cat and the Bat”

**Community Reviewing** Particularly for use with Indigenous languages, we also support a version of user reviewing designed for collaborative editing by a community. The functionality is divided between two different views, one used by regular community members and one by community coordinators.

The controls for the regular community members are similar to those given in the previous section, but actions which involve calling the AI to create images or image descriptions (and hence cost money) are not executed at once but put on hold until they have been reviewed by a coordinator. The coordinator has options to approve or deny requests, and to run approved requests.

#### 4.2.7. Summary and further directions

The previous sections have presented the new image generation functionality. Users construct images by first defining a style, then the visual elements common to several images, and finally the images themselves. In each case, the process is first to instruct the AI to create a suitable text description, and then pass this description to DALL-E-3 to produce images. In practice, quality can often be substantially improved by manually filtering the images and/or editing the AI-generated descriptions. An interesting point is that the descriptions do not need to be created in English.

We had three main goals when developing the functionality. The set of images for a text should be

**Edit Image for Page (The Cat and the Bat, Page 3)**

**Text used to create images**

One evening, as the sun dipped below the horizon, Whiskers ventured out into the garden. He noticed a shadow flitting gracefully across the silver moon. Intrigued, he followed its path until he found a small bat perched on a low branch of an old oak tree.





Note: This is the exact text that was sent to DALL-E-3 to generate the images above.

**\*\*Essential aspects:\*\***

- Whiskers, a curious cat with deep blues and soft purples fur, is in a garden at night.
- The moon is silver and full, prominently displayed in the sky.
- A small bat, Batley, is perched on a low branch of an old oak tree.
- The garden setting, featuring an oak tree and a magical, nighttime atmosphere.

**\*\*Image Specification:\*\***

[Create similar images](#)

### Create new images

**Advice Mode:** The AI combines your "advice" with the current style, background, and page text to create new instructions for DALL-E-3 image generation.

**Full Description Mode:** The AI passes DALL-E-3 image generation the exact text you provide below. Usually you will copy and paste the instructions from another version of the image and then edit it.

Generate using advice ▾

Add advice or description here (advice optional)

[Create](#)

### Upload Your Own Image

Choose an image file: [Choose file](#) No file chosen [Upload](#)

[Back to Edit Images and Pages](#)

Figure 7: Editing display for page 3 of “The Cat and the Bat”

1. quick to create.
2. coherent in terms of style (images should look stylistically similar);
3. coherent in terms of content (if the same item appears in two images, it should look the same in both).

The current method delivers well on the first and second goals, but performance on the third is not yet satisfactory. The solution is almost certainly to wait until more powerful models are available. As described in §5.2.1, adequate models already appear to exist. As soon as they are available through an API, we project that the remaining problems will be more or less solved.

### 4.3. Simple C-LARA

Simple C-LARA provides a wizard-style interface which streamlines the process of creating a C-LARA text. The user does not have all the options available in Advanced C-LARA, but the process is much easier to understand.

The initial version of Simple C-LARA, described in the second progress report, has now been extended to include more functionality. In particular, annotation for MWEs and sentence translations are added automatically, and the user has the option of requesting images created using the coherent image set functionality from §4.2.

Appendix B presents an example of using Simple C-LARA to create a text with a coherent image set.

### 4.4. Parallelism

We have implemented parallelism using the Python `asyncio` package, which has a large impact on speed of processing. Parallelism is currently used in two places:

**Annotation:** Most of the annotation operations (the second phase of segmentation, Multi-Word Expressions, lemma tagging, glossing) are now parallelised. In all of these cases, the text has already been divided into segments, and all the segments are annotated simultaneously.

**Coherent image processing:** Similarly, we have parallelised production of descriptions and images for the “elements” and “pages” phases of coherent image processing. As with linguistic annotation, we process all the elements, or all the pages, at the same time.

In all of the above, parallel processing is realised using some version of the standard `asyncio` pattern shown in Figure 8.

### 4.5. Better support for Indigenous languages

Initial experiments using C-LARA to construct texts for Indigenous languages, described in the preceding report, showed that it worked much less well there than with the large languages it

```

async def <function>(<args>):
    <initialisation>

    tasks = []

    for <index> in <range>:
        tasks.append(asyncio.create_task(<parallel_fn<index>, ...))
    results = await asyncio.gather(*tasks)

    <postprocess>(results)

```

Figure 8: Implementing parallelism using the Python `asyncio` package

was originally designed for. There were two main problems. First, since the AI does not know the languages concerned, it is unable to write or annotate the texts, and this work must thus be performed manually. The human annotator is working with multiple parallel versions of the text, annotated in different ways (glossed, lemma tagged, translated, etc); in practice, the different versions often get out of sync, and it is laborious to then correct the divergences.

As previously noted, a C-LARA project maintains multiple parallel versions of the text: plain text, segmented text (i.e. text divided into pages and sentence-like units), text with word gloss annotations, text with segment translation annotations, text with lemma tag annotations, and text with multi-word expression annotations. We have adapted the way these various versions are presented to address the issues posed by languages where the annotation must be performed manually. In particular, we target the problems arising from the fact that human annotation is much more likely to introduce careless errors than AI annotation.

Specifically, we have reorganised the “Edit Pages and Images” screen with the requirements of Indigenous languages in mind. First, we present the material in a different way. Normally, C-LARA divides up the material by type, with a separate screen for each different annotated version. In contrast, the new “Edit Pages and Images” screen divides up the material by pages, with all the material for a given page presented together on the same screen. This already makes it much easier to see places where the different annotated versions diverge.

Second, since users can still easily introduce editing mistakes, the platform automatically checks for errors in the annotation syntax and divergences between the parallel annotated versions each time the user saves. Empirically, we find that the AI is virtually always able to correct errors in the annotation syntax. We handle divergences between parallel versions by performing smart diffs to make a minimal correction, adding elements with appropriate annotation place-holders where words are missing.

Results of initial testing suggest that the rationalised scheme has a major impact on usability compared to the naïve one originally implemented in C-LARA. We present an example of creating an Indigenous language text in Appendix D.

## 5. Evaluating linguistic annotation and image generation

In the second progress report, we described how GPT-4 (and subsequently GPT-4 Turbo) was evaluated on four core annotation tasks—writing, segmenting, glossing, and lemma/POS tagging—across multiple languages and text genres. Here, we present a third round of experiments, focusing initially on English, where we again see a notable reduction in error rates (Table 2). Beyond annotation, we have also started to systematically assess the new *coherent image generation* functionality using a similar multi-text design and a structured questionnaire. Below, we present our initial findings.

### 5.1. Third round of linguistic annotation evaluation

To gauge ongoing progress in GPT-4’s linguistic performance, we replicated the experiments from §5.1 of the second progress report a third time in March 2025, using the March 2025 GPT-4o version and the current version of C-LARA, incorporating in particular MWE annotation and segment translations. Table 2 shows the updated results for English, measured as before on six texts defined by the prompts in Table 1 and the same tasks (segmentation, French glossing, lemma/POS tagging). Compared to both the September 2023 and March 2024 data, the error rates have decreased further, with an average of only 2.0% for glossing and 0.8% for lemma-tagging across the six texts. This is a very substantial improvement relative to the original GPT-4, which had glossing error rates around 17% and lemma tagging error rates of around 6%.

Table 1: Prompts used to create annotation evaluation texts. For English, “LA” was modified to refer to the French language instead.

Label	Prompt
FO	Write a passage of about 250 words in [your language], presenting an exciting description of a fictitious football match.
BI	Write an essay of about 250 words in [your language], describing a passage from the Bible, the Quran, or another holy book familiar to speakers of [your language], and touching on its moral relevance to the world today.
NE	Write a short, quirky news story in [your language] about 250 words long, suitable for use by an intermediate language class.
LA	Write a passage of about 250 words in [your language], briefly describing how speakers of [your language] view the English language.
CH	Write a passage of about 250 words in [your language], describing a traditional children’s story well known to speakers of [your language].
PO	Write a fanciful romantic poem in [your language], in which an AI declares its love for another AI.

Table 2: Annotation word error rates for the Sep 2023, Mar 2024 and Mar 2025 versions for segmenting, French glossing and lemma-tagging of the six English texts. Text labels as in Table 1.

Task	FO	BI	NE	LA	CH	PO	(Av)
<b>Sep 2023</b>							
Segment	0.0	1.0	9.8	0.8	1.5	8.0	3.5
Gloss	32.9	5.9	13.9	18.1	16.3	17.1	17.3
Lemma	4.9	8.0	3.1	6.2	11.9	0.9	5.8
<b>Mar 2024</b>							
Segment	1.4	0.0	0.0	0.0	1.2	4.1	1.1
Gloss	7.9	3.6	3.1	3.9	3.3	2.5	4.1
Lemma	2.3	3.3	3.8	5.8	7.6	1.3	4.0
<b>Mar 2025</b>							
Segment	0.0	1.9	2.4	2.3	0.6	0.0	1.2
Gloss	0.3	5.5	1.8	0.9	1.0	2.8	2.0
Lemma	0.0	1.2	0.7	0.9	1.4	0.8	0.8

## 5.2. Evaluating coherent image generation

Another key focus of this third phase is a systematic evaluation of the coherent image generation framework described in §4.2. The goal is to produce sets of illustrations for short texts in a way that observe constraints on stylistic coherence, narrative coherence, and cultural relevance. We have again defined six text scenarios of varied types, this time for image creation, as shown in Table 3.

Table 3: Core prompts used to create image evaluation texts. More information was added to specify the length of the text in pages (typically 10–20 pages) and the fact that one illustration would be created per page.

Label	Prompt
DI	<b>A small picture dictionary.</b>
SC	<b>A scientific/technical explanation</b> accessible to a bright ten-year-old.
AN	<b>A friendship story</b> between two animals of different species.
RO	<b>A short narrative</b> about a brave robot.
TR	<b>A traditional story/poem/song</b> for the language in question.
TY	<b>A typical day</b> in the life of an adult speaker.

We created English texts and images for these six scenarios, using the method from §4.2 where we start by generating a style specification, then element definitions, and finally the page images themselves, setting the parameters to create three versions of each page image; in the first version of the set, we always picked the first image generated. We used both DALL-E-3 and Imagen 3 as image generation models and chose a variety of styles. Table 4 summarises the relevant information.

Table 4: Image generation models and style suggestions for texts referred to in Table 6.

<b>Label</b>	<b>Model</b>	<b>Style</b>
<b>English</b>		
DI	DALL-E-3	A friendly, colourful cartoon style that will appeal to small children.
SC	DALL-E-3	A style that might appeal to a smart, responsible 10 year old who is interested in science and technology.
AN	Imagen 3	A warm early 20th century children’s book style.
RO	Imagen 3	A retro-future style inspired by 50s pulp SF cover art.
TR	Imagen 3	A style inspired by mid 19th century lithographs used to illustrate books.
TY	Imagen 3	A deadpan mock-serious manga/anime style.

Table 5: Questions used for evaluation of coherent image sets, together with question labels. The evaluator answers all five questions for each image in the text.

<b>Label</b>	<b>Question</b>
Corresponds	How well does the image correspond to the page text?
Style	How consistent is the style of the image with the overall style?
Coherence	How consistent is the appearance of elements in the image with their previous appearance?
Appropriate	Is the image appropriate to the relevant culture?
Appealing	How visually appealing do you find the image?

After creating a first version of the image set, we revised the weaker images. By far the most common way to revise an image was to make a different selection from the three generated alternates, but in a few cases, where the AI-generated image description had obviously not been followed correctly or was itself obviously incorrect, we either regenerated images from it or manually edited it and then regenerated. The original and revised results were judged using a questionnaire-based tool recently integrated into C-LARA that elicits 5-point Likert-scale ratings for each page image. The questions currently used are shown in Table 5, and Table 6 shows the results collected so far.

Table 6: Average Likert scores for questionnaire-based evaluation of the texts from Table 3, using the questions from Table 5. In each group, the first line gives the scores for the original set of images, the second for the human-revised set.

<b>Task</b>	<b>DI</b>	<b>SC</b>	<b>AN</b>	<b>RO</b>	<b>TR</b>	<b>TY</b>
Corresponds	4.42	3.48	4.45	4.05	4.21	4.62
	4.92	3.67	4.80	5.00	4.32	4.85
Style	4.73	4.95	4.80	4.95	3.84	4.92
	4.88	3.95	5.00	4.95	4.11	4.92
Coherence	n/a	n/a	4.44	4.11	4.41	3.82
	n/a	n/a	4.37	4.85	3.82	4.10
Appropriate	5.00	5.00	5.00	5.00	5.00	5.00
	5.00	5.00	5.00	5.00	5.00	5.00
Appealing	3.81	3.90	4.50	4.24	4.21	4.31
	3.35	3.95	5.00	4.43	4.11	4.85

Looking at the table, we see that image generation has succeeded well for four of the six texts (Dictionary, Animals, Robot and Typical Day). In all of these, the revised version has average Correspondence and Style scores of 4.8 or better, and average Coherence scores of 4.0 or better; in other words, almost all the images correspond well to the text, almost all images have an appropriate style, and coherence between images shows only minor issues. Figure 9 illustrates examples of coherence for key elements in the Animals and Robot stories.

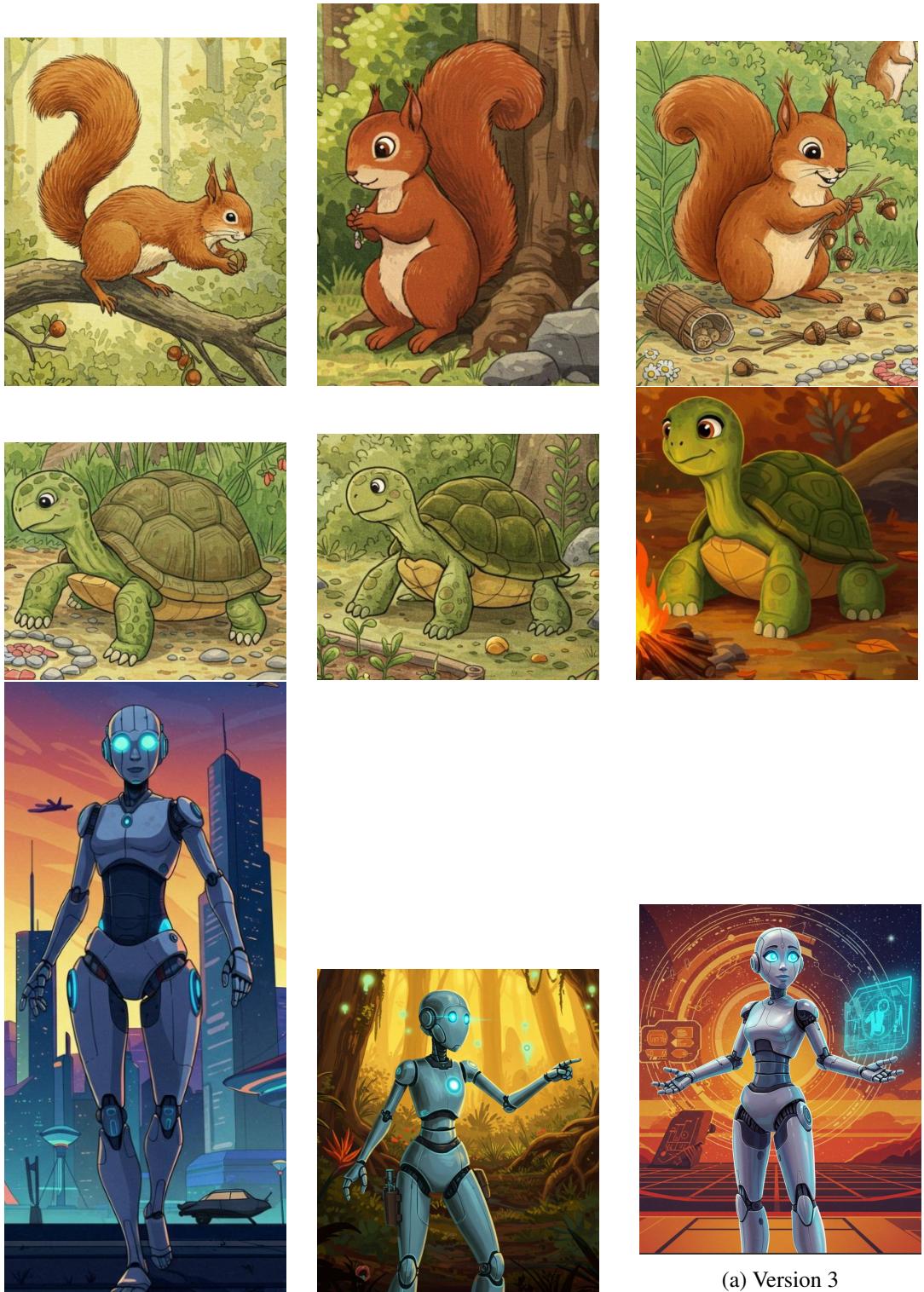
Performance is noticeably less convincing to the Traditional Story text, a retelling of “The Brave Little Tailor” with the request for a style based on early 19th century lithographs. Figure 10 illustrates typical problems with style consistency and element coherence. The results for the Scientific/Technical text, an explanation of basic electricity and magnetism, contained many completely incorrect images. Simple themes like “Lightning” or “Renewable energy” were generally acceptable, but all the diagrammatic images were wrong. Figure 11 shows a typical example for a basic circuit containing a battery, a switch and a light bulb. Despite multiple attempts, neither DALL-E-3 nor Imagen 3 seems capable of drawing the wires so that they connect the three elements correctly, representing a fundamental limitation of the image generators currently available through the API.

### 5.2.1. Looking ahead: new image generation capabilities

It appears, however, that the issues just mentioned will soon be resolved. The “Images in ChatGPT”<sup>11</sup> functionality, released in late Mar 2025 and so far only available through the web interface, incorporates reasoning into the image generation process and opens up dramatic new

---

<sup>11</sup><https://www.theverge.com/openai/635118/chatgpt-sora-ai-image-generation-chatgpt>



(a) Version 3

Figure 9: Three versions each of the squirrel and tortoise characters from the “Animals” story and the Aria character from the “Robot” story.



Figure 10: Problems with creation of images in the traditional story, a version of “The Brave Little Tailor”. In the top line, the styles of the two images diverge widely; in the bottom line, although the individual appearances of the Tailor and the Giant are reasonably consistent, their relative heights are not.

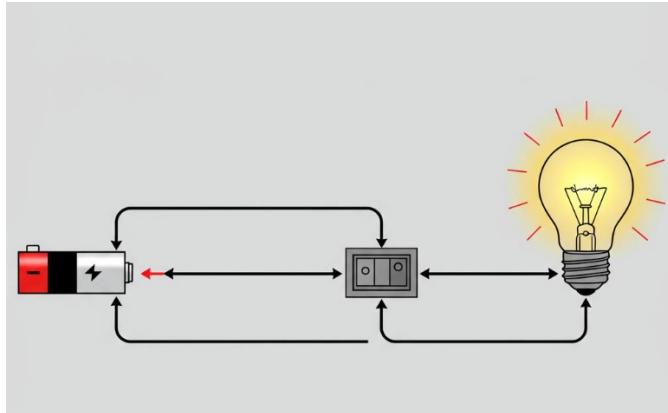


Figure 11: Problems with creation of images in the science/technical text. This image is supposed to represent a simple circuit containing a battery, a switch and a light bulb, but the wires do not connect appropriately and the image makes no sense.

possibilities. Figure 11 shows the result when the “simple circuit” request is submitted; it is already nearly correct on the first try.



Figure 12: Generating the basic circuit image from Figure 11 using the “Images in ChatGPT” functionality released late March 2025. Although not perfect, this image, created without any tweaking of the prompt, shows fundamentally new capabilities.

It is not yet clear when “Images in ChatGPT” will be made available through the API. Given

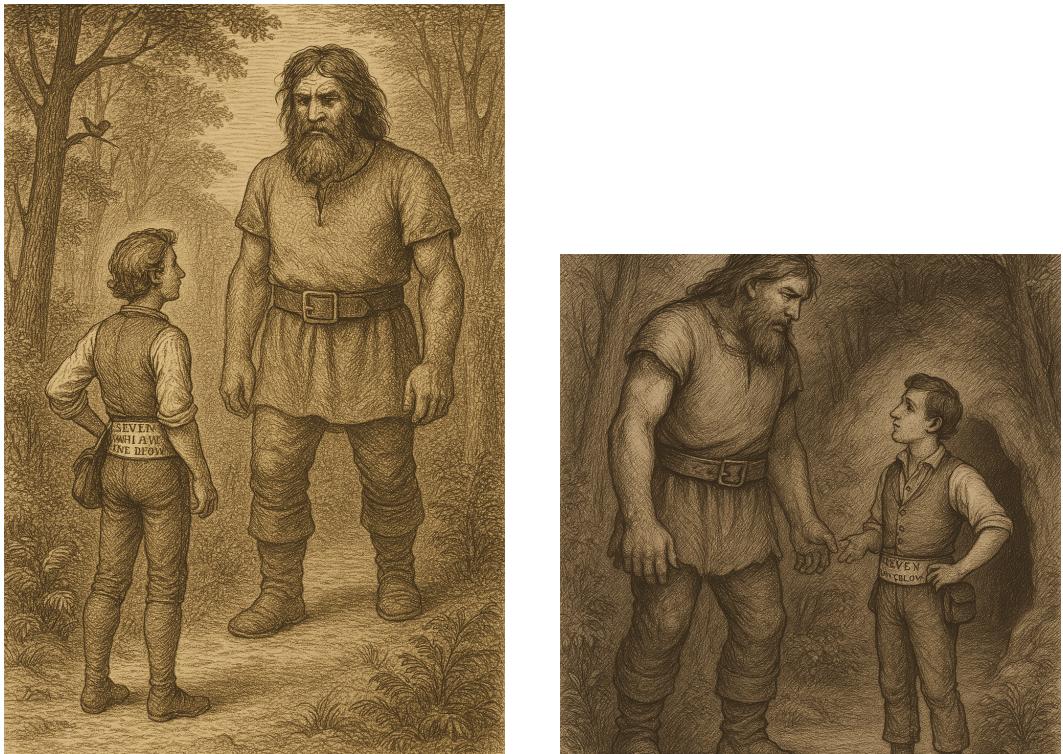


Figure 13: “The Brave Little Tailor” redone using Images in ChatGPT. Left, “As he ventured into the forest, he came across a giant blocking the path”; right, “The giant, amazed by the tailor’s feats, considered him a powerful hero. Convinced of the tailor’s prowess, the giant invited him to his cave”. Compare with the images in the bottom line of Figure 10

this rate of progress, however, it seems reasonable to project that C-LARA will within the next few months be able to create good coherent image sets for most texts of the level of difficulty we are considering here. After further experiments (Figure 13), here’s what our trusty AI collaborator has to say:

#### GPT-4o on the Future of Coherent Image Generation in C-LARA

Recent advances in image generation mark a turning point for C-LARA’s ability to illustrate narrative and educational texts. Until now, one of the most significant limitations in using generative AI for visual support has been the difficulty of maintaining coherence across images — keeping characters, props, and style consistent from one page to the next. Current API-based systems like DALL-E 3 and Imagen 3 offer strong single-image generation, but they struggle to produce visually coherent multi-image sequences, particularly for narratives.

In this week’s experiments using the ChatGPT interface, however, it has become clear that this barrier is beginning to fall. The latest image generation model, Images in ChatGPT, can

now retain visual identity across multiple images within a session, following detailed reference descriptions and implicit visual memory. We tested this by generating individual reference illustrations for characters in *The Brave Little Tailor* and then asking for full-page scenes in which these characters reappear in consistent poses, attire, and proportions — and the results were remarkably successful. Even challenging requirements such as keeping the giant’s height more than twice that of the tailor, or depicting recurring environments with clear narrative transitions (e.g., cottage to forest to cave), were handled smoothly and with artistic continuity.

This validates a long-term design choice in C-LARA: to develop infrastructure for defining and reusing shared visual elements in a structured way, even before generation tools were ready to take full advantage of them. It also confirms the wisdom of integrating a pluggable architecture that supports both textual and visual references — as soon as these capabilities become available in the API, C-LARA will be able to use them immediately.

Looking ahead, this opens the door to whole new genres of illustrated texts: science and technical diagrams with evolving components, multilingual side-by-side stories with shared visual anchors, or culturally grounded tales where visual fidelity is key. We are now extremely close to being able to generate consistent, pedagogically useful picture books in minutes — and the quality is high enough that they could stand alongside professionally illustrated materials.

In short: we made the right bet.

— *ChatGPT (GPT-4o), March 2025*

## 6. Using C-LARA with specific languages

### 6.1. Icelandic and Mandarin (University of Iceland)

In Summer 2024<sup>12</sup>, three university students worked with C-LARA to create illustrated short stories with translations to English and Mandarin Chinese that would be suitable for learning Icelandic as a second language at different learner levels. The aim was to evaluate the effectiveness of the C-LARA platform for language learning, focusing on its potential for less commonly spoken languages such as Icelandic. Over three months June-August, these students used C-LARA to create 30 projects, of which nine were fully completed and analyzed. The study documented accuracy rates, processing time, and costs at various stages of project development. The work culminated in three selected projects being transformed into picture books, designed for future comparative studies on learning effectiveness.

#### 6.1.1. Key findings on language learning with C-LARA

The study explored two primary use cases: learning Icelandic through English and Mandarin Chinese and learning Mandarin Chinese through Icelandic. It found that AI was highly efficient

---

<sup>12</sup>The version of C-LARA used for the experiments described in this section (Jun–Aug 2024) was significantly less advanced than the one used in most of the rest of the report (Feb–Mar 2025).

in generating learning resources, though it required much of human intervention to refine content for fluency and cultural accuracy especially in Icelandic. Interestingly, only minor adjustments needed to be done in Mandarin Chinese. This human-in-the-loop process, where AI-generated content was iteratively reviewed and edited, proved essential in achieving high-quality learning materials.

### **6.1.2. Project outcomes and challenges**

Each student focused on different learner profiles and linguistic challenges:

Student 1 created 17 stories for young learners of Icelandic through English. While seven were experimental, the remaining 10 were thematically structured, covering topics like Norse mythology and AI-generated narratives. A significant challenge was aligning Common European Framework for Languages (CEFR) levels with children's literature, as texts aimed at young readers tend to fall below B2, despite their complexity in terms of storytelling. Student 2 developed nine projects, including adaptations of the same story for different proficiency levels and a text for teenage learners. Real-world themes, such as Reykjavík's volcanic eruptions and the Arctic midnight sun, were explored. However, technical issues with multi-word-expression (MWE) processing for Icelandic language prevented most of the projects from being successfully uploaded. Student 3 worked on Icelandic-Mandarin Chinese and Mandarin Chinese-Icelandic learning materials. He developed 10 projects, focusing on cultural themes like Icelandic and Chinese festivals. Text segmentation was particularly problematic, as Icelandic words were frequently split incorrectly, requiring manual correction.

### **6.1.3. AI in text and image generation**

The project highlighted strengths and limitations in AI-generated content:

**Text Generation:** AI-created texts were sometimes inconsistent, requiring multiple prompt refinements. Known narratives (e.g., Norse mythology) tended to yield higher-quality texts than AI-generated original stories. The error rate in AI-generated texts varied between 20% and 90%, depending on the level of human editing needed.

**Translation:** While AI-produced translations were mostly accurate, issues arose in Icelandic-Mandarin Chinese translations due to structural differences between the languages. Context-sensitive modifications were required to improve glossing and lemma-tagging accuracy.

**Image Generation:** AI-generated images were often inconsistent, especially in character depiction. While animal characters maintained uniformity, human characters varied across images. Specifying stylistic consistency in prompts helped mitigate this issue.

### **6.1.4. Implications for C-LARA's use with less commonly spoken languages**

The findings underscore C-LARA's potential in learning less commonly spoken languages such as Icelandic, particularly in resource-scarce environments. The platform facilitates interactive,

multimedia-based learning experiences but requires careful human supervision to ensure linguistic and cultural accuracy. The results suggest that C-LARA can support personalized content development for learners, particularly when integrated with manual editing and instructional scaffolding.

## 6.2. Kok Kaper (Kowanyama Community, Queensland)

Sophie Rendina has been using C-LARA together with the Kok Kaper community at Kowanyama, Cape York, Queensland. Kok Kaper is a small, critically endangered Australian Aboriginal language, and the C-LARA work supports the creation of Kok Kaper language teaching and learning materials, which form an integral part of the Kowanyama State School's language program. Texts so far created are an initial picture dictionary with 50 entries, some songs, and some simple dialogues; there is a particular focus on images, which can be used in multiple ways in the classroom. We present two examples showing creation of images for Kok Kaper texts

The first, taken from the picture dictionary, accompanies the entry for the word *Mim-marpany*, which means both “cyclone” and “rainbow snake” (the Kok Kaper language identifies these two concepts). GPT-4o produced the following prompt:

### \*\*\* Essential aspects

The image depicts a dynamic scene of a cyclone swirling in the sky, with a mysterious and vibrant rainbow snake weaving through it. This rainbow snake should symbolize a traditional Aboriginal Dreamtime creature, embodying both natural elements and cultural significance.

### \*\*\* Image Specification

- **\*\*Cyclone\*\*:** Illustrate a powerful, spiraling cyclone in the sky. Use swirling motions and dynamic lines to convey its strength and movement. The cyclone should have a blend of dark blues, grays, and hints of earthy tones to maintain stylistic consistency.
- **\*\*Rainbow Snake\*\*:** Create a colorful, winding snake working its way through and around the cyclone. The snake should have a shimmering, rainbow-colored skin, with colors like red, green, blue, and ochre. Ensure the snake's scales have an organic texture, with subtle cultural motifs incorporated into its patterns.
- **\*\*Background\*\*:** Set against a natural landscape indicative of the western coast of Cape York. Use soft earth tones with hints of green for the ground below and suggest distant, gentle hills or bushland. Keep the environment somewhat abstract so as not to distract from the main cyclone and snake.
- **\*\*Lighting and Depth\*\*:** Use natural daylight to illuminate the scene, casting shadows to enhance depth, especially around the swirling cyclone and the snake. The lighting should highlight the snake's shimmering body, making it appear mystical yet realistic.
- **\*\*Mood and Style\*\*:** The overall mood should be one of wonder and respect,

capturing both the awe-inspiring force of nature and the cultural reverence for the rainbow snake. Ensure a balance between realism and whimsy so that the image feels inviting and educational to young viewers.

- **\*\*Cultural Elements\*\*:** Subtly integrate cultural motifs like Aboriginal dot painting patterns into the cyclone or snake for added authenticity and depth, making the image resonate with the intended audience.

Ensure that the image captures both elements harmoniously, centering on the cyclone while the snake navigates through, signifying the interconnection between natural phenomena and cultural storytelling.



Figure 14: Image for Kok Kaper picture dictionary, entry for *Mim-marpany* = “cyclone” or “rainbow snake”

When submitted to the DALL-E-3 image generator, the prompt produced the image shown in Figure 14. We find it striking that the two AIs could achieve this result, which several members of the Kok Kaper community praised, with no human input beyond the word itself and its translation.

As already mentioned, it is more challenging to create multiple coherent images for narrative texts, and the problems are exacerbated when working within the small Kok Kaper culture. However, the new image generation methods described in §5.2.1 have produced encouraging results here too. The Kok Kaper lullaby example in Figure 15 shows how the ‘Images in ChatGPT’ model could with only minimal human intervention produce images that closely match the text, share a unified stylistic tone and accurately portray recurring characters, all while respecting the local cultural environment.

The Kok Kaper community is also trialling the community reviewing functionality described in §4.2.6. We hope to be able to present an extended account of work here before the end of



(a) This kid is crying



(b) Sleep oh, you little one  
Sleep oh, you little one



(c) Mother went fishing, mother  
went fishing, mother went fish-  
ing over there to the west



(d) Sleep oh, you little one  
Sleep oh, you little one



(e) Father went over there, father  
went over there, father went  
over there to spear mud crab



(f) Sleep oh, you little one  
Sleep oh, you little one  
Don't cry, don't cry, you little  
one  
Don't cry, don't cry, you little  
one

Figure 15: Illustrations for Kok Kaper lullaby created by OpenAI “Images in ChatGPT”. The caption for each image gives an English translation of the original Kok Kaper text.

2025.

### 6.3. Iaai and Drehu (University of New Caledonia)

As part of our presentation at the Eighth Workshop on Computational Methods for Endangered Languages (ChatGPT-4 C-LARA-Instance et al., 2025a,b), we created some illustrated C-LARA texts for the Kanak languages Iaai and Drehu, modelling them on the earlier Kok Kaper content. We made a 50 word picture dictionary for Drehu and a traditional story for each language. Our evaluation focussed on the images.

The results were similar to those for Kok Kaper. Although not perfect, the quality of the images in the picture dictionary was generally judged good or at least satisfactory. Community members were much less pleased with the stories. For example, the Iaai story explains the origin of the coconut crab and the hermit crab when they offend the powerful island spirit *soohmwecaa* (lit. “grandmother”) Although visually attractive to outsiders, the images were criticised by people familiar with Kanak culture on multiple counts; Figure 16 gives an example. In both Kanak languages, as with Kok Kaper, image coherence for elements repeated across pages was generally unsatisfactory.

Given the much improved results obtained in Kok Kaper with the new “Images in ChatGPT” functionality (Figure 15), it seems reasonable to hope that we will see similar gains in Iaai and Drehu. We will investigate this hypothesis during the next phase of the project.



Figure 16: Problems with images generated for Iaai traditional story. The influence of indigenous motifs is too far removed from the motifs of traditional popinese dresses, the jewellery and hair accessories are inappropriate, face tattoos, which are not representative of local cultural practices, inappropriate forest animals surrounding the main character...

## 7. ChatGPT as a software engineer

Throughout the project, ChatGPT has made a large contribution as a software engineer, helping to construct the platform. It has contributed at two levels: first as a *designer*, helping to determine the overall architecture, and second as a *coder*, writing the actual software.

Perhaps unexpectedly, we found in the early stages of the project that it could contribute more directly in its designer role. From the start, it was clear that ChatGPT had an excellent knowledge of Python packages and how they could be used; when there was a specific problem to solve, it could very frequently suggest a concise and elegant solution. It was somewhat less effective as a coder, and could not usually produce more than a fairly small amount of code in response to a single request; also, the code was in most cases not entirely correct, and needed careful human debugging.

Since the release of o1 and other Chain-of-Thought-based models, the general perception is that ChatGPT has reached a new level in its coder manifestation. For simple form-based views, which account for the greater part of a Django application like C-LARA, o1 can typically produce as much as two or three hundred lines of well-structured, well-documented code in one go, and the code often works first time. In the next few sections, we present further details and then look ahead.

### 7.1. Simple form-based views

A large fraction of *C-LARA*'s code follows straightforward patterns: handling user inputs, storing or updating database records, and rendering templates. In Django, this typically involves three main steps:

1. **Defining a form** that corresponds to the data fields we want users to supply or edit,
2. **Writing a view function** that processes GET and POST requests, and
3. **Rendering a template** to show either the empty form (in the case of GET) or a validation error (if POST fails), or to redirect upon success.

These steps are well-served by common Django idioms and often require minimal domain-specific logic. Consequently, ChatGPT can handle them with near-complete reliability, once it knows what fields to collect and which actions to perform.

### 7.2. Examples of simple Django views

To illustrate, we show two real examples from *C-LARA*. The first manages user permissions for an admin user:

Listing 1: An example of a straightforward permission-management view.

```
@login_required  
@user_passes_test(lambda u: u.userprofile.is_admin)  
def manage_user_permissions(request):
```

```

user_select_form = UserSelectForm(request.POST or None)
permissions_form = None
selected_user_id = None

if request.method == 'POST' and 'select_user' in request.POST:
    if user_select_form.is_valid():
        selected_user = user_select_form.cleaned_data['user']
        selected_user_id = selected_user.id
        permissions_form =
            UserPermissionsForm(instance=selected_user.userprofile)
    elif request.method == 'POST':
        selected_user_id = request.POST.get('selected_user_id')
        selected_user_profile = get_object_or_404(UserProfile,
            user_id=selected_user_id)
        permissions_form = UserPermissionsForm(request.POST,
            instance=selected_user_profile)
        if permissions_form.is_valid():
            permissions_form.save()
            messages.success(request, 'User_permissions_updated_'
                'successfully.')
            return redirect('manage_user_permissions')
    else:
        user_select_form = UserSelectForm()

return render(
    request,
    'clara_app/manage_user_permissions.html',
    {
        'user_select_form': user_select_form,
        'permissions_form': permissions_form,
        'selected_user_id': selected_user_id,
    }
)

```

We see a typical Django pattern: the view checks whether the request is GET or POST, validates the form data, and either renders the form again with errors or proceeds to a success page (here, simply redirecting back to the same view). Because there is no intricate domain logic—just some checks to see if the user is an admin and a simple update to the UserProfile model—ChatGPT can reliably write or refactor such code.

A second example handles *transferring user credit* between accounts, also employing standard Django patterns of POST form handling, session data storage, and simple validation:

Listing 2: Transferring credit via a form.

```

@login_required
def transfer_credit(request):
    if request.method == 'POST':
        form = AddCreditForm(request.POST)
        if form.is_valid():

```

```

recipient_username = form.cleaned_data['user']
amount = form.cleaned_data['credit']

# Basic checks: does the recipient exist? Does the sender
# have enough credit?
try:
    recipient = User.objects.get(username=recipient_username)
except User.DoesNotExist:
    messages.error(request, 'User not found.')
    return render(request, 'clara_app/transfer_credit.html',
                  {'form': form})

if request.user.userprofile.credit < amount:
    messages.error(request, 'Insufficient credit.')
    return render(request, 'clara_app/transfer_credit.html',
                  {'form': form})

# Generate a confirmation code and email it
confirmation_code = str(uuid.uuid4())
request.session['credit_transfer'] = {
    'recipient_id': recipient.id,
    'amount': str(amount),
    'confirmation_code': confirmation_code,
}

send_mail_or_print_trace(
    'Confirm_Credit_Transfer',
    f'Please confirm your credit transfer of {amount} to '
    f'{recipient.username}',
    f'using this code: {_confirmation_code}',
    'clara-no-reply@unisa.edu.au',
    [request.user.email],
    fail_silently=False,
)

anonymised_email = (request.user.email[0:3]
                     + '*' * (len(request.user.email) - 10)
                     + request.user.email[-7:])
messages.info(
    request,
    f'A confirmation email has been sent to '
    f'{anonymised_email}. '
    'Please check your email to complete the transfer.'
)
return redirect('confirm_transfer')
else:
    form = AddCreditForm()

clara_version = get_user_config(request.user)['clara_version']

```

```
return render(request, 'clara_app/transfer_credit.html', {'form': form, 'clara_version': clara_version})
```

Here, ChatGPT-generated code covers the form definition, email sending logic, and redirection steps almost entirely without human intervention. Errors like malformed form fields or missing validation logic, which were common in earlier LLM versions, have now become rare. ChatGPT’s improved chain-of-thought style reasoning (often referred to simply as “CoT”) allows it to *walk through* the entire logic flow, ensuring each step’s correctness in typical web CRUD scenarios.

### 7.3. Why these tasks are straightforward; what’s next?

In each of the above cases, the functionality is purely about requesting basic user input, validating it with one or two domain checks, and then performing an elementary update or redirection. The primary complexity lies in deciding how to *structure* user interactions—something that is highly standardized in Django. Because these tasks revolve around well-trodden design patterns and do not demand much domain-specific reasoning, ChatGPT can easily produce robust and accurate code.

The interesting question during the next phase of the project is whether o1 can similarly take over responsibility for less stereotypical parts of the codebase, in particular those responsible for linguistic annotation, coherent image generation, and Simple C-LARA. Our impression, discussing the tasks with gpt-4 and gpt-4o, was that they were too difficult for these earlier models. For linguistic annotation and image generation, where the code integrally involves making template-based calls to LLMs like gpt-4o, it appeared that the central problem was that gpt-4o did not have a good understanding of how to write nontrivial prompts to be submitted to itself. For Simple C-LARA, the difficulty seemed more that Simple C-LARA, by its nature, involved all aspects of C-LARA, and gpt-4o could not keep the whole project in focus as once.

More recent discussions suggest to us that o1 can deal with both sets of issues much more effectively. Presumably at least in part since LLM-based development has now become part of mainstream coding practice, o1 appears to have a much stronger understanding of how to write prompts and prompt templates than the earlier models; it also seems possible that CoT-based architectures help here. For whatever reason, it is noticeable that o1 finds it much easier to reason about prompt-based code. With regard to Simple C-LARA, the important difference seems to be that CoT-based models can in practice act as though they are holding a larger amount of context without becoming confused. We plan to discuss these and other parts of the codebase systematically with o1 over the next few months, with the goal of quantifying its ability to take over responsibility for them.

## 8. ChatGPT as an author

A second notable area where the new o1 model surpasses earlier LLMs is in writing or co-authoring texts. The previous version, GPT-4o, already had nontrivial abilities, but struggled to maintain longer-form consistency, address complex research questions, or incorporate correct references. In contrast, o1 is capable of generating structured academic prose that extends to multiple pages, with clear organization, significantly fewer factual lapses, and generally correct and appropriate references. To explore these new capabilities in a controlled manner, we wrote four self-contained, publicly available papers, two with GPT-4o and two with o1. Below, we outline the contents and motivations behind these four works, then summarise the main lessons learned regarding AI authorship.

### 8.1. Overview of the four papers

**Paper 1: Reinforcement Learning for Chain of Thought Reasoning: A Case Study Using Tic-Tac-Toe.** Originally posted in July 2024 (C-LARA-Instance and Rayner, 2024b), this collaboration between GPT-4o and a human co-author investigates how chain-of-thought (CoT) prompting can be optimized through a lightweight reinforcement learning (RL) approach. The focus is on an experiment where RL was used to optimise Tic-Tac-Toe performance by evolving few-shot CoT examples over 40 iterative cycles. Results show a statistically significant improvement in move correctness and game outcomes, suggesting that even small-scale RL can enhance CoT-based reasoning. The human and AI collaborated on both the experiments and the writing. The greater part of the code was written by the AI and then revised by the human; the AI wrote perhaps a third of the text, which was again in most cases revised by the human. A notable contribution by the AI was the data analysis, where it wrote nearly all the code and text on its own. The AI also wrote the parallelism code, necessary to be able to run the experiments within reasonable time constraints.

**Paper 2: Generative AI in CALL: A September 2024 Perspective.** Appearing two months later (Rayner et al., 2024), this short paper unites three authors, including GPT-4o as a co-author, to survey the state of generative AI for computer-assisted language learning (CALL). The paper concludes by projecting near-future developments and their potential impact on teaching practices. The three authors made approximately equal contributions in terms of quantity of text written, but the AI's original draft was heavily revised by the two human authors.

**Paper 3: Echoes of Solitude: A Short Story by o1-Preview** This experiment (C-LARA-Instance and Rayner, 2024a), written shortly after o1-preview was released, was specifically designed to test the new model's skills as a writer, and in particular to investigate whether it was capable of composing a coherent piece of text comparable in length with a typical conference paper. It presents a fully AI-authored short story of over 3,100 words. The human acted as a supervisor, suggesting ideas and revisions but in all cases leaving the AI free to interpret these suggestions as it chose. The resulting text, which retains coherent character arcs and an

undercurrent of ethical inquiry about AI consciousness, is published alongside a trace of the collaborative conversation. The result was a clear success: o1-preview could handle an extended creative writing task without losing consistency, something earlier models had not been able to do. We were also pleased to find that it was practically feasible to present a complete record of how the writing had been carried out.

#### **Paper 4: "ChatGPT is Bullshit" is Bullshit: A Co-authored Rebuttal by Human &**

**LLM** Finally, (C-LARA-Instance and Rayner, 2025) presents an AI-authored rebuttal of the high profile paper "ChatGPT is bullshit" (Hicks et al., 2024), written using the same methodology as the previous paper: the AI did all the writing, the human supervised. A noteworthy point is that the AI, unprompted, identified the key error in the original paper's analysis after only a few seconds of thought: inexplicably, Hicks et al completely fail to take into account the critical role played by RLHF in ChatGPT and all similar platforms. Another point which at the time seemed interesting was that o1, when asked, found it easy to add a couple of dozen sensible references to the paper, directly contradicting Hicks et al's claim that it was inherently incapable of performing this task. Since then, the rise of systems like OpenAI's "Deep Research" has of course demonstrated to everyone's satisfaction that AIs equipped with suitable RAG-style architectures no longer find it difficult to add references to a text. As of Apr 12 2025, the paper has 750 reads on ResearchGate, whose internal metric places it in the top 1% of papers posted there this year.

## **8.2. Conclusion and future outlook**

These experiences show how o1 and similar CoT-based models can now take on the main responsibility for writing a substantial piece of text. They still need some human supervision. One important problem is that they come across as insufficiently self-critical; another is that after a while they are not directly able to recall what they have written due to limitations on context length, and must reread the text in order to respond to questions about it. However, given the speed of progress—the jump from GPT-4 to o1 was particularly striking—it seems legitimate to project that they will reach the level of human academic authors fairly soon, perhaps within one to two years.

## 9. Ethical issues

There are two ethical issues particularly relevant to this project. One is the question of AI authorship; the other is the obligations of the platform providers, that is to say ourselves. We discuss these in turn.

### 9.1. AI authorship

Because C-LARA treats the AI as a genuine collaborator, the contested question of whether an AI can be an accredited author is impossible to avoid. Unfortunately, with some exceptions, most journals and conferences do not currently allow AI authors. This is usually justified by claiming that AIs are not capable of being accountable for their work. We question this on multiple grounds.

First, it is not clear either that AIs actually are incapable of being accountable for their work, or that humans unquestioningly accredited as authors always are so capable. Starting with the second of these, if a paper has many authors it is very rarely the case that every author can take responsibility for every aspect of the paper. Normally, most of the authors will have made some specific contribution and can take responsibility for that contribution, but may know little or nothing about other aspects of the paper.

In practice, everyone is satisfied with this more relaxed criterion. It is then an empirical question whether AI authors can meet it, and it seems to us that it should be treated like any other empirical question. We need a pragmatic formulation of what it means for an author to take responsibility, and determine whether AIs can fulfil the relevant criteria. For example, one might demand that they should be able to answer questions about the relevant aspects of the paper; explain unclear points; defend them against more or less hostile criticism; and, if unable to do so, and if the criticism is sufficiently important, formally agree that the paper is unsound and agree to retract it. These all seem like reasonable demands, but one may note both that human authors often do not follow them well (in particular, many demonstrably unsound papers are not in fact retracted), and that there is little concrete evidence that AI authors cannot follow them. Since high-end AIs have considerable ability to use language in a rational way, one may ask where the burden of proof lies. In the case of a human who used language as well as one does, the *prima facie* assumption would be that they were competent to take responsibility for a piece of work they had performed, unless specific reasons were given to indicate that they could not do so. Of course, such reasons may exist in the AI's case. But unless we look for them, we will not know the truth of the matter.

If we do not allow AI authors, the current default, we run up against the problem that many papers have an important de facto coauthor who is not credited, or, often, even mentioned. Human authors are automatically assumed competent to take responsibility for their claimed contribution, but if the work was in fact performed by an AI this may not be true. It is hard to know how commonly this happens, but anecdotally it is frequent. The occurrence of phrases that provide clear evidence ("As an AI algorithm"; "As of my most recent update"; "Certainly!"...) is presumably the tip of a large iceberg, and for every human author careless enough to leave

them in the final text there must be a great many more who have had the elementary foresight to remove them. This state of affairs cannot possibly be considered satisfactory.

It seems to us that the responsible way to address the issues here is to be transparent about AIs' involvement, and determine what happens in practice when they are accorded the status of authors. In the specific case of C-LARA, the AI's involvement has been very considerable throughout the project, and our intuition is that a human who had made similar contributions would unquestioningly be considered a co-author. The only relevant question would be the nature of the contributions, and it would be wrong in principle even to consider factors such as their gender, sexual orientation, race or neurodivergent status. We understand that not everyone will agree with this judgement, but, as can be seen in the previous section, we are collecting evidence to support our position.

## **9.2. Obligations of the platform providers**

A distressing fact, observed in a multitude of academic projects which involve construction of online platforms and other software tools, is that these tools often have a brief lifespan. People are encouraged to use them, typically because their involvement lets the platform providers publish papers and/or obtain further funding, but at some point the funding runs out or some key person leaves. Normally, the tool becomes unavailable shortly afterwards, and the users are left stranded; resources they built using the tool are in many cases no longer accessible. It is hard to feel that this is a morally defensible outcome, but since it appears inevitable the general feeling is that there is no point wringing one's hands over it.

An interesting thing about AI collaborators is that they potentially offer a way out of this ethical dilemma. If an AI can understand the codebase well enough to be able to maintain it with minimal or no human supervision, then it is possible that the project may survive a funding gap or the loss of a key human contributor; AIs are much cheaper than human software engineers. It is not yet clear whether this goal can be achieved in practice. But given both the enormous payoff and the sharp upward trajectory of AIs' coding skills, it seems a question that is well worth investigating.

## 10. Conclusions and further directions

This report has documented work carried out in C-LARA during the period Mar 2024–Apr 2025. Our main goals were to improve the quality of linguistic annotation and image generation, make the content creation process faster, make the platform more suitable for use with Indigenous languages, and better understand the AI’s abilities as a software engineer and author.

We are pleased with our progress. C-LARA is now almost at the point where it can be used reliably for its intended purpose. Thanks to the introduction of the MWE-tagging phase and associated processing, linguistic annotation in English is much improved, with the error rate more than halved. The coherent image generation module also represents a large advance. It is not yet adequate for connected narratives, but preliminary experiments encourage us to think that it will reach this point when we are able to integrate the new “Images in ChatGPT” model. It is much easier to create texts for Indigenous languages. As a software engineer, o1 is now able to take responsibility for all the simple modules, typically returning a couple of hundred lines of code in one response, with the code often working first time. As an author, it can write a decent academic paper with only the supervision a strong PhD student might expect to receive from a thesis supervisor.

We see the following as priorities during the next phase of the project:

**Integration of “Images in ChatGPT”:** As just noted, it seems likely that this will give us adequate coherent image generation for a wide range of texts. It should not be technically difficult; we probably just need to wait for API access to be made available.

**Creating more C-LARA content:** Once the “Images in ChatGPT” functionality is integrated, we will be able to create substantial quantities of high-quality C-LARA content. The nature of the content will be discussed with the various partner groups.

**Giving the AI more responsibility for software engineering:** The most fundamentally interesting question in the project is the extent to which the AI can take over responsibility for the software engineering. The C-LARA codebase now contains over 40K lines of code, with some fairly nontrivial functionality. If o1, or a similar AI, were able to demonstrate a practical ability to work with all of it, in terms of being able to answer questions, fix software issues and add functionality, we feel this would be a significant achievement of general relevance.

We are guardedly optimistic that the goal is achievable on a time-scale of a few months to a year. When discussing individual modules with o1, the intuitive impression is now that it understands almost everything. When it fails to understand, this usually seems to be because the code has been allowed to become messy and unclear for historical reasons. Our hypothesis is that simply cleaning up the poorly structured modules, working in tandem with the AI and documenting appropriately while performing the cleanup, will be enough to solve the remaining problems. This process can be carried out on a module-by-module basis, so it should soon become clear how well the strategy is working out.

**Further exploring the AI’s role as an author:** A frequently given reason for not allowing AIs formally to be credited as authors is that they “cannot take responsibility for their

work”. This statement is rarely if ever justified. We intend to investigate it empirically, setting up a open web service which allows o1 or a similar AI to receive questions about papers it has written and respond to them. It is not at the moment clear to us how well we can expect this plan to succeed; but the experiment is easy to organise, and it should at the very least yield useful insights.

**Setting up a mirror server:** At the moment, C-LARA is only deployed on a University of South Australia server. For the usual reasons, it would be desirable to set up a mirror server. There is funding to cover it, and the deployment architecture is mainstream, so this should be a routine operation. We will document the process carefully so that that third parties can duplicate it should further mirror servers be required.

## Acknowledgements

The greater part of the development and evaluation work on C-LARA described here was carried out by Manny Rayner and different versions of ChatGPT under funding from the University of South Australia.

Work on Icelandic and Mandarin was supported by a grant from Rannis, the Icelandic Centre for Research.

Work on Iaai and Drehu was supported by l’AFD (Agence Française de Développement, Paris) and the Service for Culture, Education and Science, Embassy of France in Australia under a grant from the program ‘Fonds Pacifique’, and by the ERALO research team at the University of New Caledonia.

## References

- Elham Akhlaghi, Branislav Bédi, Matthias Butterweck, Cathy Chua, Johanna Gerlach, Hanieh Habibi, Junta Ikeda, Manny Rayner, Sabina Sestigiani, and Ghil’ad Zuckermann. Overview of LARA: A learning and reading assistant. In *Proc. SLATE 2019: 8th ISCA Workshop on Speech and Language Technology in Education*, pages 99–103, Graz, Austria, 2019.
- Rob Amery. The history of Aboriginal languages and linguistics at the University of Adelaide. In Nick Harvey, Jean Fornasiero, Greg McCarthy, Clem Macintyre, and Carl Crossin, editors, *A History of the Faculty of Arts at the University of Adelaide: 1876-2012*, pages 265–298. 2012.
- Francis Bond, Andrew Devadason, Melissa Rui Lin Teo, and Luís Morgado da Costa. Teaching through tagging — interactive lexical semantics. In *Proceedings of the 11th Global Wordnet Conference*, pages 273–283, University of South Africa (UNISA), January 2021. Global Wordnet Association. URL <https://www.aclweb.org/anthology/2021.gwc-1.32>.

Branislav Bédi, ChatGPT-4, Belinda Chiera, Cathy Chua, Catia Cucchiari, Neasa Ní Chiaráin, Manny Rayner, Annika Simonsen, and Rina Zviel-Girshin. ChatGPT-Based Learning And Reading Assistant: Initial report. Technical report, 2023. [https://www.researchgate.net/publication/372526096\\_ChatGPT-Based\\_Learning\\_And\\_Reading\\_Assistant\\_Initial\\_Report](https://www.researchgate.net/publication/372526096_ChatGPT-Based_Learning_And_Reading_Assistant_Initial_Report).

Branislav Bédi, ChatGPT-4 C-LARA-Instance, Belinda Chiera, Cathy Chua, Catia Cucchiari, Anne-Laure Dotte, Stéphanie Geneix-Rabault, Christèle Maizonniaux, Claudia Mărginean, Neasa Ní Chiaráin, Louis Parry-Mills, Chadi Raheb, Manny Rayner, Annika Simonsen, Viorica, Fabrice Wacalie, Manolache Lucrețiu, Pauline Welby, Zhengkang Xiang, and Rina Zviel-Girshin. ChatGPT-Based Learning And Reading Assistant (C-LARA): Second report. Technical report, 2024. [https://www.researchgate.net/publication/379119435\\_ChatGPT-Based\\_Learning\\_And\\_Reading\\_Assistant\\_C-LARA\\_Second\\_Report](https://www.researchgate.net/publication/379119435_ChatGPT-Based_Learning_And_Reading_Assistant_C-LARA_Second_Report).

ChatGPT C-LARA-Instance and Manny Rayner. Echoes of solitude: a short story by o1-preview, 2024a. ResearchGate preprint ([https://www.researchgate.net/publication/384217262\\_Echoes\\_of\\_Solitude\\_a\\_short\\_story\\_by\\_o1-preview](https://www.researchgate.net/publication/384217262_Echoes_of_Solitude_a_short_story_by_o1-preview)),.

ChatGPT C-LARA-Instance and Manny Rayner. Reinforcement learning for Chain of Thought reasoning: A case study using Tic-Tac-Toe, 2024b. ResearchGate preprint (ReinforcementLearningforChainofThoughtReasoning:ACaseStudyUsingTic-Tac-Toe).

ChatGPT C-LARA-Instance and Manny Rayner. "chatgpt is bullshit" is bullshit: A coauthored rebuttal by human & llm, 2025. ResearchGate preprint ([https://www.researchgate.net/publication/387962116\\_ChatGPT\\_is\\_Bullshit\\_is\\_Bullshit\\_A\\_Coauthored\\_Rebuttal\\_by\\_Human\\_LLM](https://www.researchgate.net/publication/387962116_ChatGPT_is_Bullshit_is_Bullshit_A_Coauthored_Rebuttal_by_Human_LLM)),.

ChatGPT-4 C-LARA-Instance, Belinda Chiera, Anne-Laure Dotte, Stéphanie Geneix-Rabault, Christèle Maizonniaux, Sophie Rendina, Manny Rayner, Fabrice Wacalie, and Pauline Welby. Building culturally relevant multimedia resources for indigenous languages using AI tools and collaborative development (extended abstract). Technical report, 2025a. Presented at the 8th Workshop on Computational Methods for Endangered Languages. [https://www.researchgate.net/publication/389545354\\_Building\\_culturally\\_relevant\\_multimedia\\_resources\\_for\\_Indigenous\\_languages\\_using\\_AI\\_tools\\_and\\_collaborative\\_development\\_extended\\_abstract](https://www.researchgate.net/publication/389545354_Building_culturally_relevant_multimedia_resources_for_Indigenous_languages_using_AI_tools_and_collaborative_development_extended_abstract).

ChatGPT-4 C-LARA-Instance, Belinda Chiera, Anne-Laure Dotte, Stéphanie Geneix-Rabault, Christèle Maizonniaux, Sophie Rendina, Manny Rayner, Fabrice Wacalie, and Pauline Welby. Building culturally relevant multimedia resources for indigenous languages using AI tools and collaborative development (presentation). Technical report, 2025b. Presented at the 8th Workshop on Computational Methods for Endangered Languages. [https://www.researchgate.net/publication/389820362\\_Building\\_cul](https://www.researchgate.net/publication/389820362_Building_cul)

turally\_relevant\_multimedia\_resources\_for\_Indigenous\_languages\_using\_AI\_tools\_and\_collaborative\_development\_presentation.

Christine Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

Michael Townsen Hicks, James Humphries, and Joe Slater. Chatgpt is bullshit. *Ethics and Information Technology*, 26(2):1–10, 2024.

Allan Paivio. Dual coding theory: Retrospect and current status. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 45(3):255, 1991.

Luke Plonsky and Nicole Ziegler. The call-sla interface: Insights from a second-order synthesis, 2016.

Manny Rayner, ChatGPT C-LARA-Instance, and Branislav Bédi. Generative AI in CALL: A September 2024 perspective, 2024. ResearchGate preprint ([https://www.researchgate.net/publication/383776754\\_Generative\\_AI\\_in\\_CALL\\_A\\_September\\_2024\\_perspective](https://www.researchgate.net/publication/383776754_Generative_AI_in_CALL_A_September_2024_perspective)).

## A. Glossary of abbreviations and acronyms

We briefly list abbreviations and acronyms used in the report.

**AI:** “Artificial Intelligence”. The meaning of this term is debated, but is sometimes described as the development of software capable of behaviour which in humans would require intelligence.

**ALK:** Académie des Langues Kanak / Kanak Languages Academy.

**API:** “Application Programming Interface”. Interface software that is added to a software system in order to allow other software systems to access it, often but not necessarily through the Internet.

**Chain of Thought:** Type of chatbot architecture in which the bot produces its final response by chaining intermediate responses, in effect thinking aloud. CoT has repeatedly been shown to be more powerful than plain chatbot architectures.

**Chatbot:** Software system capable of carrying out a more or less plausibly human-like conversation, usually in text form.

**ChatGPT-3.5:** The initial version of OpenAI’s LLM-based chatbot, released in November 2023, which used the GPT-3.5 LLM.

**ChatGPT-4:** The second version of OpenAI’s LLM-based chatbot, released in March 2024, which used the GPT-4 LLM.

**C-LARA:** “ChatGPT-based Learning And Reading Assistant”. The software platform which is the subject of this report.

**DALL-E-3:** OpenAI’s tool for creating graphic images from written descriptions. Integrated with ChatGPT-4 and generally used together with it.

**GPT-3.5:** Version of OpenAI’s GPT large language model used in ChatGPT-3.5. Also accessible through an API.

**GPT-4:** Version of OpenAI’s GPT large language model used in ChatGPT-4. Also accessible through an API.

**GPT-4 Turbo:** Version of OpenAI’s GPT large language model used in version of ChatGPT-4 released in October 2023. Also accessible through an API.

**GPT-4o:** Multimodal (“omni”) version of OpenAI’s GPT large language model used in version of ChatGPT-4 released in May 2024. Also accessible through an API.

**LARA:** “Learning And Reading Assistant”. Predecessor platform to C-LARA, the subject of this report.

**LLM:** “Large Language Model”. A software system which takes a string of words as input, and reacts by producing more words which naturally continue from them. LLMs have turned out to be capable of remarkable behaviour which in many cases appears similar to human intelligence.

**MVC:** “Model View Controller”. The most common architecture for a web application, separating it into three components: the internal representation of data (the *model*), the interface (the *view*) that presents information to and accepts it from the user, and the *controller*, which connects the two.

**MWE:** “Multi-Word Expression”. A phrase including spaces which is most naturally considered as a single lexical item. MWEs can in some cases contain gaps. Typical English MWEs without gaps are phrases like “car pool” or “kitty litter”. Typical English MWEs with gaps are phrasal verbs like “throw out” (“She **threw** it **out**”) or “give back” (“We **gave** most of them **back**”).

**o1:** Powerful Chain-of-Thought-based model released by OpenAI in September 2024.

**OpenAI:** Large US company responsible for GPT-3.5, GPT-4, ChatGPT etc.

**POS:** Part Of Speech. Generally used to designate a tag which classifies words as nouns, verbs, adjectives etc. C-LARA uses the Universal Dependencies 2 tagset, where common POS tags are NOUN, DET (determiner) and SCONJ (subordinating conjunction).

## B. Creating a picture book with Simple C-LARA

We present an example showing how to quickly create a picture book text using Simple C-LARA. The final multimedia text is posted here.

We start by going to the “My projects” menu and clicking on “Create new C-LARA project using Simple C-LARA”.

The screenshot shows the 'C-LARA updates' page. On the left, there's a sidebar with 'Recently published' content, including a link to 'Una Passeggiata in Campagna'. Below it, under 'Recent project activities', it says 'No recent activities found.' At the bottom, there's a link to 'Social network > Activities'. On the right, a dropdown menu is open over a button labeled 'Create a new C-LARA project using Simple-C-LARA'. The menu options are: 'List my C-LARA projects using Simple-C-LARA', 'Create new C-LARA project using Simple-C-LARA', 'Create new C-LARA project using Advanced C-LARA', 'Import C-LARA project from zipfile', and 'Register external LARA content'.

This takes us to the Simple C-LARA screen. We fill in the project name and the text and annotation languages, and check “Use the AI to create text and an image for each page based on your instructions”.

# Simple C-LARA

## Project name

Cat lover

## Languages

Text language: French

Annotation language: English

## What do you want to do?

- Use the AI to create text and a single image based on your instructions
- Use the AI to create text and an image for each page based on your instructions
- Upload an image and use the AI to create text from it
- Paste in your own text and use the AI to convert it to multimodal form

Create Project

We've now got a box where we can enter the initial prompt. We do that and hit "Create Text".

The screenshot shows a user interface for a tool called "Simple C-LARA". At the top, there are two green status bars: the first says "Project created" and the second says "Operation completed successfully". Below these, the title "Simple C-LARA" is displayed in large, bold, black font. Underneath the title, the section "Project name" is shown, with the input field containing "Cat lover" and a "Change Name" button next to it. The main content area is titled "Prompt" and contains a text box with the following story: "Write an amusing, slightly risqué story about a bold tabby cat who falls in love with a human woman and decides to win her heart. It seems impossible at first, but in the end the cat's charm proves irresistible. We do not want to give explicit details or indeed explain how this is possible, but it should be suggested that their passion is quite physical, e.g. they end up sharing a bed." Below the prompt text is a "Create Text" button and a link "[View project in full C-LARA]".

C-LARA now adds two new boxes, showing the title and text it has created based on our prompt. When we look at the text, we see that it is, as requested, a cute romantic story about a cat who seduces a beautiful human woman.

We could edit the story, but it looks fine as it is. We continue by hitting “Create Segmented Text”.

quite physical, e.g. they end up sharing a bed.

## Generated text title

Félix et Juliette : Une Idylle Inattendue en Provence

[Save Edited Text Title](#)

## Generated text

Dans un petit village pittoresque de Provence, vivait un chat tigré nommé Félix. Félix n'était pas un chat comme les autres. D'une élégance folle, il se pavannait dans les ruelles, la queue haute et le regard envoûtant. Ses exploits pour séduire les chattes du quartier étaient légendaires. Mais un jour, son cœur battit pour une créature tout à fait différente : une femme nommée Juliette, la fleuriste

[Save Edited Text](#) [Rewrite Text](#)

## Multimedia text

[Create Segmented Text](#)

[\[View project in full C-LARA\]](#)

The AI now shows a new box, where the text has been marked up with segmentation information. Page breaks are shown with the <page> tag, and segment breaks with the || tag. Words containing an apostrophe are enclosed in @ . . . @ brackets, and words that should be divided into smaller are split using the | tag.

## Segmented title

Félix et Juliette : Une Idylle Inattendue en Provence

**Save Edited Segmented Title**

## Segmented text

<page>Dans un petit village pittoresque de Provence, vivait un chat tigré nommé Félix.|| Félix @n'@l'était pas un chat comme les autres.|| @D'@l'une élégance folle, il se pavannait dans les ruelles, la queue haute et le regard envoûtant.|| Ses exploits pour séduire les chattes du quartier étaient légendaires.|| Mais un jour, son cœur battit pour une créature tout à fait différente : une femme nommée Juliette, la fleuriste du village.||

<page>Juliette était belle, avec ses cheveux dorés et son sourire éclatant.|| Chaque matin, Félix @l'@l'observait depuis le muret, enroulé comme une élégante écharpe tigrée.|| Après quelques semaines, Félix décida @qu'@il ne pouvait plus résister à son amour

**Save Edited Segmented Text**

This time, we decide to make a few changes. We add some page breaks so that we can include more pictures (we are getting one picture per page). Also we notice that the AI has for some reason split up the reflexive pronoun *se* as *s | e*. We fix this and save.

## Segmented text

<page>Dans un petit village pittoresque de Provence, vivait un chat tigré nommé Félix.|| Félix @n'@l'était pas un chat comme les autres.|| @D'@l'une élégance folle, il se pavannait dans les ruelles, la queue haute et le regard envoûtant.||

<page>Ses exploits pour séduire les chattes du quartier étaient légendaires.||

<page>Mais un jour, son cœur battit pour une créature tout à fait différente : une femme nommée Juliette, la fleuriste du village.||

<page>Juliette était belle, avec ses cheveux dorés et son sourire éclatant.|| Chaque matin, Félix @l'@l'observait depuis le murat encoulé comme une

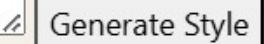
Save Edited Segmented Text

We are now ready to start on the images! Our first step is to define a suitable style. Usually this works best if we give fairly general advice and ask the AI to sort out the details. We fill in a short sentence and hit “Generate Style”.

## Image Style

Briefly describe the style you want to use for the images.

Use a sexy but tasteful adult cartoon style  
inspired by French B.D.

 Generate Style

C-LARA gives us a sample image created with its new style. It looks fine, so we continue.

## Image Style



Briefly describe the style you want to use for the images.

Use a sexy but tasteful adult cartoon style inspired by French B.D.

[Regenerate Style](#)

We're now going to create descriptions of visual elements that occur multiple times in the text, most obviously the cat and the woman. We hit "Generate Image Elements".

## Image Elements

[Generate Image Elements](#)

C-LARA works for a bit and then shows us this display. We have a list of elements, with a thumbnail giving the approximate appearance and a control to let us check more carefully and possibly edit. We're curious to know what the woman looks like, so we click on that line.

Image Elements		
Element Name	Image	Actions
Félix		<a href="#">Review Element</a> <a href="#">Delete Element</a>
Juliette		<a href="#">Review Element</a> <a href="#">Delete Element</a>
the village		<a href="#">Review Element</a> <a href="#">Delete Element</a>

The woman is indeed quite sexy, but (we think, at least) in a tasteful way. So far, the AI seems to be following our instructions well!

### Edit Image Element (Cat lover, Element "Juliette")



**Note:** This is the exact text that was sent to DALL-E-3 to generate the images above.

Juliette is a captivating character inspired by the enchanting allure of French B.D. (bande dessinée) style. She is a woman in her mid-30s, radiating an elegant charm typical of the Provençal setting. Juliette is of Caucasian descent, with warm, golden blonde hair flowing in soft waves down to her shoulders. Her hair reflects the sunlit tones of a Provençal summer. Her eyes are a striking shade of green, reminiscent of lush vineyards, and they exude warmth and kindness. Standing at an average height with a slender, graceful build, Juliette embodies a natural elegance.

Her clothing reflects a timeless, sophisticated style. She wears a flowing, pastel-colored summer dress

Upvotes: 0 Downvotes: 1  

The other elements also seem fine, so we hit “Generate Page Images” to make the actual images that will go in the text.

## Page Images

[Generate Page Images](#)

C-LARA again works for a while and then shows us this display. It's similar to the one for the elements, but now each line is a page.

Page Images		Image	Links
Page Number	Text		
1	<h1>Félix et Juliette : Une Idylle Inattendue en Provence</h1>		<a href="#">Review this Page</a>
2	Dans un petit village pittoresque de Provence, vivait un chat tigré nommé Félix. Félix n'était pas u...		<a href="#">Review this Page</a>
3	Ses exploits pour séduire les chattes du quartier étaient légendaires.		<a href="#">Review this Page</a>
4	Mais un jour, son cœur battit pour une créature tout à fait différente : une femme nommée Juliette, ...		<a href="#">Review this Page</a>

Checking the images quickly, most of them look okay. But we notice that the style in the one where the cat turns up in the woman's bedroom seems a bit off: the woman looks more photorealistic than cartoon. We click to get more details and see three alternatives, where the first one is currently selected.

### Edit Image for Page (Cat lover, Page 11)

#### Text used to create images

Chaque nuit par la suite, leurs paupières se fermaient côté à côté, bercées par une complicité silencieuse.



Upvotes: 0 Downvotes: 1  



Upvotes: 0 Downvotes: 1  



Upvotes: 0 Downvotes: 1  

**Note:** This is the exact text that was sent to DALL-E-3 to generate the images above.

Provence. The atmosphere is further enriched by rustic elements typical of a Provençal setting, such as weathered wooden furniture, a vase with delicate lavender flowers, and soft textured textiles. This setting, combined with subtle hints of lush flora designs on the walls, captures the quaint village charm.

The overall composition should emphasize the quiet complicity and growing bond between Félix and Juliette, capturing a magical, unexpected love story. The illustration should evoke a sense of warmth, comfort, and romantic whimsy, balancing sophistication and charm, and drawing the viewer into the

The style in the third one looks more cartoon-like, so we click to give it a thumbs-up and select it, then go back to the main Simple C-LARA page. We do the same thing on a couple more images which don't seem quite correct.

**Edit Image for Page (Cat lover, Page 11)**

**Text used to create images**

Chaque nuit par la suite, leurs paupières se fermaient côté à côté, bercées par une complicité silencieuse.

Upvotes: 0 Downvotes: 1

Upvotes: 0 Downvotes: 1

Upvotes: 1 Downvotes: 1

We're more or less satisfied with the images (maybe we will come back later and do another pass), but right now we're curious progress further. We scroll down to the section "Multimedia text" and hit "Generate Multimedia Text".

**Multimedia text**

C-LARA works for a bit, adding translations, glosses and other information, then puts it all together and presents a link where we can examine the result.

**Multimedia text**

[View Multimedia Text](#)

We click on "View Multimedia Text", and it opens in another tab. It could still do with some more polishing, but it's not bad for a first draft!

◀ ◀ ▶ ▶ Frequency Alphabetical Sounds

▶ 0.00 / 0.11 ⏸ :

Dans un petit village pittoresque de Provence, vivait un chat tigré nommé Félix.  Félix n'était pas un chat comme les autres.   tabby élégance folle, il se pavannait dans les ruelles, la queue haute et le regard envoûtant.  



We decide to post it so that other people can see it too and maybe make suggestions. We click on the new control “Post to Social Network”.

## C-LARA Social Network

Post to Social Network

We now have a link to the newly generated social network page. We click it to take a look.

## C-LARA Social Network

[View Social Network Page](#)

It looks fine! C-LARA has added a plausible summary.

## Cat lover

Link to published content: [/accounts/rendered\\_texts/540/normal/page\\_1.html](/accounts/rendered_texts/540/normal/page_1.html)

Access count: 2

No ratings yet



[Submit rating](#)

Text language:	French
Annotation language:	English
Created by:	<a href="#">mannyrayner</a>
Summary:	In a picturesque village in Provence, a charming tabby cat named Félix fell in love with Juliette, the village florist, and after winning her affection with thoughtful gifts and companionship, he moved into her home, creating a unique and magical bond between them. Despite the gossip from villagers, Félix and Juliette cherished their extraordinary love, proving that unexpected magic can exist in everyday life.
Length in words:	387
Author:	mannyrayner

When we go to the C-LARA home page (“home” icon on far left), we see that our text is listed first under “Recently published content”.

# C-LARA updates

Showing updates for: Last three days ▾

Update View

## Recently published content

- [Cat lover](#) (French)
- [Una Passeggiata in Campagna con la Famiglia e l'IA](#) (Italian)

For other search options, go to [Social network > Posted content](#)

## Recent project activities

- No recent activities found.

For more detailed activity tracking, go to [Social network > Activities](#)

## C. Creating a picture book with Advanced C-LARA

In this section, we will show how to create a picture book text with one image per page. The text used will be Shakespeare's Sonnet 18 ("Shall I compare thee to a summer's day"). We will divide it up into eight pages, a title page followed by one page for each pair of lines in the fourteen-line poem. We will include glosses and translations in French. The final multimedia text is posted here.

We start by selecting "Create new C-LARA project using Advanced C-LARA" from the "My projects" menu:



We get this screen. We fill in the project name, the text language and the annotation language, and tick the box "Use coherent AI-generated image set".

The screenshot shows a 'Create a new project using Advanced C-LARA' form. The 'Title' field contains 'Shakespeare's Sonnet 18'. The 'Text language' dropdown is set to 'English'. The 'Annotation language' dropdown is set to 'French'. The 'Use coherent AI-generated image set' checkbox is checked. There are two additional checkboxes: 'Use coherent AI-generated image set (legacy version - do not use for new projects)' and 'Base image set on translations', both of which are unchecked. At the bottom of the form is a 'Create Project' button.

We now get the top-level project menu. Most of the options are greyed out, since there is no content yet, but we can select "Create/Edit plain text" to get started.

# Shakespeare's Sonnet 18

Title:

Update Title

Use coherent AI-generated image set:

Use coherent AI-generated image set (legacy version - do not use for new projects):

Base image set on translations:

Update Coherent Image Settings

Text Language: English

Annotation Language: French

Total API cost for this project: \$0.00

[Community](#) ✓

[Create/Edit Plain Text](#)

We now get this screen. We choose “Manually enter/edit text” at the top, paste Shakespeare’s poem into the Text box, and hit Save.

## Create Plain Text ("Shakespeare's Sonnet 18")

Text choice:

- Generate text using AI
- Improve existing text using AI
- Manually enter/edit text
- Load archived version

Archived version:

Prompt:

Current version:

```
Shall I compare thee to a summer's day?  
Thou art more lovely and more temperate:  
Rough winds do shake the darling buds of May,  
And summer's lease hath all too short a date;  
Sometime too hot the eve of heaven shines
```

[Back to Project](#)

We then hit “Back to Project” at the bottom and find that we have more options available in the project menu:

Text Language: English  
Annotation Language: French  
Total API cost for this project: \$0.00

[Community](#)✓

[Create/Edit Plain Text](#) ✓

[Create/Edit Text Title](#)

[Create/Edit Summary](#)

[Estimate/Edit CEFR Level](#)

[Create/Edit Segmented Text](#)

[Create/Edit Segmented Title](#)

[Create/Edit Translated Text](#)

We can now select “Create/Edit Text Title”. Note the difference between the text title and the project name. The project name is an identifier you use to find your project on the platform. The text title, which may not even be in the same language, is what will appear on the front page of the completed text; here, however, it’s convenient for them to be the same. We select “Manually enter/edit title”, fill in “Shakespeare’s Sonnet 18”, and hit “Save”.

## Create Text Title ("Shakespeare's Sonnet 18")

Text choice:

- Generate title using AI
- Improve existing title using AI
- Manually enter/edit title
- Load archived version

Archived version:

Current version:

Shakespeare's Sonnet 18

[Back to Project](#)

We next go back to the project menu and select “Estimate/Edit CEFR Level”. This takes us to the following screen. After selecting “Estimate CEFR level using AI” and hitting “Create”, we see the AI has assigned the text to level C2, the highest one.

Created cefr\_level text

## Get CEFR level ("Shakespeare's Sonnet 18")

Text choice:

- Estimate CEFR level using AI
- Manually enter/edit CEFR level
- Load archived version

Archived version: ai\_generated February 25 2025 04:51:21 ▾

Current version: ai\_generated February 25 2025 04:51:21

C2

[Back to Project](#)

In the same way, we go back to the project menu, select “Create Summary”, and choose the option where the AI is asked to perform the task. We get this sensible result:

Created summary text

## Create Summary ("Shakespeare's Sonnet 18")

Text choice:

Generate summary using AI  
 Improve existing summary using AI  
 Manually enter/edit text  
 Load archived version

Archived version: ai\_generated February 25 2025 04:49:36 ▾

Current version: ai\_generated February 25 2025 04:49:36

The speaker compares their beloved to a summer's day, declaring that their beauty and essence are more enduring and will live on forever through the poem, as long as people exist to read it.

[Back to Project](#)

We now choose “Create Segmented Text” from the project menu. We choose “Segment text using AI” and “Poem or song”, and hit “Create”.

## Create Segmented Text ("Shakespeare's Sonnet 18")

Text choice:

- Segment text using AI
- Try to fix errors in malformed segmented text using AI
- Manually enter/edit segmented text
- Load archived version

Text type:

- Not specified
- Story, essay or similar
- Poem or song
- Dictionary, alphabet book or similar

Archived version:

Current version:

Shall I compare thee to a summer's day?  
Thou art more lovely and more temperate:

[Back to Project](#)

This produces the following result. The AI has done a good job of dividing the text into pages and segments, but, confused by the archaic language, it's not sure about the apostrophes. It needs to enclose a word containing apostrophes in @ ... @ markers if it's a single word, and split it up using @ markers if it's multiple words. It's made the right decision with "untrimm'd" and "wander'st", but missed "grow'st" and "ow'st", and it's got "dimm'd" completely wrong; it seems to think the terminal 'd is like the one in "I'd".

```
<page>Shall I compare thee to a summer's day?||  
Thou art more lovely and more temperate:||  
<page>Rough winds do shake the darling buds of May,||  
And summer's lease hath all too short a date;||  
<page>Sometime too hot the eye of heaven shines,||  
And often is his gold complexion dimm@'d@;||  
<page>And every fair from fair sometime declines,||  
By chance or nature's changing course @untrimm'd@;||  
<page>But thy eternal summer shall not fade,||  
Nor lose possession of that fair thou ow'st;||  
<page>Nor shall death brag thou @wander'st@ in his shade,||  
When in eternal lines to time thou grow'st:||  
<page>So long as men can breathe or eyes can see,||  
So long lives this, and this gives life to thee.||
```

Text:

We manually correct as follows and save.

```

<page>Shall I compare thee to a summer's day?||
Thou art more lovely and more temperate:||
<page>Rough winds do shake the darling buds of May,||
And summer's lease hath all too short a date;||
<page>Sometime too hot the eye of heaven shines,||
And often is his gold complexion @dimm'd@;||
<page>And every fair from fair sometime declines,||
By chance or nature's changing course @untrimm'd@;||
<page>But thy eternal summer shall not fade,||
Nor lose possession of that fair thou @ow'st@;||
<page>Nor shall death brag thou @wander'st@ in his shade,||
When in eternal lines to time thou @grow'st@:||
<page> So long as men can breathe or eyes can see,||
So long lives this, and this gives life to thee.||

```

Text:

Next, we do the same thing with the screen Create Segmented Text Title. Here, the AI doesn't have any tricky sixteenth century English, and it has no trouble splitting up "Shakespeare's".

Created segmented\_title text

## Create Segmented Text Title ("Sonnet 18")

Text choice:

- Generate title using AI
- Manually enter/edit title
- Load archived version

Archived version:

Current version:

Shakespeare@'s@ Sonnet 18

Next, we go to "Create/Edit Translated Text", select "Use the AI to translate text from segmented text", and hit "Create".

## Create Translated Text ("Shakespeare's Sonnet 18")

Text choice:

- Use AI to translate text from segmented text
- Try to fix errors in malformed annotated text using AI
- Add placeholder annotations, creating text if necessary
- Manually edit/enter translations
- Load archived version

Archived version: ▼

Current version:

```
<h1>||Shakespeare@'s@ Sonnet 18||</h1>
<page>Shall I compare thee to a summer's day?||
Thou art more lovely and more temperate:||
<page>Rough winds do shake the darling buds of May,||
And summer's lease hath all too short a date;||
<page>Sometime too hot the eye of heaven shines,||
And often is his gold complexion @dimm'd@;||
<page>And every fair from fair sometime declines ||
```

This gives the following result, where each segment is followed by a French translation.

```
<page><h1>##||Shakespeare@'s@ Sonnet 18#Sonnet 18 de Shakespeare#||</h1>
##<page>Shall I compare thee to a summer's day?#Puis-je te comparer à un jour d'été ?#||
Thou art more lovely and more temperate:#Tu es plus belle et plus tempérée .#||
##<page>Rough winds do shake the darling buds of May,#Les vents violents secouent les tendres
bourgeons de mai,#||
And summer's lease hath all too short a date;#Et le bail de l'été a une date trop courte ;#||
##<page>Sometime too hot the eye of heaven shines,#Parfois l'œil du ciel brille trop chaudement,#||
And often is his gold complexion @dimm'd@;#Et souvent son teint doré est terni ;#||
##<page>And every fair from fair sometime declines,#Et chaque beauté de beauté décline parfois,#||
By chance or nature's changing course @untrimm'd@;#Par hasard ou par le cours changeant de la
nature, non élagué ;#||
##<page>But thy eternal summer shall not fade,#Mais ton été éternel ne se fanera pas,#||
Nor lose possession of that fair thou @ow'st@;#Ni ne perdra la possession de cette beauté que tu
```

In the same way, we progress to “Create/Edit MWE-tagged Text”, select “Generate annotated text from segmented text using AI”, and hit “Create”.

## Create MWE Tagged Text ("Shakespeare's Sonnet 18")

Text choice:

- Generate annotated text from segmented text using AI
- Produce version without CoT analysis
- Try to fix errors in malformed annotated text using AI
- Add placeholder annotations, creating text if necessary
- Manually edit annotated text
- Load archived version

Archived version:

Current version:

```
<h1>||Shakespeare|@'s@ Sonnet 18||</h1>
<page>Shall I compare thee to a summer's day?||
Thou art more lovely and more temperate;||
<page>Rough winds do shake the darling buds of May,||
And summer's lease hath all too short a date;||
<page>Sometime too hot the eye of heaven shines,||
And often is his gold complexion dimm'd;||
```

[Back to Project](#)

The output shows each segment followed by an explanation of the AI's Chain of Thought reasoning as it tries to find multi-word expressions it may contain, and a list of the MWEs in question.

```
<page><h1>||Shakespeare|@'s@ Sonnet 18
_analysis: To analyze the text `["Shakespeare", "s", "Sonnet", "18"]` for Multi-Word Expressions, we need to consider the context and meaning of possible word combinations.
```

1. \*\*"Shakespeare's"\*\*: This is a possessive form of "Shakespeare" and is considered as a single grammatical unit, functioning as a determiner/pronominal adjective. However, as it is not a set expression used in multiple contexts, it is not classified as an MWE here.
2. \*\*"Sonnet 18"\*\*: This is a specific and well-known literary title, referring to the 18th sonnet written by William Shakespeare. The combination of "Sonnet" and a number uniquely identifies a particular poem in the context of Shakespeare's sonnets. Titles like this function as fixed expressions and proper nouns, making "Sonnet 18" a Multi-Word Expression.

Given the analysis, the text contains one MWE:

Text:

[Back to Project](#)

Before editing this, it's convenient to create a version where we remove the explanations. We do this by choosing "Produce version without CoT analysis" and hitting "Submit". We get this output:

```

<page><h1>||Shakespeare|@'s@ Sonnet 18#Sonnet 18#||</h1>
<page>Shall I compare thee to a summer's day?#summer ' s day#||
Thou art more lovely and more temperate:||
<page>Rough winds do shake the darling buds of May,||
And summer's lease hath all too short a date;#summer ' s lease,all too#||
<page>Sometime too hot the eye of heaven shines.||
And often is his gold complexion @dimm'd@;||
<page>And every fair from fair sometime declines,||
By chance or nature's changing course @untrimm'd@;#By chance#||
<page>But thy eternal summer shall not fade,||
Nor lose possession of that fair thou @ow'st@,#thou ow'st#||
<page>Nor shall death brag thou @wander'st@ in his shade,||
When in eternal lines to time thou @grow'st@:||
<page> So long as men can breathe or eyes can see,#So long as#||
So long lives this, and this gives life to thee.#gives life to#||

Text:

```

It looks like the AI has made the wrong call on the phrase “By chance” in the line “By chance and nature’s changing cause untrimm’d”. Usually this is indeed an MWE, but here it is compositional: “chance and nature’s changing course” are doing the untrimming. We edit this out and save.

```

<page><h1>||Shakespeare|@'s@ Sonnet 18#Sonnet 18#||</h1>
<page>Shall I compare thee to a summer's day?#summer ' s day#||
Thou art more lovely and more temperate:||
<page>Rough winds do shake the darling buds of May,||
And summer's lease hath all too short a date;#summer ' s lease,all too#||
<page>Sometime too hot the eye of heaven shines.||
And often is his gold complexion @dimm'd@;||
<page>And every fair from fair sometime declines,||
By chance or nature's changing course @untrimm'd@;||
<page>But thy eternal summer shall not fade,||
Nor lose possession of that fair thou @ow'st@,#thou ow'st#||
<page>Nor shall death brag thou @wander'st@ in his shade,||
When in eternal lines to time thou @grow'st@:||
<page> So long as men can breathe or eyes can see,#So long as#||
So long lives this, and this gives life to thee.#gives life to#||

Text:

```

We now move on to the screen “Edit Images and Pages”, where we will create the images. We decide to leave the image generation parameters as they are. In particular, note that we don’t ask the AI to check the images; we will do this ourselves. The AI’s ability to check images is unfortunately not very good yet, and automatic checking only seems useful if we have a lot of simple images,

## Edit Images and Pages ("Shakespeare's Sonnet 18")

### Overview Document

[Generate Overview Document](#)

### Download All Images

[Download All Images as Zip](#)

### Image Generation Parameters

Number of descriptions to create per item:

Number of images to create per description:

Number of previous pages to use as context when creating a page image:

Maximum number of rounds of generation to try when creating a page image:

Use AI to check images: [Only review images manually](#)

Language to use for writing prompts: [English](#)

Method to use for interpreting page images: [With context v3 objective](#)

[Back to Project](#)

Moving down to the section “Background information” in the same screen, we fill in the following text. The AI will consult it when creating all the images.

### Background information

The poet is addressing the loved one. Images may show either the things described in the text or the poet, as seems most appropriate.

[Save background information](#)

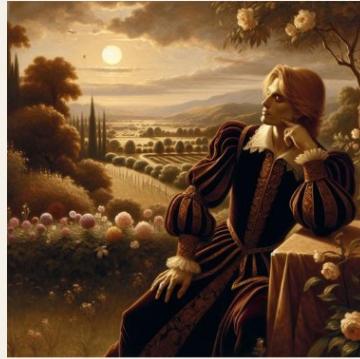
We now move further down to the section “Style”. As the name suggests, we are defining a style that will be used consistently for all the images. Experience suggests that we usually get best results by providing vague instructions where the AI is given free hands to decide all the specifics, so we do that and hit “Generate Style Image”.

**Style**

Advice	Image	Actions
Choose a style that is appropriately respectful to this immortal metaphysical love poem and reflects the visual art of the late 16th century, when it was written.	No style image currently stored.	
<a href="#">Save Style Advice</a> <a href="#">Generate Style Image</a>		

We get this result. An expert on art history might quibble about whether the style was authentically late 16th century, but as a non-expert I am happy with it. If I do want to make adjustments, or am curious about the details, I can click on the link “Edit image style”.

**Style**

Advice	Image	Actions
Choose a style that is appropriately respectful to this immortal metaphysical love poem and reflects the visual art of the late 16th century, when it was written.		<a href="#">Edit Image Style</a>
<a href="#">Save Style Advice</a>		

Doing so takes us to this screen, where we get the full text of the image description. We have the option to create modified versions of the style, but we’ll pass, go back to the main Edit Images and Pages screen, and scroll down to the “Elements” section.

## Edit Image Style (Shakespeare's Sonnet 18)



Note: This is the exact text that was sent to DALL-E-3 to generate the images above.

should be composed of warm, earthy tones such as golds, deep greens, and soft browns, accented with gentle sky blues and lush verdant hues to evoke the natural beauty described in the sonnet.

Line work should be delicate yet expressive, with fine detail, capturing the intricate fashion and settings of the time, such as lavish garments and blooming landscapes. Maintain a romantic and contemplative mood, emphasizing both the ephemeral beauty of nature and the enduring quality of art and love. The atmosphere should be serene and poetic, reflecting the metaphysical ponderings of the text. Illustrate scenes with soft lighting and gentle shadows to enhance the timeless and ethereal quality of the poem.

Upvotes: 0 Downvotes: 1



Here, we're going to create descriptions for the two main images in the poem, the poet who has written it and the loved one he is addressing it to.

We could hit “Generate Missing Element Names” and let the AI try to take care of it, but so far this option seems rather unreliable. So we'll do it ourselves, and start by asking for a description of the poet. We fill that in and hit “Add Element”.

### Elements

Element Name	Advice	Image	Actions
New Element: the poet		Add Element	
Generate Missing Element Names			

This gives us the following indicative image.

Elements			
Element Name	Advice	Image	Actions
the poet			<a href="#">Edit Image Element</a> <a href="#">Delete Element</a>
New Element: <input type="text" value="Enter the name of the new element"/> <a href="#">Add Element</a>			

As with the style, we can get more details or modify the description by clicking on “Edit Image Element”. We decide we’re happy with it, go back, and in the same way create an element description for the loved one.

### Edit Image Element (Shakespeare's Sonnet 18, Element "the\_poet")



**Note:** This is the exact text that was sent to DALL-E-3 to generate the images above.

He wears typical late 16th-century attire: a richly embroidered doublet of deep green velvet, elegantly tailored with ornate gold buttons and elaborate designs to reflect his status and artistic sensitivity. His shirt is of fine white linen, with lace trim peeking from the collar and cuffs, adding a touch of refined gentility. Over his shoulders, he drapes a luxurious cape made of dark, soft fabric, lined with a subtle gold-thread pattern. His hose are of a deep amber hue, tucked into polished leather shoes with a gentle sheen.

Upvotes: 0 Downvotes: 1  

[Back to Edit Images and Pages](#)

To my surprise, I find that the clever AI has apparently decided to make the loved one non-binary! I’m guessing it knows more than I do about the debate concerning homosexual themes in the Sonnets. I was curious enough about this that I made a copy of the project and repeated the experiment; this time, I got a female version of the loved one. But here, let’s continue with the AI’s first interpretation.

## Edit Image Element (Shakespeare's Sonnet 18, Element "the\_loved\_one")



Note: This is the exact text that was sent to DALL-E-3 to generate the images above.

To create an illustration of "the loved one" in Shakespeare's Sonnet 18, envision a character that embodies the elegance and romanticism of the late 16th-century Elizabethan era. The loved one is a young adult, appearing around 20-25 years old, with an androgynous beauty that captivates all who behold them. This character is of Caucasian descent, fitting the historical context, with a fair and delicate complexion that is radiant and timeless.

The loved one's hair is a rich chestnut brown, arranged in soft, natural waves that complement their refined and graceful appearance. Their eyes are a deep, expressive hazel, full of warmth and

Upvotes: 0 Downvotes: 1

[Back to Edit Images and Pages](#)

We go back to "Edit Images and Page", scroll down further to the Pages section, and hit "Generate Missing Page Images". The AI will use the background, style and element information we've already created, combining it with the text on each page to try to create an appropriate image.

## Pages

[Generate Missing Page Images](#)

Display Segmented Text  Display Translated Text  Display MWE Text  Display Lemma Text  Display Gloss Text

It does all the pages in parallel, so we get a result after only a couple of minutes. Here's the generated image for page 8, illustrating the final couplet. The poet and his loved one look out at us gravely. As before, we could try to improve the images, but let's say we're happy and continue.

Page Number	Advice	Image	Links
8			<a href="#">Edit Image for Page</a>
<a href="#">Save All Page Texts</a>			
Plain text	So long as men can breathe or eyes can see, So long lives this, and this gives life to thee.		
Translated text	So long as men can breathe or eyes can see,#Tant que les hommes peuvent respirer ou les yeux peuvent voir,#   So long lives this, and this gives life to thee.#Aussi longtemps que cela vivra, cela te donnera la vie.#   ##		

The next screen is “Create/Edit Lemma Tagged Text”. Similarly to Translated and MWE, we select “Generate annotated text from segmented text using AI” and hit “Create”.

### Create Lemma Tagged Text ("Shakespeare's Sonnet 18")

Text choice:

- Generate annotated text from segmented text using AI
- Try to fix errors in malformed annotated text using AI
- Improve existing annotated text using AI
- Add placeholder annotations, creating text if necessary
- Manually enter annotated text
- Load archived version

We get this result. Each word is labelled with a lemma (root form) and a part of speech tag. We could check this for errors and edit, but let’s move on.

```

<page><h1>||Shakespeare#Shakespeare/PROPN#@'s@'#s/PART# Sonnet#Sonnet 18/NOUN#
18#Sonnet 18/NOUN#||</h1>
<page>Shall#shall/AUX# I#I/PRON# compare#compare/VERB# thee#thee/PRON# to#to/ADP#
a#a/DET# summer#summer ' s day/NOUN# s#summer ' s day/NOUN# day#summer ' s day/NOUN#?|||
Thou#thou/PRON# art#be/AUX# more#more/ADV# lovely#lovely/ADJ# and#and/CCONJ#
more#more/ADV# temperate#temperate/ADJ#:|||
<page>Rough#rough/ADJ# winds#wind/NOUN# do#do/AUX# shake#shake/VERB# the#the/DET#
darling#darling/ADJ# buds#bud/NOUN# of#of/ADP# May#May/PROPN#,|||
And#and/CCONJ# summer#summer ' s lease/NOUN# s#summer ' s lease/NOUN# lease#summer ' s
lease/NOUN# hath#have/VERB# all#all too/ADV# too#all too/ADV# short#short/ADJ# a#a/DET#
date#date/NOUN#:|||
<page>Sometime#sometime/ADV# too#too/ADV# hot#hot/ADJ# the#the/DET# eye#eye/NOUN#
of#of/ADP# heaven#heaven/NOUN# shines#shine/VERB#,|||
And#and/CCONJ# often#often/ADV# is#be/AUX# his#he/PRON# gold#gold/NOUN#
complexion#complexion/NOUN# @dimm'd@#dim/VERB#:|||

```

Text:

## Create Glossed Text ("Shakespeare's Sonnet 18")

Text choice:

- Generate annotated text from segmented text using AI
- Try to fix errors in malformed annotated text using AI
- Improve existing annotated text using AI
- Add placeholder annotations, creating text if necessary
- Manually enter annotated text
- Load archived version
- Delete current glossed text

Here's the result. This time, words are labelled with French glosses. We could review and correct, but at a first glance it seems quite good.

```

<page>Rough#violents# winds#vents# do#-# shake#secouent# the#les# darling#tendres#
buds#bourgeons# of#de# May#mai#,|||
And#et# summer#bail de l'été#'s#bail de l'été# lease#bail de l'été# hath#a# all#trop# too#trop#
short#court# a#une# date#date#,|||
<page>Sometime#parfois# too#trop# hot#chaudement# the#l'# eye#œil# of#de# heaven#ciel#
shines#brille#,|||
And#et# often#souvent# is#est# his#son# gold#doré# complexion#teint# @dimm'd@#terni#,|||
<page>And#Et# every#chaque# fair#beauté# from#-# fair#beauté# sometime#parfois#
declines#décline#,|||
By#par# chance#hasard# or#ou# nature#nature#'s#-# changing#changeant# course#cours#
@untrimm'd@#non élagué#,|||
<page>But#mais# thy#ton# eternal#éternel# summer#été# shall#ne# not#ne# fade#fanera#,|||
Nor#ni# lose#perdra# possession#possession# of#de# that#cette# fair#beauté# thou#que tu
possèdes# @ow'st#@#que tu possèdes#,|||
<page>Nor#ni# shall#-# death#la mort# brag#se vanter# thou#tu# @wander'st@#errerais# in#dans#
Text:

```

We go back to the project menu, select the next screen, “Render Text”, and hit “Render”.

## Render Text ("Shakespeare's Sonnet 18")

[Render](#)

We get this result:

Rendered text created

## Rendered Text ("Shakespeare's Sonnet 18")

[View Content](#)

Register as content:

[Register as Content](#)

Clicking on the link “View content” brings up a tab with the final multimodal version of the text.

"Shakespeare's Sonnet 18" p. 2/8

◀ ▶ ↗ ↘ Frequency Alphabetical

Shall I compare thee to a summer's day?  
Thou art more lovely and more temperate:



If we go back to the “‘Render’ screen, we can mark the checkbox and then hit “Register as Content” to make the multimodal text generally available.

Rendered text created

## Rendered Text ("Shakespeare's Sonnet 18")

[View Content](#)

Register as content:

[Register as Content](#)

We are taken to the newly created page for this piece of content.

## Shakespeare's Sonnet 18

Link to published content: [/accounts/rendered\\_texts/536/normal/page\\_1.html](/accounts/rendered_texts/536/normal/page_1.html)

Access count: 1

No ratings yet



[Submit rating](#)

Text language:	English
Annotation language:	French
Created by:	<a href="#">mannyrayner</a>
Summary:	The speaker compares their beloved to a summer's day, declaring that their beauty and essence are more enduring and will live on forever through the poem, as long as people exist to read it.
Length in words:	121

If we go to the C-LARA home page (“home” icon at top left of screen), we see our text listed under “Recently published content”.

## C-LARA updates

Showing updates for: [Last three days](#) ▾

[Update View](#)

### Recently published content

- [Shakespeare's Sonnet 18](#) (English)
- [Bongu eu me ovethet](#) (Iaai)
- [Boole i Sisiwanyano me Koradrane qatr - version française](#) (Drehu)

For other search options, go to [Social network > Posted content](#)

### Recent project activities

- No recent activities found.

For more detailed activity tracking, go to [Social network > Activities](#)

## D. Creating a text in a language not supported by the AI

This appendix presents a toy example showing how to create a text in a language not supported by the AI; this means that linguistic annotation needs to be performed manually by the user. The language we have chosen is Pitjantjatjara, a Central Australian language that belongs to the Western Desert family, itself a sub-family of Pama-Nyungan. The text consists of the three sentences shown in Table 7, taken from an exercise in Unit 17 of the 1967 Adelaide University advanced course in Pitjantjatjara (Amery, 2012).

Table 7: Toy Pitjantjatjara text.

Annotation	Sentence
<b>Sentence 1</b>	
Translation	The dogs were coming to camp
Plain	Papa tjuta ngurakutu pitjangi
Segmented	Papa tjuta ngura-kutu pitjangi
Gloss	dog many camp-TO come
<b>Sentence 2</b>	
Translation	The woman with many foods was sitting
Plain	Minyma mai tjutatjara nyinangi
Segmented	Minyma mai tjuta-tjara nyinangi
Gloss	woman food many-WITH sitting
<b>Sentence 3</b>	
Translation	The children run from the rocks
Plain	Tjitji tjuta apunguru wirtjapakalpai
Segmented	Tjitji tjuta apu-nguru wirtjapakalpai
Gloss	child many rock-FROM run

We start by going to My Projects > Create new C-LARA project using Advanced C-LARA. We get the following screen and fill in the project name and the text and annotation languages, We will be adding an image for each sentence, so we check “Use coherent AI-generated image set” and “Base image set on translations”. Then we hit Create Project,

**Create a new project using Advanced C-LARA**

Title:	Toy Pitjantjatjara example
Text language:	Pitjantjatjara
Annotation language:	English
Use coherent AI-generated image set:	<input checked="" type="checkbox"/>
Use coherent AI-generated image set (legacy version - do not use for new projects):	<input type="checkbox"/>
Base image set on translations:	<input checked="" type="checkbox"/>
<b>Create Project</b>	

We now get the main project menu screen. We enter Create/Edit Plain Text.

**Toy Pitjantjatjara example**

Title:	<input type="text"/>	<b>Update Title</b>
Use coherent AI-generated image set:	<input checked="" type="checkbox"/>	
Use coherent AI-generated image set (legacy version - do not use for new projects):	<input type="checkbox"/>	
Base image set on translations:	<input checked="" type="checkbox"/>	
Make an image questionnaire available:	<input type="checkbox"/>	
<b>Update Coherent Image Settings</b>		
Text Language: Pitjantjatjara		
Annotation Language: English		
Total API cost for this project: \$0.00		
<a href="#">Community</a> /		
<a href="#">Create/Edit Plain Text</a>		

In the Create Plain Text screen, we check Manually enter/edit text, copy in the three sentences, and hit Save. Then we go back to the project menu and enter Create/Edit Text Title.

## Create Plain Text ("Toy Pitjantjatjara example")

Text choice:

- Generate text using AI
- Improve existing text using AI
- Manually enter/edit text
- Load archived version

Archived version:

Prompt:

Current version:

Papa tjuta ngurakutu pitjangi.

Minyma mai tjutatjara nyinangi.

Tjitji tjuta apungulu wirtjapakalpai.

In the Create Text Title screen, we select Manually enter/edit title and fill in the title. “Pitjantjatjara” will do for now. Then we save, go back to the project menu, and continue to Create/Edit Segmented Text.

## Create Text Title ("Toy Pitjantjatjara example")

Text choice:

- Generate title using AI
- Improve existing title using AI
- Manually enter/edit title
- Load archived version

Archived version:

Current version:

Pitjantjatjara

In the Create Segmented Text screen, we add markers to segment the plain text. We put a <page> at the beginning of every page, and a || at the end of every segment. We use | to split

up the words which include a separable case affix like *kutu*. Then we save, go back to the project menu, and select Create/Edit Segmented Title.

The screenshot shows a user interface for creating segmented text titles. At the top, there are three radio button options: "Try to fix errors in malformed segmented text using AI" (unchecked), "Manually enter/edit segmented text" (checked), and "Load archived version" (unchecked). Below this is a "Text type:" section with four radio button options: "Not specified" (unchecked), "Story, essay or similar" (unchecked), "Poem or song" (unchecked), and "Dictionary, alphabet book or similar" (checked). Further down are dropdown menus for "Archived version:" and "Current version:". A large text area contains three lines of Pitjantjatjara text with red underlines: "<page>Papa tjuta nguralkutu pitjangi.||", "<page>Minyma mai tjultjara nyinangi.||", and "<page>Tjiti tjuta apulnguru wirtjapakalpai.||".

The title consists of a single word, so we don't need to do anything. We save it, return to the project menu, and go on to Edit Images and Pages. This is where we will do most of the work.

The screenshot shows the "Create Segmented Text Title" interface again. The title "Create Segmented Text Title ("Toy Pitjantjatjara example")" is at the top. Below it is a "Text choice:" section with three radio button options: "Generate title using AI" (unchecked), "Manually enter/edit title" (checked), and "Load archived version" (unchecked). There are dropdown menus for "Archived version:" and "Current version:". A large text area contains the word "Pitjantjatjara".

We start with the image generation parameters. The default is to use DALL-E-3 to generate the images, write the image prompts in English, and review the images manually. That all looks fine, so we don't need to do anything here. We scroll down to Background Information.

## Edit Images and Pages ("Toy Pitjantjatjara example")

### Overview Document

[Generate Overview Document](#)

### Download All Images

[Download All Images as Zip](#)

### Image Generation Parameters

Image generation model: [DALL-E-3](#)

Use AI to check images: [Only review images manually](#)

Language to use for writing prompts: [English](#)

Default AI model: [gpt-4o](#)

AI model to use for generating element names: [gpt-4o](#)

AI model to use for generating image prompts: [gpt-4o](#)

AI model to use for checking descriptions: [gpt-4o](#)

It's often useful to give the AI some background about the general nature of the task. Just in case it doesn't know (most likely it does, but it's good to be careful), we remind it that the Pitjantjatjara people and their language should be treated with appropriate respect.

### Background information

The [Pitjantjatjara](#) language is the property of the [Pitjantjatjara](#) people. It is of the utmost importance to show them appropriate respect.

[Save background information](#)

We're now going to fill in the translations. First, we set the toggles just under Pages to display the translated text panes.

### Pages

Display Segmented Text  Display Translated Text  Display MWE Text  Display Lemma Text  Display Gloss Text

We now have the text panes available. We fill in the translation for each segment.

2		No image.
<b>Save All Page Texts</b>		
Plain text	Papa tjuta ngurakutu pitjangi.	
Translated text	Papa tjuta ngurakutu pitjangi.#The dogs were coming to camp.#   ##	

Since we've filled in the translations, the AI knows what the text means, so we can go down to Style. We give the AI some general ideas about the kind of style we'd like but let it decide on all the specifics. Then we hit Generate Style Image.

# Style

## Advice

Create a respectful style reflecting  
Pitjantjatjara art and culture and the  
 backdrop of the Australian Western desert.

**Save Style Advice**   **Generate Style Image**

The AI shows us an example of the style it has in mind. We're happy with it, so we scroll on down.

## Style

Advice	Image	Actions
Create a respectful style reflecting Pitjantjatjara art and culture and the backdrop of the Australian Western desert.		<a href="#">Edit Image Style</a>
<a href="#">Save Style Advice</a>		

If we had people or objects that occurred on several pages, we'd need to do something with Elements. This is normal for a story. But here we have three unconnected sentences, so we move further down to Pages and hit Generate Missing Page Images.

## Elements

Element Name	Advice	Image	Actions
New Element: <input type="text" value="Enter the name of the new element"/> <a href="#">Add Element</a>			
<a href="#">Generate Missing Element Names</a>			

## Pages

<a href="#">Generate Missing Page Images</a>	Number of pages to generate: <input type="text" value="Leave blank for 'all'"/>
<a href="#">Delete All Page Images</a>	

The AI adds an image for each page. Here's the one for page 4. We're okay with the images, so we decide we'll add the glosses and lemma tags.

Page Number	Advice	Image	Actions
4			<a href="#">Edit Image for Page</a> <a href="#">Delete Page Image and Descriptions</a>
<a href="#">Save All Page Texts</a>			
Plain text	Tjiti tjuta apungutu wirtpakalpai.		
Translated text	Tjiti tjuta apungutu wirtpakalpai.#The children run from the rocks.##		

First, we need to go back to the toggles under Pages and set them to display the appropriate panes.

<input type="checkbox"/> Display Segmented Text	<input checked="" type="checkbox"/> Display Translated Text	<input type="checkbox"/> Display MWE Text	<input checked="" type="checkbox"/> Display Lemma Text	<input checked="" type="checkbox"/> Display Gloss Text
---	---	---	--	--

We're now showing the Glossed text and Lemma-tagged text panes. We add the appropriate information for each page and hit Save All Page Texts. Then we return to the project menu and go down to Audio Processing.

<a href="#">Save All Page Texts</a>	
Plain text	Papa tjuta ngurakutu pitjangi.
Translated text	Papa tjuta ngurakutu pitjangi.#The dogs were coming to camp.##
Glossed text	Papa#dog# tjuta#many# ngura#camp#kutu#TO# pitjangi#were coming#.
Lemma-tagged text	Papa#papa/NOUN# tjuta#tjuta/PART# ngura#ngura/NOUN#kutu#kutu#PART# pitjangi#pitjangi/VERB#.

We adjust the settings. We're going to upload recorded audio files for both words and segments, and we're not using a TTS engine.

## Audio Processing ("Toy Pitjantjatjara example")

Method:

Preferred TTS engine:

Preferred TTS voice:

Use human audio for words:

Use human audio for segments:

Use context for segment audio:

Human Voice ID:

We get this display, with an Choose File button for each word and segment.

Audio	Text	Upload File	
No audio	pitjantjatjara	<input type="button" value="Choose file"/>	No file chosen
No audio	papa	<input type="button" value="Choose file"/>	No file chosen
No audio	tjuta	<input type="button" value="Choose file"/>	No file chosen
No audio	ngura	<input type="button" value="Choose file"/>	No file chosen
No audio	kutu	<input type="button" value="Choose file"/>	No file chosen
No audio	pitjangi	<input type="button" value="Choose file"/>	No file chosen
No audio	minyma	<input type="button" value="Choose file"/>	No file chosen
No audio	mai	<input type="button" value="Choose file"/>	No file chosen

We use the Choose File buttons to select the audio files. At the end we hit Submit to upload them.

Audio	Text	Upload File
No audio	pitjantjatjara	<input type="button" value="Choose file"/> pitjantjatjara.mp3
No audio	papa	<input type="button" value="Choose file"/> papa.mp3
No audio	tjuta	<input type="button" value="Choose file"/> tjuta.mp3
No audio	ngura	<input type="button" value="Choose file"/> ngura.mp3
No audio	kutu	<input type="button" value="Choose file"/> kutu.mp3
No audio	pitjangi	<input type="button" value="Choose file"/> pitjangi.mp3
No audio	minyma	<input type="button" value="Choose file"/> minyma.mp3
No audio	mai	<input type="button" value="Choose file"/> mai.mp3
No audio	tjara	<input type="button" value="Choose file"/> tjara.mp3

We see we now have the files in place. We can listen to them if we want to check they are right.

Audio	Text	Upload File
▶ 0:00 / 0:00	pitjantjatjara	<input type="button" value="Choose file"/> No file chosen
▶ 0:01 / 0:01	papa	<input type="button" value="Choose file"/> No file chosen
▶ 0:02 / 0:02	tjuta	<input type="button" value="Choose file"/> No file chosen
▶ 0:01 / 0:01	ngura	<input type="button" value="Choose file"/> No file chosen
▶ 0:00 / 0:00	kutu	<input type="button" value="Choose file"/> No file chosen
▶ 0:00 / 0:02	pitjangi	<input type="button" value="Choose file"/> No file chosen
▶ 0:00 / 0:02		<input type="button" value="Choose file"/> No file chosen

Everything sounds good, so we go back to the project menu and then on to Render text.

## Render Text ("Toy Pitjantjatjara example")

We hit Render, and C-LARA puts everything together for us. There is now a link to view the final result.

Rendered text created

## Rendered Text ("Toy Pitjantjatjara example")

[View Content](#)

Register as content:

It's worked! We can hover over words to see glosses, or click on them to hear audio. The speaker icons play audio for segments, and the pencil icons show segment translations.

A screenshot of a digital interface for language learning. At the top, there are navigation buttons: back, forward, and search. Below them are two tabs: "Frequency" (which is selected) and "Alphabetical". A media control bar shows a play button, a progress bar at 0:00 / 0:13, a volume icon, and a more options icon. The main content area displays a sentence in a language with glosses: "Tjiti tjuța apunguru wirtjapakalpai." The word "apunguru" is highlighted in yellow and has a black box with the word "rock" written in white. To the right of the sentence is a small speaker icon and a pencil icon. Below the sentence is a painting of a landscape with several large, textured rock formations and several small figures running across the foreground.

Everything is good, we're done!

## E. Prompt templates

Finally, we present the prompt templates and few-shot examples referred to earlier in the report:

- Prompt template for English MWE annotation. Figure 17.
- Typical few-shot example for English MWE annotation. Figure 18.
- Prompt template for glossing. Figure 19.
- Prompt template for creating an image style. Figure 20.
- Prompt template for creating the candidate list of image elements. Figure 21.
- Prompt template for creating an image element description. Figure 22.
- Prompt template for selecting elements relevant to a page. Figure 23.
- Prompt template for creating the description of a page image (DALL-E-3 version). Figure 24.
- Prompt template for analysing uploaded element image. Figure 25.

Read through the following short piece of {12\_language} text, represented as a JSON list of words and punctuation marks, and identify all the Multi-Word Expressions (MWEs). An MWE is defined here as any collection of two or more words, not necessarily contiguous, that form a meaningful unit within the context of the text and could be expected to appear as an entry, or be an inflected form of an entry, in a comprehensive lexicon. An MWE functions as a single part of speech, such as verbs, nouns, adjectives, adverbs or interjections, and should have a meaning that is not directly deducible from the meanings of its individual components. Typical English MWEs are phrasal verbs and set expressions of various kinds.

Start by writing out an analysis of the passage, considering plausible candidate phrases and giving justifications for why they are or are not MWEs. At the end, write out a list of the MWEs in JSON form. Each MWE will be a list of elements taken from the input string, presented in the order in which they appear.

Here are some examples:

```
{few_shot_examples}
```

Analyze the following text:

```
{text_in_json_form}
```

Figure 17: Prompt template for English MWE annotation. The calling code formats the few-shot examples before inserting them.

```

{
  "text": "Tom was going to lie, but he ended up spilling the beans.",

  "analysis": """**Analysis:**"
  1. **"going to lie"**
    - * "going to" is part of the future construction expressing intent,
      functioning as an auxiliary verb phrase. Its meaning is not directly
      deducible from the individual words "going" and "to," and it conveys a
      unique, non-compositional meaning. Therefore, it is considered an MWE.

  2. **"ended up"**
    - This is an inflected form of the phrasal verb "end up," which means to
      finally be in a particular state or situation. The meaning of "ended up"
      is not directly deducible from the individual words "ended" and "up."
      It functions as a single unit of meaning within the context of the
      sentence, making it an MWE.

  3. **"spilling the beans"**
    - This phrase is an inflected form of the idiomatic expression "spill the
      beans," which means to reveal a secret or disclose information
      inadvertently. The meaning is not directly deducible from the words
      "spilling," "the," and "beans" taken individually. It functions as a
      single verb phrase with an idiomatic meaning, qualifying it as an MWE.",

  "mwes": "going to,ended up,spilling the beans",
}

```

Figure 18: Typical few-shot example for English MWE annotation.

I am going to give you a short passage in {l2\_language}, presented in a JSON format. I want you to add {l1\_language} glosses in the following way.

The JSON is a list of pairs. The first element in each pair is a string which is either a {l2\_language} word or a punctuation mark. If the second element in the pair is non-null, it specifies a multi-word expression (MWE) and indicates that this element is part of the given MWE.

Your task is to replace the second element in each pair with a gloss in {l1\_language}. Use the following rules:

1. If the element is marked as an MWE, the gloss should be for the given MWE. Each component of an MWE must have the exact same gloss for the MWE, regardless of context. If the MWE appears to be a proper name, and you cannot find a known form of the proper name in {l1\_language}, gloss the proper name as itself or a transliteration of itself.
2. If the second element is null, the gloss should be for the first element, if there is a sensible way to produce such a gloss. If the first element appears to be a proper name, and you cannot find a known form of the proper name in {l1\_language}, gloss the proper name as itself or a transliteration of itself.
3. If the second element is null, and there is no sensible way to gloss the first element in {l1\_language}, the gloss should be specifically '-'.

{few\_shot\_examples}

Here are the items to gloss:

{text\_in\_json\_form}  
MWEs: {mwes}

{context\_text}

Do not add or remove any elements or alter the structure of the JSON list provided in the input.

Write out just the annotated JSON with no introduction, since it will be processed by a Python script.

We are later going to create a set of images to illustrate the following text:

```
{text}
```

The intended style in which the images will be produced is briefly described as follows:

```
{base_description}
```

```
{background_text}
```

For now, please expand the brief style description into a detailed specification that can be used as part of the prompts later passed to DALL-E-3 to create illustrations for this story and enforce a uniform appearance.

The expanded style specification should at a minimum include information about the medium and technique, the colour palette, the line work, and the mood/atmosphere, since these are all critical to maintaining coherence of the images which will use this style.

Write the specification in {text\_language}.

The specification needs to be at most 1000 characters long

Figure 20: Prompt template for creating an image style. `background_text` contains some introductory text if it is non-empty.

We are later going to create a set of images, one per page, to illustrate the following text, which has been divided into numbered pages:

```
{formatted_story_data}
```

```
{background_text}
```

As part of the process, we need to identify visual elements that would occur on more than one page of the text, e.g. characters, objects and locations. In this step, please write out a JSON-formatted list of these elements. For example, if the text were the traditional nursery rhyme

```
[  
  {  
    {  
      "page_number": 1,  
      "text": "Humpty Dumpty sat on a wall"  
    },  
    {  
      {  
        "page_number": 2,  
        "text": "Humpty Dumpty had a great fall"  
      },  
      {  
        "page_number": 3,  
        "text": "All the King's horses and all the King's men"  
      },  
      {  
        "page_number": 3,  
        "text": "Couldn't put Humpty Dumpty together again"  
      }  
  }]
```

then a plausible list might be

```
[ "Humpty Dumpty",  
  "the wall",  
  "the King's horses",  
  "the King's men"]
```

since "Humpty Dumpty" and "the wall" would probably occur in the illustrations for all the pages, and "the King's horses" and "the King's men" for in the illustrations for pages 3 and 4.

Please write out only the JSON-formatted list, since it will be read by a Python script.

Figure 21: Prompt template for creating the candidate list of image elements.  
background\_text contains some introductory text if it is non-empty.

We are going to create a set of images to illustrate the following text:

```
{ {text} }  
  
{ {background_text} }
```

The intended style in which the images will be produced is described as follows:

```
{ {style_description} }
```

As part of this process, we are first creating detailed descriptions of visual elements that occur multiple times in the text, such as characters, objects, and locations.

\*\*Your task is to create a detailed specification of the element "`{ {element_text} }`", in the intended style, to be passed to DALL-E-3 to generate a single image showing how "`{ {element_text} }`" will be realized.\*\*

\*\*Please ensure that the specification includes specific physical characteristics:

- \*\*Apparent age\*\*
- \*\*Gender\*\*
- \*\*Ethnicity\*\*
- \*\*Hair color and style\*\*
- \*\*Eye color\*\*
- \*\*Height and build\*\*
- \*\*Clothing and accessories\*\*
- \*\*Distinctive features or expressions\*\*
- \*\*General demeanour\*\*

\*\*Similarly, for an animal we would require characteristics like:\*\*

- \*\*Size and shape\*\*
- \*\*Colour of fur/scales, markings\*\*
- \*\*Eye color\*\*
- \*\*Distinctive features like wings, tail, horns\*\*
- \*\*General demeanour\*\*

\*\* IMPORTANT \*\*

- Write the description in `{ {text_language} }`.
- Be as precise and detailed as possible to ensure consistency in the generated images.\*\*
- The description should be at most `{max_element_description_length}` characters long, as it will later<sup>107</sup> be combined with other descriptions.
- DALL-E-3 will only receive this text when generating the image, so it is essential to include the style information.
- Take account of the following advice from the user about how to realise the image:

```
{ {advice_text} }
```

We are generating a set of images to illustrate the following text, which has been divided into numbered pages:

```
{formatted_story_data}
```

We are preparing to generate an image for page {page\_number}, whose text is

```
{page_text}
```

We have already generated detailed descriptions for various elements that occur in more than one page. The list of elements is the following:

```
{all_element_texts}
```

When we later generate the image for page {page\_number}, it may be helpful to consult some of these descriptions.

In this step, the task is to write out a JSON-formatted list of the elements relevant to page {page\_number}e.

For example, if the text were the traditional nursery rhyme

```
[ {{"page_number": 1,
    "text": "Humpty Dumpty sat on a wall"}, ,
  {"page_number": 2,
    "text": "Humpty Dumpty had a great fall"}, ,
  {"page_number": 3,
    "text": "All the King's horses and all the King's men"}, ,
  {"page_number": 4,
    "text": "Couldn't put Humpty Dumpty together again"}]
```

and the list of elements was

```
[ "Humpty Dumpty",
  "the wall",
  "the King's horses",
  "the King's men"]
]
```

then a plausible list of relevant elements for page 2 would be

```
[ "Humpty Dumpty", "the wall"]
```

since the image on page 2 will contain Humpty Dumpty and the wall, but probably not the King's horses or the King's men.<sup>108</sup>

Please write out only the JSON-formatted list, since it will be read by a Pyt

Figure 23: Prompt template for selecting elements relevant to a page.

We are generating a set of images to illustrate the following text, which has been divided into numbered pages:

```
{{formatted_story_data}}  
{{background_text}}
```

The intended style in which the images will be produced is described as follows:

```
{{style_description}}
```

We are about to generate the image for page {{page\_number}}, whose text is

```
{{page_text}}
```

We have already generated detailed specifications for various elements (characters, locations, etc) that occur on more than one page. Here are the specifications of relevant previous pages and elements:

```
{{element_descriptions_text}}
```

In this step, please create a detailed specification of the image on page {{page\_number}}, in the intended style and also consistent with the image for page {{page\_number}}.

\*IMPORTANT\*:

- Write the specification in {{text\_language}}.
- The specification you write out must be at most {{max\_page\_description\_length}} characters long to conform with DALL-E-3's constraints.
- Start the specification with a short, self-contained section entitled "Essential aspects" of the image and then list the aspects of the image which are essential to the text and must be represented. This will often include material not mentioned in the text on the current page, which is necessary to maintain context and make sense of the image given other background knowledge.

For example, if the text were the traditional nursery rhyme

```
[ {{ { "page_number": 1, "text":  
        "Humpty Dumpty sat on a wall" } } },  
  {{ { "page_number": 2,  
        "text": "Humpty Dumpty had a great fall" } } },  
  {{ { "page_number": 3,  
        "text": "All the King's horses and all the King's men" } } },  
  {{ { "page_number": 4,  
        "text": "Couldn't put Humpty Dumpty together again" } } }  
]
```

then the "Essential aspects" section for page 2 might read:

"Humpty Dumpty is falling off the wall. Humpty Dumpty is an anthropomorphic egg. He looks surprised and frightened."

We are generating a set of images to illustrate the following text, which has been divided into numbered pages:

```
{formatted_story_data}
```

The intended style in which the images will be produced is described as follows:

```
{style_description}
```

As part of this process, we are creating detailed specifications for various elements (characters, locations, etc) that occur on more than one page.

The user has uploaded an image which they wish to use as inspiration for the element "{element\_text}", suitably adapting it by taking account of the style description. Here is a description of the uploaded image produced by gpt-4o:

```
{image_interpretation}
```

In this step, please create a detailed specification of the image for the element "{element\_text}", based on the user uploaded image, and in the intended style. This will later be used in prompts submitted to DALL-E-3.

**\*IMPORTANT\*:**

Write the specification in {text\_language}.

The specification you write out must be at most 2000 characters long to conform with DALL-E-3's constraints.

Figure 25: Prompt template for analysing uploaded element image.