LAWS: Language Annotation Web System Using ELAN Format

Jeremias Kalebe Estumano
Universidade Federal do Pará
Tucuruí, PA-Brasil
jeremiaskalebe@gmail.com

Antônia Fernanda de Souza Nogueira
Universidade Federal do Pará
Belém, PA-Brasil
afernanda@ufpa.br

Marcos Amaris
Universidade Federal do Pará
Belém, PA-Brasil
amaris@ufpa.br

Abstract

Context: Linguists dedicate extensive time to annotating audiovisual media, where the efficiency of annotation tools significantly impacts their productivity. Existing tools, such as ELAN, are well-known and widely used but present challenges due to their complex interfaces, especially for novice users.

Solution: To overcome these challenges, we propose LAWS, a web-based linguistic annotation tool that emphasizes simplicity, intuitiveness, and a user-friendly interface to improve accessibility and usability for linguists.

Methodology: The development of LAWS focused on streamlining access, handling, and documentation processes in linguistic annotation through a visually accessible web-based platform designed to reduce learning curves and enhance user experience.

Results: LAWS offers a straightforward alternative to traditional tools, addressing usability gaps and providing linguists with an efficient, easy-to-navigate solution for their annotation tasks.

Conclusions: By simplifying the linguistic annotation process, LAWS empowers linguists with an intuitive tool, promoting productivity and reducing the barriers associated with complex annotation software.

CCS Concepts

• **Information systems → Multimedia content creation**; **Open source software**; **Markup languages**; **Extensible Markup Language (XML)**.

Keywords

Language Annotations, Web Systems, ELAN Format, REACT, XML, JSON

1 INTRODUCTION

One approach to preserving endangered languages is language documentation. According to Austin (2006, 89) [3], language documentation consists of a series of stages, some of which occur in parallel. These stages are: recording media; capturing analogue materials; analysis (transcription, translation, annotation); archiving; and mobilization (publication and distribution). Language endangerment threatens the loss of community-specific social practices, including rituals, traditional songs, and oral narratives. Thus, the recording, proper storage, transcription, and translation of these practices are crucial for both the linguistic community and humanity at large.

According to estimates by linguists Austin and Sallabank (2011) [4], approximately 7,000 languages are spoken globally. However, they assert that "at least half of these languages may not survive beyond a few generations, as they are no longer being learned by children as first languages." These languages are classified as endangered languages. Estimates of the number of Indigenous languages in Brazil range from 274 [1] to 150 [14], most of which have extremely low speaker numbers [15]. The process of documenting endangered languages typically begins with linguists conducting linguistic fieldwork to collect audio or video recordings. These recordings are then transcribed and translated in a process known as linguistic annotation. Annotations can be categorized as minimal or elaborated. Minimal annotations involve transcription and free translation accompanied by primary data (audiovisual recordings)[12].

While ELAN facilitates both minimal and elaborated annotations, the precise extent and format of annotations required for each session remain topics of debate. Minimal annotation schemes typically consist of a transcription and a free translation for a substantial number of primary data segments, while more elaborated schemes may include interlinear glosses, grammatical commentary, ethnographic notes, and extensive cross-references between sessions and resources. Annotations are created using specialized computational tools. Among the most powerful tools for transcription, translation, and subsequent linguistic analysis is the ELAN software[2] [22, 26]. ELAN allows users to view or listen to media (audio and/or video) while making annotations [25]. Originally designed for analyzing sign languages and gestures, ELAN supports linking annotations to media streams, as well as defining custom relationships between annotation layers [16].

Despite its powerful features, ELAN has limitations, particularly for novice users [6]. Its interface is intricate, requiring training to use effectively. For example, in a project documenting Indigenous languages in northern Amapá, Campetela et. al. (2017) [8] provided workshops and training in tools like ELAN to equip Indigenous teachers with the theoretical and technical skills necessary for language documentation within their communities.

Additionally, ELAN is not ideal for presenting materials to non-linguistic audiences. As pointed out by Nogueira and Silva (2021), "One challenge, however, is that handling texts in ELAN and FLEx can become difficult for an untrained audience, due to the technical details, requiring program installation, navigation, and knowledge of the buttons, the editing areas of each editing area, fields, menus, and submenus" [9]. Such materials often attract interest from diverse fields, including anthropology, history, musicology, art, and the general public, who may lack technical expertise in linguistic documentation.

[1] available at https://indigenas.ibge.gov.br/estudos-especiais-3/o-brasil-indigena/lingua-falada, Accessed on March 5th, 2025.

[2] ELAN (Version 6.9) [Computer software]. (2024). Nijmegen: Max Planck Institute for Psycholinguistics, The Language Archive. Retrieved from https://archive.mpi.nl/tla/elan

Professional linguistic annotation tools are often complex to learn and manage. However, advancements in web technologies have given rise to new computational tools. Modern web applications are typically built using HTML, CSS, and JavaScript, with numerous JavaScript libraries and frameworks emerging over the years. One prominent framework is **Node.js**, which facilitates server-side and client-side application development.

Recent efforts have sought to create web-based annotation systems [21]. For instance, Dobrin and Ross (2017) [10] introduced the IATH ELAN Text-Sync Tool (ETST), which enables time-synchronized display of annotations [10]. ETST utilizes:

(1) An HTML file to transform XML-based ELAN data into web-readable content,
(2) A JavaScript file to manage interactive features like scrolling and playback controls, and
(3) A Cascading Style Sheet (CSS) file for formatting instructions.

Similarly, Pride *et al.* (2020) developed LingView [19], a web interface for viewing ELAN and FLEx (FieldWorks Language Explorer) files. LingView leverages the **React.js** framework, a JavaScript library optimized for building modular and interactive user interfaces [2]. LingView converts ELAN and FLEx files into JSON, which is then rendered as HTML by React.js on the client side, with back-end processing in Node.js.

The documentation of endangered languages is an exhaustive endeavor. Reiman (2010) reports that annotating half a day's recordings (approximately three hours) can require up to 33 hours of work [20]. Therefore, the involvement of community members—such as Indigenous individuals with computer skills—is invaluable.

Nonetheless, training in professional annotation tools like ELAN for beginners requires investment in fieldwork, instructors, interested people, time, and equipment, posing a challenge that demands innovative solutions.

This study is primarily intended to contribute to the linguistic annotations (specifically, transcription and translation) of primary data from the Wayoro language and culture documentation project [18]. With only one elderly fluent speakers and less than 20 semi-speakers remaining, documenting this language is an urgent task. However, any project that requires linguistic annotation will benefit from this contribution.

We propose the Language Annotation Web System (LAWS), a lightweight and user-friendly web tool for transcribing and annotating audiovisual materials. Unlike ELAN, LAWS operates entirely on the client side using JavaScript modules, without requiring a web server or database. It supports two annotation layers: transcription and translation. LAWS reuses code from LingView [19], which itself integrates components from open-source projects [10].

LAWS plays a crucial role in advancing social equity, inclusion, and diversity by democratizing access to linguistic annotation tools. Traditional annotation systems, while powerful, often present steep learning curves and complex interfaces that can alienate novice users and underrepresented groups in linguistics. By prioritizing simplicity, intuitiveness, and accessibility, LAWS ensures that linguistic annotation—an essential process for language documentation—becomes more inclusive and equitable. This innovation not only reduces barriers for linguists from diverse backgrounds but also fosters collaborative efforts in documenting and analyzing underrepresented languages, thereby contributing to a more inclusive understanding of global linguistic diversity.

The contributions of this work are:

- Requirement analysis and functional design of a tool leveraging LingView's code, adapted for client-side processing;
- Development of LAWS, a web-based annotation system using React.js for linguistic annotation;
- Evaluation of LAWS for functionality and usability;
- Creation of a user manual for LAWS.

This article is divided into six sections. Section 1 presents the introduction. Section 2 outlines the fundamental concepts essential for understanding this project. In Section 3, the most relevant related works are discussed. Section 4 details the methodology, decisions, and steps followed for the agile development of the proposed web system. Section 5 presents the results, and finally, Section 6 provides the conclusions.

2 Concepts and Theoretical Background

2.1 Language Documentation

Language documentation, in essence, refers to the creation of a durable and multifunctional record of a language. This process involves the collection of audio and/or video recordings, supplemented by linguistic annotations that enable comprehensive understanding [12].

According to Himmelmann (2006) [12], annotations can be categorized into two types: minimal and complex. Minimal annotations include only the transcription and translation of the recorded material, while complex annotations—often referred to as descriptive annotations—involve in-depth analyses such as grammatical structures, cross-linguistic comparisons, and other linguistic features.

Himmelmann (2006) [11] further define linguistic annotation as the transcription and translation of audiovisual recordings in a given language. Audiovisual recordings are considered primary linguistic data, whereas transcriptions, annotations, and associated metadata (e.g., grammatical descriptions) constitute the corpus apparatus, also called secondary data [7].

2.2 Concepts of the LAWS Software Architecture

The development of the LAWS tool leverages popular JavaScript libraries. Which is commonly defined as "A collection of functions, methods, and objects aimed at simplifying and accelerating application development. By reusing pre-existing functions, developers can avoid repetitive coding. Libraries are generally less complex and more flexible than frameworks."

The core technology underlying the LAWS architecture is **ReactJS**, a JavaScript library for building user interfaces. Known for its efficiency and rapid development capabilities, ReactJS features intuitive documentation, making it accessible for beginners. The official ReactJS documentation can be found on its website.

ReactJS operates by creating a virtual DOM (Document Object Model) to facilitate the manipulation of webpage elements. The virtual DOM is an in-memory copy of the actual DOM, enabling intelligent updates to the browser by applying only the necessary changes. This approach significantly improves page manipulation efficiency.

Another critical library used in LAWS is **Redux**, a JavaScript library for managing application state. Inspired by Facebook's Flux architecture, Redux can be integrated with React or other JavaScript libraries. Created by Dan Abramov and Andrew Clark in 2015, Redux centralizes application data [1], represented in LAWS as JSON files.

During development, ReactJS code is compiled using the Node.js framework, a server-side JavaScript runtime built on Google's V8 engine. Node.js and V8 are primarily implemented in C and C++, focusing on performance and low memory consumption [24].

JavaScript supports the concept of modules, which are files containing specific classes or functions. These modules can be imported using the require keyword. However, web browsers do not natively support require. To address this limitation, the LAWS tool employs Webpack, a module bundler that generates a new file by resolving dependencies recursively. This bundled file is browser-compatible and contains the original code.

Webpack operates based on five main concepts:

- **Entry:** The entry point where Webpack begins constructing the internal dependency graph (e.g., ./src/App.js for LAWS).
- **Output:** Specifies the name and location of the generated bundle.
- **Loaders:** Webpack processes only JavaScript files by default; loaders enable support for other file types, converting them into valid modules for inclusion in the dependency graph.
- **Plugins:** Extend Webpack's functionality for specific tasks.
- **Mode:** Defines the environment—development mode for ongoing development or production mode for finalized tools.

By leveraging these technologies and concepts, the LAWS tool achieves efficient development, robust performance, and streamlined deployment.

3 Related Works

This section presents projects or tools related to the study, along with their main features. The tools discussed include ELAN, LingView, the ELAN Text-Sync Tool (ETST), and the Tycho Brahe Platform.

3.1 ELAN

ELAN (Eudico Linguistic Annotator) is the most widely used professional tool in the research community for documenting and analyzing linguistic materials [25]. Developed by the Max Planck Institute for Psycholinguistics, ELAN organizes annotations into layers, referred to as "Tiers." These tiers can have various relationships, such as being independent, aligned, or embedded. Figure 1 illustrates the ELAN interface.

The ELAN documentation highlights several key features of the tool:

- High Precision: ELAN enables annotations with a maximum precision of 1 millisecond, ensuring the accuracy required by linguists. Combined with its precise playback capabilities, this allows for highly accurate annotations.
- Media Support: ELAN supports video files in '.mpg' and '.mp4' formats and audio files in '.wav' format.
- Flexibility: The tool allows different videos to be added or removed from the document at any stage of transcription, offering users maximum flexibility in the annotation process.

However, ELAN requires training for new users, making it challenging for beginners to master this professional tool.

3.2 LingView

LingView is a web-based tool designed to display linguistic materials to audiences who may not belong to the linguistic research community but are interested in language documentation [19]. Figure 2 illustrates LingView in action.

The LingView processing pipeline is shown in Figure 3. Pride et al. (2020) outline the pipeline's design to meet specific requirements:

- Information Preservation: Ensures no data is lost during processing.
- Cross-Platform Performance: Enables the site to load quickly and function accurately across various platforms.
- Offline Functionality: Allows the site to operate without an internet connection.

LingView supports both ELAN and FLEx files. The first step in the pipeline is the **preprocessing** of these files, converting them into a JSON format. Once in JSON format, the data is displayed on an HTML page using JavaScript code.

The preprocessing script, implemented as a Node.js package, converts FLEx and ELAN files into a unified JSON format. Figure 4 illustrates the JSON structure, with metadata and sentence parameters summarized below:

- Metadata: Includes author information, title, date, audio/video media, and speaker IDs (e.g., "S1" and "S2"). Detailed metadata specifications can be found in Pride et al. (2020).
- Sentences: Lists annotations derived from ELAN tiers or FLEx sentences, each with associated attributes like speaker, tier, and start/end times.

Additionally, LingView uses a secondary JSON file, 'database.json', which lists stories and provides an index for generating a story catalog. This file is integral to the site generation process. LingView follows a client-server model and is built using ReactJS, a component-based JavaScript library. Figure5 showcases a component displaying JSON data.

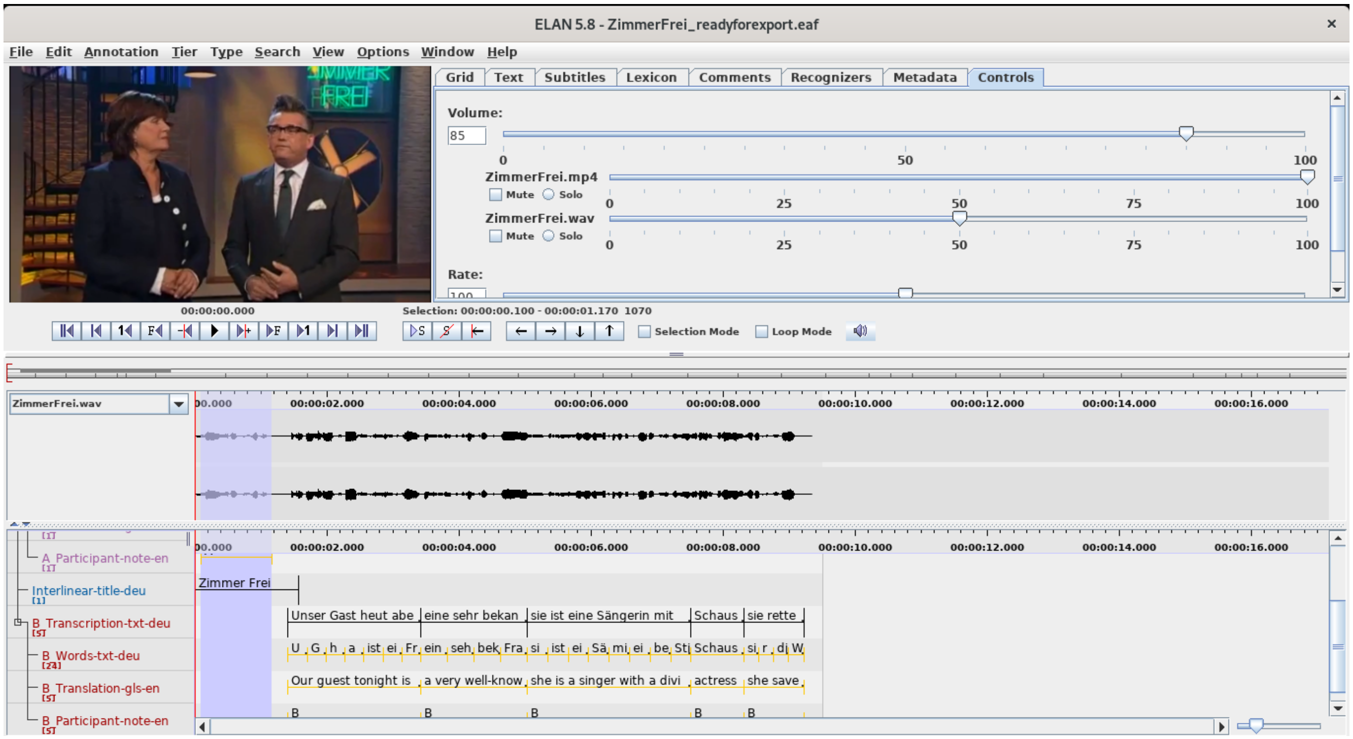
To install LingView, the following steps are required:

(1) Download the project from https://github.com/BrownCLPS/LingView.git.
(2) Extract the project files.
(3) Install Node.js and necessary packages using the command 'npm install'.

While LingView is an excellent tool for displaying linguistic materials, its installation process may pose challenges for beginners. Further details are available in Pride et al. (2020).

3.3 ELAN Text-Sync Tool (ETST)

ETST is a browser-based transcription playback tool designed to function as a plug-and-play solution or as a foundation for further development. Built entirely with standards-compliant HTML, ETST is efficient, reliable, and compatible across systems [10], web interface of ETST tool is shown in Figure 6. It is responsive to various



A_Participant-note-en [1]

Interlinear-title-deu [1]: Zimmer Frei

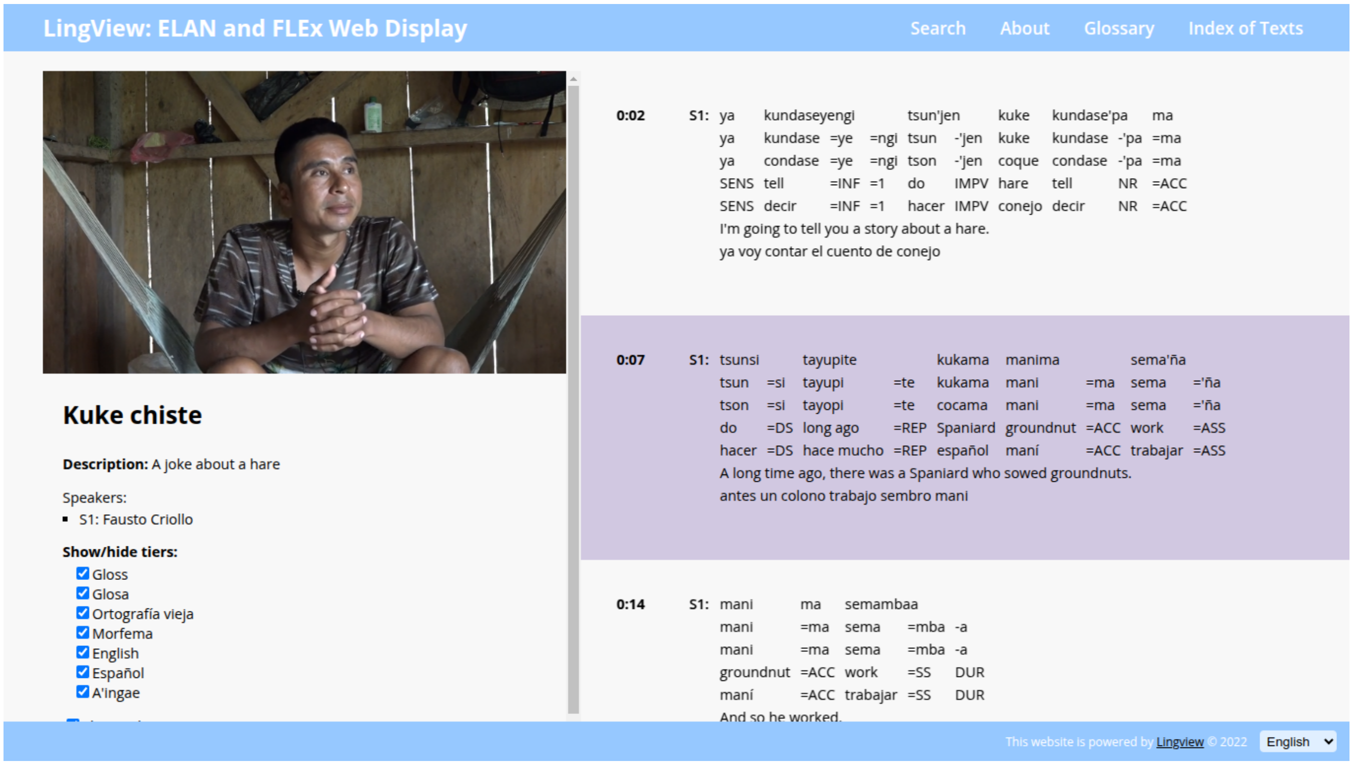
B_Transcription-txt-deu [5]: Unser Gast heut abe | eine sehr bekan | sie ist eine Sängerin mit | Schaus | sie rette

B_Words-txt-deu [24]: U | G | h | a | ist | ei | Fr | ein | seh | bek | Fra | si | ist | ei | Sä | mi | ei | be | Sti | Schaus | si | r | di | W

B_Translation-gls-en [5]: Our guest tonight is | a very well-know | she is a singer with a divi | actress | she save

B_Participant-note-en [5]: B | B | B | B | B

Figure 1: Graphical User Interface of ELAN. https://archive.mpi.nl/tla/elan


LingView: ELAN and FLEx Web Display
Search
About
Glossary
Index of Texts
This website is powered by Lingview © 2022
English

Kuke chiste

Description: A joke about a hare

Speakers:

- S1: Fausto Criollo

Show/hide tiers:

☑ Gloss
☑ Glosa
☑ Ortografía vieja
☑ Morfema
☑ English
☑ Español
☑ A'ingae

```
0:02   S1:  ya    kundaseyengi          tsun'jen       kuke    kundase'pa     ma
            ya    kundase  =ye   =ngi   tsun    -'jen  kuke    kundase  -'pa  =ma
            ya    condase  =ye   =ngi   tson    -'jen  coque   condase  -'pa  =ma
            SENS  tell     =INF  =1     do      IMPV   hare    tell     NR    =ACC
            SENS  decir    =INF  =1     hacer   IMPV   conejo  decir    NR    =ACC
            I'm going to tell you a story about a hare.
            ya voy contar el cuento de conejo

0:07   S1:  tsunsi      tayupite           kukama    manima           sema'ña
            tsun   =si  tayupi      =te    kukama    mani       =ma   sema      ='ña
            tson   =si  tayopi      =te    cocama    mani       =ma   sema      ='ña
            do     =DS  long ago    =REP   Spaniard  groundnut  =ACC  work      =ASS
            hacer  =DS  hace mucho  =REP   español   maní       =ACC  trabajar  =ASS
            A long time ago, there was a Spaniard who sowed groundnuts.
            antes un colono trabajo sembro mani

0:14   S1:  mani       ma    semambaa
            mani       =ma   sema      =mba  -a
            mani       =ma   sema      =mba  -a
            groundnut  =ACC  work      =SS   DUR
            maní       =ACC  trabajar  =SS   DUR
            And so he worked.
```

Figure 2: Web interface of the tool Lingview. https://brownclps.github.io/LingView/#/index

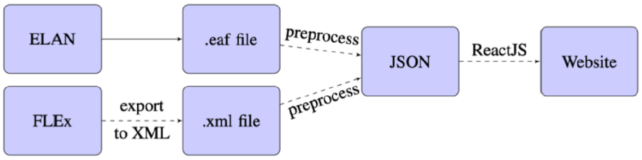


Figure 3: Pipeline of LingView tool

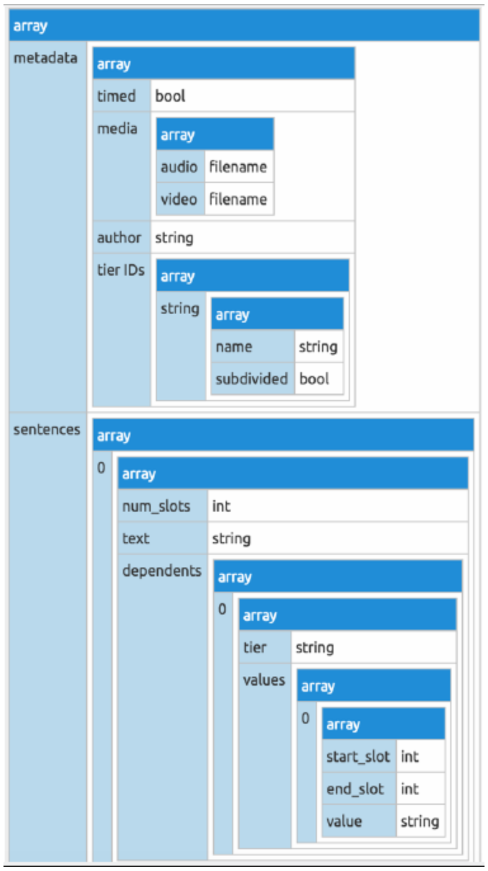


Figure 4: JSON data structure of the XML ELAN Format

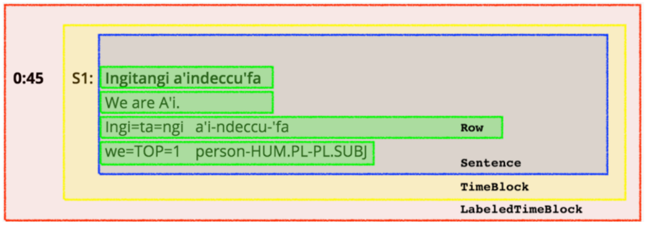
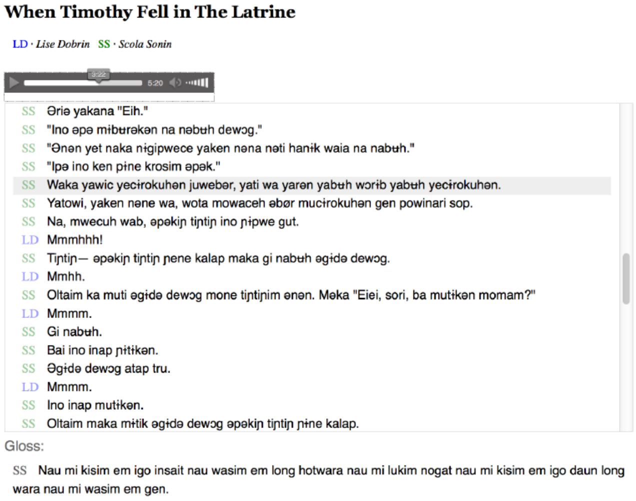


Figure 5: JavaScript Sentence Component Representation

digital environments and adaptable to different devices and screen resolutions.

The system comprises three main components:


When Timothy Fell in The Latrine

LD · *Lise Dobrin* SS · *Scola Sonin*

SS Əriə yakana "Eih."
SS "Ino əpə mɨbʉrəkən na nəbʉh dewɔg."
SS "Ənən yet naka nɨgipwece yaken nəna nəti hanɨk waia na nabʉh."
SS "Ipə ino ken pɨne krosim əpək."
SS Waka yawic yecɨrokuhən juwebər, yati wa yarən yabʉh wɔrɨb yabʉh yecɨrokuhən.
SS Yatowi, yaken nəne wa, wota mowaceh əbør mucɨrokuhən gen powinari sop.
SS Na, mwecuh wab, əpəkiɲ tiɲtiɲ ino ɲɨpwe gut.
LD Mmmhhh!
SS Tiɲtiɲ— əpəkiɲ tiɲtiɲ ɲene kalap maka gi nabʉh əgɨdə dewɔg.
LD Mmhh.
SS Oltaim ka muti əgɨdə dewɔg mone tiɲtiɲim ənən. Məka "Eiei, sori, ba mutɨkən momam?"
LD Mmmm.
SS Gi nabʉh.
SS Bai ino inap ɲɨtɨkən.
SS Əgɨdə dewɔg atap tru.
LD Mmmm.
SS Ino inap mutɨkən.
SS Oltaim maka mɨtik əgɨdə dewɔg əpəkiɲ tiɲtiɲ ɲɨne kalap.

Gloss:

SS Nau mi kisim em igo insait nau wasim em long hotwara nau mi lukim nogat nau mi kisim em igo daun long wara nau mi wasim em gen.

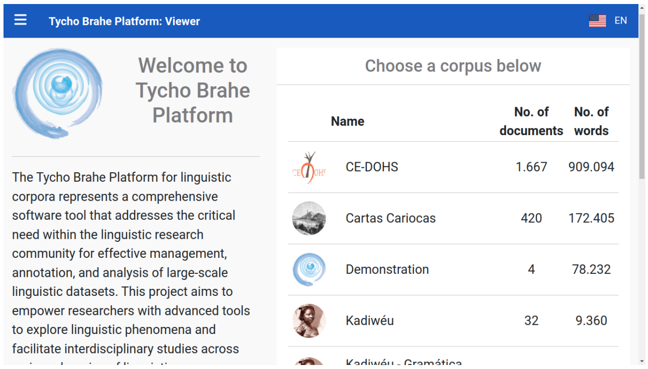
Figure 6: Web Interface of ETST Tool

- Preprocessing Script (‘txt sync.php‘): Converts XML and ELAN files into HTML-readable formats.
- JavaScript File (‘txt sync.js‘): Handles interactive screen features such as scrolling and play/pause functionality.
- Cascading Style Sheet (‘txt sync.css‘): Provides formatting instructions for the display.

Although ETST is a simple visualization tool, its data processing is not robust. Users often need to modify ELAN input files to conform to the system’s tier structure specifications, which can be challenging for beginners. This limitation underscores a need for enhanced data compatibility in future versions.

3.4 Tycho Brahe Platform

The Tycho Brahe Platform offers tools for creating, editing, managing, and analyzing syntactically annotated corpora. The platform includes integrated point-of-view (PDV) identifiers and syntactic analysis tools (<https://www.tycho.iel.unicamp.br/home>).



Welcome to Tycho Brahe Platform

The Tycho Brahe Platform for linguistic corpora represents a comprehensive software tool that addresses the critical need within the linguistic research community for effective management, annotation, and analysis of large-scale linguistic datasets. This project aims to empower researchers with advanced tools to explore linguistic phenomena and facilitate interdisciplinary studies across

Choose a corpus below

Name	No. of documents	No. of words
CE-DOHS	1.667	909.094
Cartas Cariocas	420	172.405
Demonstration	4	78.232
Kadiwéu	32	9.360
Kadiwéu - Gramática		

Figure 7: Online visualization of corpora available on the Tycho Brahe platform

Key features of the platform include:

- Tycho Brahe Browser: Allows users to navigate available corpora. Figure 7 shows a document in the Kadiwéu language, a Guaikuru language spoken in Mato Grosso do Sul, Brazil.
- Tycho Brahe eDictor: A tool for annotation editing. Access is restricted (https://www.tycho.iel.unicamp.br/edictor/unauthorized).
- Tycho Brahe Search: Enables linguistic searches within available corpora (https://www.tycho.iel.unicamp.br/search/).
- Tycho Brahe Parser: Provides parsing and interpretation mechanisms. Access is restricted (https://www.tycho.iel.unicamp.br/parser).

While the Tycho Brahe Platform is highly comprehensive, its tools for annotation systems require special access permissions. Another limitation is its lack of support for video files, which restricts its applicability in certain contexts.

This overview highlights the strengths and limitations of the tools discussed, emphasizing their relevance and areas for improvement in linguistic research workflows.

4 Methodology

This section details the design, development, and implementation of LAWS: Language Annotation Web System Using ELAN. LAWS is designed to support linguistic annotation creation by users without requiring prior training. The methodology employed in this project is the case study approach, aiming to thoroughly examine and detail each stage of the work. The methodology is divided into system requirements, business rules, and implementation phases to ensure a systematic approach to achieving the project objectives.

Based on the Interactive Software Development Model, a specialized adaptation of this model was proposed for the development of the tool. It is worth noting that other models could have been adopted instead of the Interactive Model. This choice was made due to the simplicity and limited functionality of the LAWS tool, as well as the initial lack of clearly defined requirements. This necessitated iterative user interaction with the system to uncover their actual needs [23].

4.1 System Overview

LAWS is a web-based platform designed to streamline the annotation of linguistic data using ELAN-compatible files. The system enables users to upload audio or video media and XML-based annotation files (EAF format), facilitating the editing and management of linguistic annotations such as transcription and translation. It provides a tool that is less complex and less powerful than ELAN, but it is meant to complement this program without attempting to replace them.

While creating a web tool with significant processing capabilities would typically require a backend (remote server), limitations were introduced to manage media input to a maximum duration of 10 minutes. This approach facilitates a lightweight tool and encourages users to plan recordings and organize materials into thematic modules, such as culture, cuisine, and traditions.

The core functionalities of LAWS include:

- **Media File Management:** Upload, preview, and validate audio (MP3, WAV) and video (MP4, MPEG-4, WebM) files.
- **Annotation Processing:** Upload and validate EAF files to ensure compatibility with ELAN standards.
- **Annotation Management:** View existing annotations with details on Participant, Start Time, End Time, Transcription, and Translation. Edit transcription or translation fields based on user input.
- **Conflict Detection:** Identify and resolve overlapping time intervals in annotations to maintain data consistency.

4.2 Requirements Gathering and Definition

This phase involves analyzing requirements based on user needs, defining new functionalities for the software. Since the preprocessing (backend) was integrated into the frontend, only the requirements of the web platform are detailed here.

The requirements for LAWS were specified by a linguistic researcher of Federal University of Pará which is not mentioned for double blind review, a linguist specializing in the documentation of the Wayoro indigenous language. Requirements, use cases, and business rules were identified and associated with use cases (UC), see Figure 8, wich are designed from Table 1.

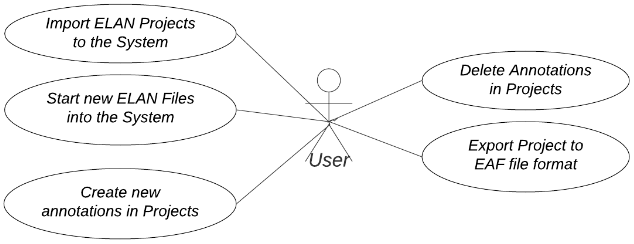

Import ELAN Projects to the System
Start new ELAN Files into the System
Create new annotations in Projects
User
Delete Annotations in Projects
Export Project to EAF file format

Figure 8: Use Cases Diagram of LAWS

UC	Description
UC01	Create New Project
UC02	Open Project
UC03	Delete Annotation
UC04	Insert Annotation
UC05	Edit Annotation
UC06	Export to EAF (ELAN Annotation Format)

Table 1: Minimum Functionalities and Use Cases of LAWS

The implementation adheres strictly to the following business rules:

- Audio media files must be in one of the following formats: MP3 or WAV.
- Video media files must be in one of the following formats: MP4, MPEG-4, or WebM.
- Both file input fields must be completed.
- The XML input file must adhere to the EAF format.
- All fields (Participant, Start Time, End Time, Transcription, and Translation) must be filled out.

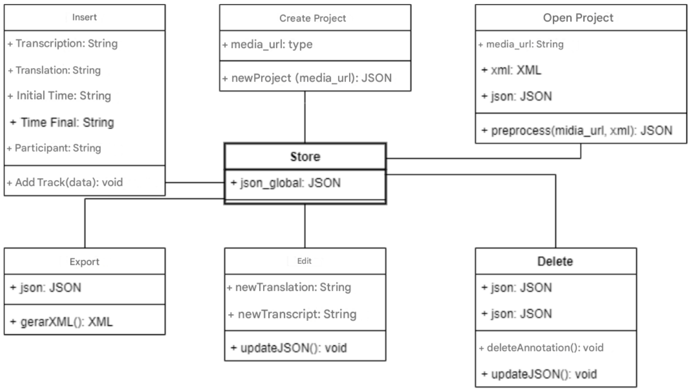


Figure 9: UML Class Diagram of LAWS

- The Start Time and End Time of a new annotation must not conflict with or overlap the Start Time and End Time of existing annotations.
- The desired field for editing (either Transcription or Translation) must be completed.

A ReactJS class diagram is presented in Figure 9. Each functionality is encapsulated in its class, leveraging React components. Redux is employed at the core to manage a global JSON object that coordinates application states.

4.3 Web Adaptation and Frontend

The frontend is lightweight, and adapting backend functionalities to the frontend required removing high-processing elements, such as large media handling and database operations. Redux was introduced for efficient state management, dynamically updating components whenever the global state changes.

Existing professional tools like ELAN offer extensive features but often require user training. The user interface (UI) was designed with simplicity and usability in mind, allowing users to upload, preview, and annotate files efficiently. To address this challenge, the LAWS tool focuses on providing only essential functionalities through a user-friendly interface.

- Open Project: The user selects input files (XML and media) to open a project. The system verifies media compatibility; if supported, annotations are displayed alongside the media. Otherwise, an error message is shown.
- Create Project: For creating a project, the user provides only the media file. The system verifies compatibility and either creates a new project or prompts the user to attach a compatible media file.
- Insert Annotation: The user insert a new annotation with its transcription and traduction.
- Delete Annotation: An annotation is deleted.
- Edit Annotation: User can modify a previous annotation.
- Export: The project can be export in any moment in format .EAF wich is completely comptible with ELAN Software.

File validation scripts ensure compliance with supported formats (MP3, WAV, MP4, MPEG-4, WebM) and EAF standards. Invalid files trigger immediate feedback to users, detailing the reasons for rejection.

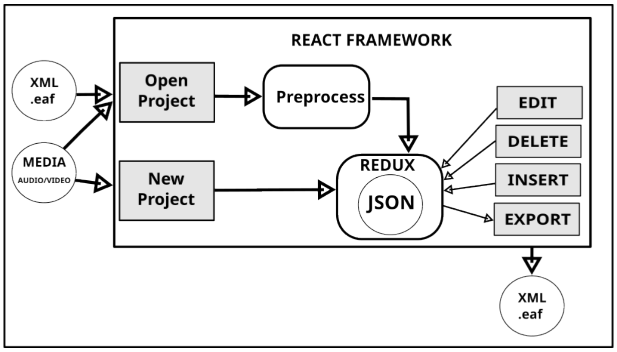


Figure 10: Software Architecture of LAWS

EAF files are parsed to extract annotation metadata, which is displayed in an interactive table format. The table allows users to sort and filter annotations by fields such as Participant and Start Time.

Users can select specific annotations for editing. The desired field (Transcription or Translation) must be completed before saving changes. Conflict detection algorithms ensure that new or updated annotations do not overlap with existing ones. Conflicting annotations are flagged for user review.

A dataset of annotated media files was used to evaluate system performance and accuracy, including handling edge cases such as overlapping time intervals and missing fields.

4.3.1 Backend and Database. A modular backend architecture ensures scalability and maintainability. The system is built using NodeJS. The backend server, provides high-performance services. NodeJS modules like ‘fs‘ (for file system operations) and libraries such as ‘XML2js‘ (for converting XML to JSON) are integral components.

The backend employs a JSON file (‘database.json‘) as its database, eliminating the need for complex database management systems. This choice simplifies installation for users with limited technical expertise.

Annotations and metadata (e.g., project title, media title, participant names) are stored in a global JSON object managed by Redux, ensuring seamless data accessibility across components. Figure 10 shows the software architecture of the Language Annotation Web system (LAWS), main contribution of this project.

4.3.2 Deployment and Tool Hosting. The system is hosted and deployed on Github Pages and, chosen for its free services and ease of updates. It can be accessed in this link https://marcosamaris.github.io/laws. LAWS functionalities were tested using Google Chrome and Mozilla Firefox browsers on personal computers. It was also tested using the Google Chrome browser on Android phones and it works perfectly.

4.4 Tests During Development

Testing was conducted by 2 specialized linguist researchers, using primary audiovisual files from linguistic fieldwork. Researchers interacted with the tool without developer instructions, identifying errors and providing suggestions, summarized as follows:

- Annotation sync issues: Fixed by adding 'setupTextSync' calls to React's 'componentDidUpdate()' lifecycle method.
- Large media files not loading: Imposed a 10-minute duration limit for media uploads.
- Lack of user manual: Added a step-by-step guide on the "Open Project" page.
- Editing function errors: Adjusted logic to independently verify and update transcription and translation fields.
- Route change warnings: Added alerts prompting users to save documentation before navigating away.
- Disable edit button: Added a button to close the editing frame when users finish or cancel editing.
- Homepage text adjustments: Incorporated feedback to enhance introductory text.

5 RESULTS AND USE OF LAWS

This section presents the interface of LAWS, explains its functionalities, and discusses qualitative results.

The home page of LAWS is shown in Figure 11. At the top navigation bar, there are four links: **Home** (LAWS home page), **Open Project**, **New Project** and **Manual**. ELAN projects can be opened using a ELAN file (eaf extension) and its respective audio or video file. And, new projects can be started only from a audio or video file.

Figure 12 shows LAWS when a project is started and the annotations can be edited, after importing and processing a project. On the left side, there is an HTML form for entering information to create a new annotation. The required fields include: participant, start time, end time, transcription, and translation. Below the HTML form, a button labeled **Export XML** is available, generating a file named 'Data.eaf' in XML format. This exported file can be used in ELAN for deeper analysis.

Below the annotation form, a media component is displayed, implemented with an HTML video tag for loading audiovisual files. This media player includes standard controls such as pause, a duration slider, volume control, and more.

On the right side, existing annotations are listed, along with their transcriptions, translations, participants, and buttons for Edit and Delete. Deleting an annotation is straightforward—simply click the **Delete** button under the respective annotation, as illustrated in Figure 12. Similarly, clicking **Edit** opens a small form below the annotation, enabling transcription and translation updates.

As outlined in Section 1, the focus of this study was on the system's ease of interaction. The LAWS tool is accessible at the following URL: https://marcosamaris.github.io/laws. Figure 13 shows the graphical User interface of LAWS in android smarthphone using the computer version option in Google Chrome Browser.

5.1 Experiments and LAWS Use Cases with an Endangered Language

This section presents qualitative results from a linguist documenting an endangered language in the Brazilian Amazon, the Wayoro language, spoken by the Wajuru people.

The Wajuru people reside in the state of Rondônia, Brazil, specifically in the Rio Guaporé Indigenous Territory (municipality of Guajará-Mirim) and Porto Rolim de Moura (municipality of Alta Floresta d'Oeste) (Nogueira, Macurap & Wajuru, 2019). The coordinates of this place are latitude -11.964 and longitude -64.709.

Since 2008, Nogueira has been conducting linguistic research and documentation of the Wayoro language [17]. According to her studies, only one fluent speakers and less than twenty remain, all living in the Rio Guaporé Indigenous Territory or Porto Rolim de Moura. The Wayoro Language Documentation Project, funded by the Endangered Languages Documentation Programme (Grant ID: MDP0435)[3], is set to run from 2022 to 2025. The project aims to create a collection of recordings that capture Wayoro culture and language, along with a corpus of linguistic annotations. Nogueira tested the LAWS tool while documenting the Wayoro language, as we can see in Figure 12[4].

The tests demonstrated that using LAWS significantly enhanced the experience of creating and modifying ELAN projects. The time required for multimedia transcription and translation was substantially reduced. The user highlighted that deleting and adding annotations in LAWS was faster and more intuitive than in ELAN. After completing the transcription and translation phases, the ELAN tool could then be used for more advanced analyses.

Nogueira and Silva (2021) reported that the Ricardo Franco village school received broadband internet access in 2020 [9]. In an ideal linguistic documentation scenario, the community itself would conduct recordings and create annotations, with routine cloud backups [5]. LAWS could play a pivotal role in this context by offering an accessible transcription and translation tool tailored to the needs of indigenous users.

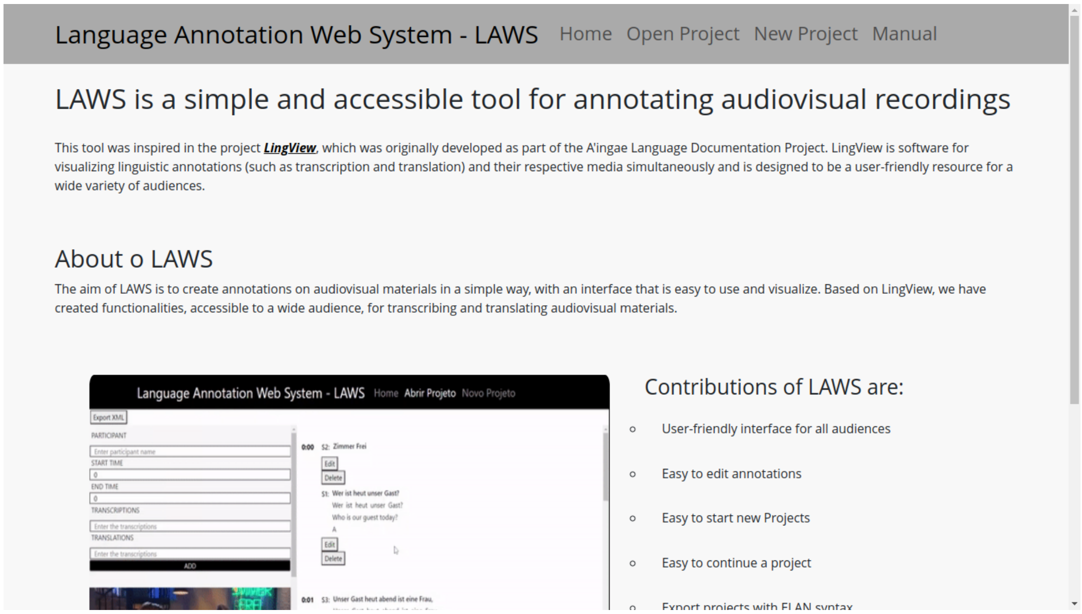
We asked a young Wajuru researcher to test LAWS in the Rio Guaporé Indigenous Land. In addition to finding the program "very practical" and "very interesting", Jaqueline Wajuru pointed out: "Besides, the program is very useful for us, because you can open it on your cell phone. As you know [Nogueira], it's much better for us here in the village because it's a bit difficult to have access to a computer or any device other than a cell phone".

5.2 Test of Usability with Users

We conducted a test with undergraduate students from the 6th semester onwards from the Federal University of Pará's Faculty of Letters and fellows of the scientific initiation program in the linguistics area of the Emílio Goeldi Museum. Previously, an instructional video was shown to the Participants to explain the LAWS tool and its basic functionalities. Then, they were tasked with transcribing and translating a 60-second video of an English or Portuguese speaker using LAWS. After the experiment, participants completed a survey with three questions about their experiences and suggestions for improvement. Sixteen students participated, and we provide both quantitative and qualitative analyses of their responses below.

[3] See Collection Handle http://hdl.handle.net/2196/5004e53b-79f6-440d-81e6-266a64579366

[4] Recording available at http://hdl.handle.net/2196/cd808a4d-41b9-4bdb-bf7b-612f246c4d19



LAWS is a simple and accessible tool for annotating audiovisual recordings

This tool was inspired in the project *LingView*, which was originally developed as part of the A'ingae Language Documentation Project. LingView is software for visualizing linguistic annotations (such as transcription and translation) and their respective media simultaneously and is designed to be a user-friendly resource for a wide variety of audiences.

About o LAWS

The aim of LAWS is to create annotations on audiovisual materials in a simple way, with an interface that is easy to use and visualize. Based on LingView, we have created functionalities, accessible to a wide audience, for transcribing and translating audiovisual materials.

0:00 S2: Zimmer Frei

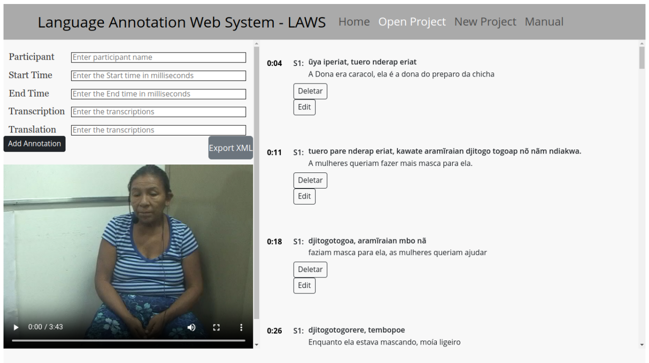
S1: Wer ist heut unser Gast?
Wer ist heut unser Gast?
Who is our guest today?
A

0:01 S3: Unser Gast heut abend ist eine Frau,

Contributions of LAWS are:

- User-friendly interface for all audiences
- Easy to edit annotations
- Easy to start new Projects
- Easy to continue a project
- Export projects with ELAN syntax

Figure 11: Main Interface of LAWS



0:04 S1: ũya iperiat, tuero nderap eriat
A Dona era caracol, ela é a dona do preparo da chicha

0:11 S1: tuero pare nderap eriat, kawate aramĩraian djitogo togoap nõ nãm ndiakwa.
A mulheres queriam fazer mais masca para ela.

0:18 S1: djitogotogoa, aramĩraian mbo nã
faziam masca para ela, as mulheres queriam ajudar

0:26 S1: djitogotogorere, tembopoe
Enquanto ela estava mascando, moía ligeiro

Figure 12: Tests of LAWS with a single speaker of the Indigenous Wajuru Language

(1) **Did you find the LAWS user interface user-friendly?** The majority (81%) rated the interface as user-friendly, praising its intuitiveness and clean design. Responses included: "Yes, very user-friendly", "very intuitive", and "simple to navigate." Two negative and one neutral response were recorded. One participant noted initial complexity but eventual ease of use, stating: "Although complex at first, once you get used to it, it's easy."

(2) **What did you think of the experience of using LAWS for linguistic annotation?** Most respondents praised LAWS for its usability, though some highlighted challenges in complex or non-standard scenarios. Comments like "hard to get lost" and "quick notes" underscored the tool's productivity benefits. A participant observed that LAWS is "useful for languages with defined orthography" but less effective for those in preliminary linguistic analysis.

(3) **Do you have any suggestions for improving LAWS?** Suggestions focused on:
Time interaction: Allowing finer adjustments (e.g., seconds instead of milliseconds).
User experience: Enhancing the interface and adding detailed tutorials.
Functionality: Expanding video controls and language support.

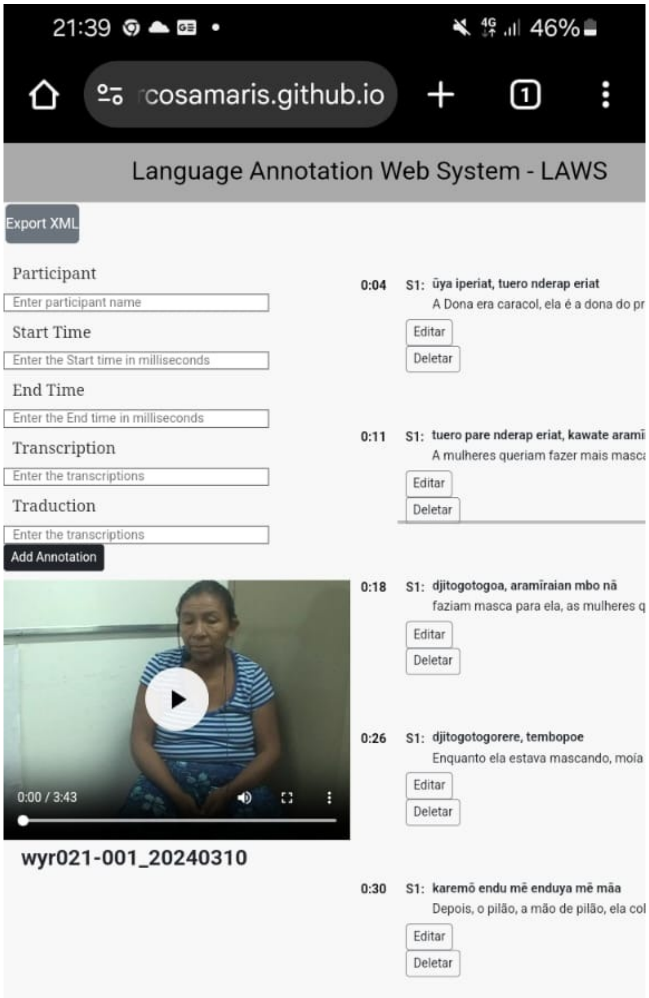
6 Conclusions and Suggestions for Future Work

In this paper, we presented a user-friendly web-based system for manual linguistic annotation of audiovisual files, named LAWS (Language Annotation Web System). The tool was implemented using a web architecture, making it accessible online. All functionalities are executed on the client side, leveraging JavaScript. The development process utilized the React framework along with libraries such as Redux, Bootstrap, and Webpack for code modularization.

JavaScript code blocks were employed to import XML specifications into JSON files, which are subsequently used to render HTML elements in the browser. This approach allows users to interactively add and delete transcriptions and time-aligned translation annotations for audiovisual files.

This project provided the team with significant learning opportunities, particularly in coding concepts. These achievements were made possible through the dedication of the participants. Beyond developing a functional product, the project offered valuable insights into the challenges faced by students as they prepare for future careers in the job market. We believe this project holds great importance as it represents a practical solution with direct applicability in society.

For future work, we aim to enhance the system by developing new features. We are currently working on a mobile application



wyr021-001_20240310

0:04 S1: ũya iperiat, tuero nderap eriat
A Dona era caracol, ela é a dona do pr

0:11 S1: tuero pare nderap eriat, kawate aramĩ
A mulheres queriam fazer mais masca

0:18 S1: djitogotogoa, aramĩraian mbo nã
faziam masca para ela, as mulheres q

0:26 S1: djitogotogorere, tembopoe
Enquanto ela estava mascando, moía

0:30 S1: karemõ endu mẽ enduya mẽ mãa
Depois, o pilão, a mão de pilão, ela col

Figure 13: Graphical User Interface of LAWS in Android Smarthphone

using the phone microphone to create specific audio tier called **careful speech** and **Oral Translation** proposed by [13] and widely used in the linguistic annotation workflow. In the future, we would like to integrate the mobile app, LAWS, and ELAN to include the exported XML and audio files.

References

[1] Dan Abramov and Andrew Clark. 2015. Redux. http://redux.js.org Accessed on March 4th, 2025.

[2] Sanchit Aggarwal et al. 2018. Modern web-development using reactjs. *International Journal of Recent Research Aspects* 5, 1 (2018), 133–137.

[3] Piter Austin. 2006. Data and language documentation. In *Essentials of language documentation*, Jost Gippert, Nikolaus P Himmelmann, and Ulrike Mosel (Eds.). Mouton de Gruyter, Berlin, New York, 1–30.

[4] Peter K Austin and Julia Sallabank. 2011. *The Cambridge handbook of endangered languages.* Cambridge University Press, Cambridge.

[5] Shannon T Bischoff and Carmen Jany. 2018. *Insights from practices in community-based research.* De Gruyter Mouton, California.

[6] Tim Bodt. 2022. An integrated FLEx–ELAN workflow for linguistic analysis with multiple transcriptions and translations and multiple participants. *Language Documentation and Conservation* 16 (2022), 417–452.

[7] Ana Paula Brandão, Patience Epps, Susan Smythe Kung, Denny Moore, Zachary O'Hagan, and Jorge Rosés Labrada. 2023. Archiving and Language Documentation. *Cadernos de Linguística* 4, 1 (2023).

[8] Cilene Campetela, Gélsama Mara Ferreira dos Santos, Elissandra Barros da Silva, and Glauber Romling da Silva. 2017. Documentação linguística, pesquisa e ensino: revitalização no contexto indígena do norte do Amapá. *Revista Linguística* 13, 1 (2017), 151–167.

[9] Antônia Fernanda de Souza Nogueira and Rayanne Corrêa Sarraf Silva. 2021. Visualização de anotações linguísticas de arquivos de áudio e/ou vídeo: ELAN, FLEX e LINGVIEW. *Cadernos de Linguística* 2, 2 (2021), e369–e369.

[10] Lise M Dobrin and Douglas Ross. 2017. The IATH ELAN Text-Sync Tool: A simple system for mobilizing ELAN transcripts on-or off-line. *Language Documentation & Conservation* 2 (2017), 94–102.

[11] Stefan Th Gries and Andrea L Berez. 2017. Linguistic annotation in/for corpus linguistics. In *Handbook of linguistic annotation.* Springer, Netherlands, 379–409.

[12] Nikolaus Himmelmann. 2006. Language documentation: What is it and what is it good for. In *Essentials of language documentation*, Jost Gippert, Nikolaus P Himmelmann, and Ulrike Mosel (Eds.). Mouton de Gruyter, Berlin, New York, 1–30.

[13] Sarah Ruth Moeller. 2014. Review of saymore, a tool for language documentation productivity. *Language Documentation & Conservation* (2014).

[14] Denny Moore and Ana Vilacy Galucio. 2016. *2. Perspectives for the documentation of indigenous languages in Brazil.* De Gruyter Mouton, Berlin, Boston, 29–58. https://doi.org/doi:10.1515/9783110428902-002

[15] Christopher Moseley. 2010. *Atlas of the World's Languages in Danger.* UNESCO Publishing, Paris. available at: <https://unesdoc.unesco.org/ark:/48223/pf0000187026>. Accessed on March 5th, 2025..

[16] Medet Mukushev, Arman Sabyrov, Madina Sultanova, Vadim Kimmelman, and Anara Sandygulova. 2022. Towards Semi-automatic Sign Language Annotation Tool: SLAN-tool. In *Proceedings of the LREC2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources.* 159–164.

[17] Antonia Fernanda Nogueira. 2011. *Wayoro ẽmẽto: fonologia segmental e morfossintaxe verbal.* Master's thesis. University of São Paulo.

[18] Antônia Fernanda de Souza Nogueira Nogueira, Ana Vilacy Galucio, and Carla Daniele Costa. 2023. Wayoro: Documentation of language and culture. http://hdl.handle.net/2196/5004e53b-79f6-440d-81e6-266a64579366

[19] Kalinda Pride, Nicholas Tomlin, and Scott AnderBois. 2020. LingView: A Web Interface for Viewing FLEx and ELAN Files. *Language Documentation & Conservation* 14 (2020), 87–107.

[20] D Will Reiman. 2010. Basic oral language documentation. *Language Documentation & Conservation* 4 (2010), 254–268.

[21] Snehesh Shrestha, William Sentosatio, Huiashu Peng, Cornelia Fermuller, and Yiannis Aloimonos. 2023. FEVA: Fast Event Video Annotation Tool. *arXiv preprint arXiv:2301.00482* (2023).

[22] Han Sloetjes and Peter Wittenburg. 2008. Annotation by category-ELAN and ISO DCR. In *6th international Conference on Language Resources and Evaluation (LREC 2008).*

[23] B Taroco and Claudete Werner. 2007. Uma Abordagem sobre Modelos de Processos Ágeis.

[24] Stefan Tilkov and Steve Vinoski. 2010. Node. js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing* 14, 6 (2010), 80–83.

[25] Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. 2006. ELAN: a professional framework for multimodality research. In *5th International Conference on Language Resources and Evaluation (LREC 2006).* 1556–1559.

[26] Yuting Zheng and Jian-E Peng. 2022. Elan (eudico linguistic annotator). *RELC Journal* 53, 2 (2022), 469–474.

Received 18 November 2024; revised 5 February 2025; accepted 21 February 2025