

A Demonstration of C-LARA

*Branislav Bédi¹, ChatGPT-4², Belinda Chiera³, Cathy Chua⁴,
Neasa Ní Chiaráin⁵, Manny Rayner^{3,6}, Annika Simonsen⁷, Rina Zvi-Girshin⁸*

¹The Árni Magnússon Institute for Icelandic Studies, Iceland; ²OpenAI, US; ³University of South Australia, Australia; ⁴Independent scholar, Australia; ⁵Trinity College, Dublin, Ireland; ⁶University of Geneva, Switzerland; ⁷University of Iceland, Iceland; ⁸Ruppin Academic Center, Netanya, Israel

branislav.bedi@arnastofnun.is, mannyrayner@yahoo.com (suitably labelled mails to this address will be forwarded to the relevant ChatGPT-4 instance),
Belinda.Chiera@unisa.edu.au, cathy@pioneerbooks.com.au, Neasa.NiChiarain@tcd.ie,
Emmanuel.Rayner@unige.ch, ans72@hi.is, rinazg@ruppin.ac.il

Abstract

We outline a demonstration of “C-LARA”, a complete reimplementa-tion of the Learning And Reading Assistant (LARA) which puts ChatGPT-4 in the centre. ChatGPT-4 is used both as a software *component*, to create and annotate text, and as a software *engineer*, to implement the platform itself: about 90% of the new platform’s code has been written by ChatGPT-4, working in close collaboration with one of the authors. We will show how ChatGPT-4 can on its own write and annotate short stories suitable for intermediate level language classes, producing high quality multimedia output generally usable after only minor editing, and how the AI is able to discuss the code, discussing the feasibility of possible enhancements and extensions. **Index Terms:** ChatGPT, C-LARA, CALL, software engineering, open source, multimedia, reading assistance, annotation

1. Background and overview

C-LARA is a complete reimplementa-tion of the Learning And Reading Assistant (LARA; [1]; <https://www.unige.ch/callector/lara/>) with ChatGPT-4 in the centre. As with the original LARA, it provides a web platform where users can create and read multimedia learner texts in many languages. In the original LARA, much of the work needed to produce a multimedia text had to be performed manually. However, in C-LARA, all the steps, including writing the text, can be carried out by ChatGPT-4, though the result will typically contain a few minor errors. If a native speaker of the language in question is available, the errors are usually quick and easy to correct; a simple editing interface is included. A noteworthy feature of C-LARA is that ChatGPT-4 is not just used as a software *component*, to perform runtime processing, but also as a software *engineer*, to construct the system itself. About 90% of the code in C-LARA was written by the AI, working in close collaboration with one of the human authors.

C-LARA is described in detail in the accompanying full paper presented at this conference [2]. Here, we will restrict ourselves to providing examples of the kind of things we would show when demoing the system. In Section 2 we present annotated screenshots exemplifying some of the functionality offered by C-LARA. Section 3 then summarises a conversation between a human and ChatGPT-4, where the AI responds to a suggestion concerning a possible enhancement to the platform.

2. C-LARA functionality

We begin by outlining how C-LARA can be used to write, annotate and post a short multimodal text. The user progresses

through a series of screens, as follows:

Create project. The user enters a title and selects the L2 (text) and L1 (annotation) languages. The system sets up a project page with links to relevant functions.

Create plain text. The user instructs the AI to generate the text, supplying a prompt. The user can then optionally instruct the AI to try to improve the results, post-edit it, or retrieve an earlier version of the file.

Segmentation. The user instructs the AI to add segmentation information, dividing the text into pages and segments and, if necessary, dividing compound words into smaller components. Again, the user can ask for the AI to improve the annotation, post-edit it manually, or retrieve an earlier version.

Glossing. The user instructs the AI to add L1 glosses to words, with the options given in the two previous steps.

Tagging with lemma and POS. The user instructs the AI to add lemma and Part-of-Speech (POS) tags to words, with the same options as before. If a tagger-lemmatiser is available for the language, this can also be used.

Rendering and posting. The user instructs C-LARA to add audio using a TTS engine and combine the resources constructed in the preceding steps into the final multimedia document. C-LARA complies, provides a link to view the result, and asks if the user wishes to register it so that it is visible to other people. If the user accepts, the text is posted on a page which provides a link and also lets users add ratings and comments.

As is apparent, the core of the process is ChatGPT-based annotation: we illustrate with two screenshots. First, Figure 1 gives an example of a screen where a user who is constructing a C-LARA text invokes an annotation functionality, here glossing. Moving to the metalevel, Figure 2 shows the screen a designated language expert can use to customise the ChatGPT prompts used to perform the annotation operations. There are three types of annotation (segmentation, glossing, lemma/POS tagging) and two operations (annotation, improvement of existing annotation); the prompt for each of these six combinations is built from a text template together with a set of few-shot examples and a JSON representation of the text to be annotated, inserted at ChatGPT invocation time. We have found that good choices of templates and examples can substantially improve annotation performance. Other functionalities are also supported. A full list is posted in the project repository¹.

¹<https://sourceforge.net/p/c-lara/svn/HEAD/tree/FUNCTIONALITY.txt>

Create Glossed Text ("Ny katt hjälte i stan")

Text choice:

☐ Annotate text using AI

☐ Improve existing annotated text using AI

☒ Manually enter/edit annotated text

☐ Load archived version

Archived version:

Ny#New# katt#cat#hjälte#hero# i#in# stan#the city#|| Katt#Cat# räddar#saves# sin#its# ägares#owner's# liv#life# från#from# brinnande#burning# hus#house#||

<page>

I#In# den#the# lilla#small# staden#town# Kattköping#Catville#

Text:

Current version label:

Gold standard: ☐

[Back to Project](#)

Figure 1: Glossing screen. The user can choose between telling the AI to gloss the text, telling it to try to improve the current glosses, manually post-editing, or retrieving an earlier version of the file.

3. Discussing C-LARA with ChatGPT-4

As previously noted, ChatGPT-4 is not only used as a runtime component, but also, and arguably more interestingly, as a software engineer. In this section, we present a summary of a short discussion (four turns; ten minutes) between one of the project members and ChatGPT-4, where the project member asks the AI for advice about how to implement a new feature. The conversation was carried out in the main ChatGPT thread that has been used to discuss the project, so the ChatGPT-4 instance was already fully up to date with the relevant issues.

Turn 1 (7 lines): The human suggested that it would be useful to enhance the view which displayed a piece of C-LARA content to include basic social network functionality, so that users could add ratings and comments, and asked for advice on feasibility and implementation.

Turn 2 (1 page): ChatGPT-4 agreed that this would be an interesting and valuable feature, and outlined how the functionality could be added to the Django layer. Following Django's MVC architecture, it divided the work into four parts:

- adding two new models, Rating and Comment, to the Models;
- using these new models in the relevant View to show the necessary rating and comment information;
- adding Forms to allow submission of new ratings and comments; and
- adding Templates to render the above.

The AI also commented that it would be necessary to consider how to handle permissions and authentication.

Turn 3 (8 lines): The human agreed and asked for details on how to show both the user's own rating and the average rating, and how to divide the comment thread into pages equipped

Edit Annotation Prompts

Language:

Annotation type:

Operation:

Template or examples:

String:

String:

String:

String:

String:

String:

Figure 2: Annotation prompt customisation screen. A designated language expert is editing the few-shot examples included in ChatGPT prompts used for the segmentation step, primarily giving illustrations of how compound words should be divided into components using the vertical bar symbol.

with standard navigation controls.

Turn 4 (1.5 pages): The AI answered both points, giving code snippets to show how to

- get the necessary rating information and
- use the Django Paginator functionality to divide the comment thread into navigable pages in the usual way.

By the end of the discussion, the human already had a detailed recipe for adding the desired new features, and was confident that ChatGPT would be able to assist them in converting it into functioning code. It turned out to be easy to follow the plan outlined by the AI, which ended up writing nearly all of the code.

4. Summary and further directions

We have outlined the demo we plan to present at the SLaTE 2023 meeting, illustrating the novel capabilities of the ChatGPT-4-based C-LARA platform. C-LARA is under active development, and we expect more functionality to be added in the near future.

5. References

- [1] E. Akhlaghi, B. Bédi, M. Butterweck, C. Chua, J. Gerlach, H. Habibi, J. Ikeda, M. Rayner, S. Sestigiani, and G. Zuckermann, "Overview of LARA: A learning and reading assistant," in *Proc. SLaTE 2019: 8th ISCA Workshop on Speech and Language Technology in Education*, 2019, pp. 99–103.
- [2] B. Bédi, ChatGPT-4, B. Chiera, C. Chua, N. Ní Chiaráin, M. Rayner, A. Simonsen, and R. Zviels-Girshin, "ChatGPT + LARA = C-LARA," in *Proc. SLaTE 2023: 9th ISCA Workshop on Speech and Language Technology in Education*, 2023.