An Open-source Web-based Application for Development of Resources and Technologies in Underresourced Languages

Siddharth Singh
CTRANS, Dr. Bhimrao Ambedkar University
Agra, India
siddharth.unreal@outlook.com

Shyam Ratan
CALTS, University of Hyderabad
Hyderabad, India
shyam.unral@outlook.com

Neerav Mathur
UnReaL-TecE LLP
Agra, India
neerav.unreal@outlook.com

Ritesh Kumar
UnReaL-TecE LLP
Agra, India
ritesh.unreal@outlook.com

Abstract

The paper discusses the Linguistic Field Data Management and Analysis System (LiFE), a new open-source, web-based software that systematises storage, management, annotation, analysis and sharing of linguistic data gathered from the field as well as that crawled from various sources on the web such as YouTube, Twitter, Facebook, Instagram, Blog, Newspaper, Wikipedia, etc. The app supports two broad workflows - (a) the field linguists' workflow in which data is collected directly from the speakers in the field and analysed further to produce grammatical descriptions, lexicons, educational materials and possibly language technologies; (b) the computational linguists' workflow in which data collected from the web using automated crawlers or digitised using manual or semi-automatic means, annotated for various tasks and then used for developing different kinds of language technologies.

In addition to supporting these workflows, the app provides some additional features as well - (a) it allows multiple users to collaboratively work on the same project via its granular access control and sharing option; (b) it allows the data to be exported to various formats including CSV, TSV, JSON, XLSX, LaTeX, PDF, Textgrid, RDF (different serialisation formats) etc as appropriate; (c) it allows data import from various formats viz. LIFT XML, XLSX, JSON, CSV, TSV, Textgrid, etc; (d) it allows users to start working in the app at any stage of their work by giving the option to either create a new project from scratch or derive a new project from an existing project in the app.

The app is currently available for use and testing on our server[1] and its source code has been released under AGPL license on our GitHub repository[2]. It is licensed under separate, specific conditions for commercial usage.

[1] http://life.unreal-tece.co.in/
[2] https://github.com/unrealtecellp/life

1 Introduction

Field linguists constantly need tools for collecting, storing, annotating, analysing, sharing and managing linguistic data. As field linguists gather a lot of data for lots of languages, including comparatively under-represented, lesser-known, minoritised, and endangered languages around the globe, this data has to be properly preserved, quickly and accurately analysed and processed and made available to the larger community for social good. On the other hand, there are very few publicly available and accessible datasets for developing language tools and technology for a vast array of languages around the world including most of those which have been worked upon by field linguists - the field linguists', if made available in a structured format, could help in alleviating this situation to a certain extent.

To address this dual issue of accelerating the process of collecting and processing datasets of under-resourced languages (possibly with assistance from available state-of-the-art language technologies) and developing language technologies for these under-resourced languages, a unified platform with an easily available/accessible and handy interface targeted for linguists is required. Our application, Linguistic Field Data Management and Analysis System "LiFE" aims to offer a practical intervention in the field. The app creates a structured framework for the management, analysis and sharing of primary linguistic field data. It also provides interfaces for producing the derivatives of this data such as digital and print lexicons, sketch grammars, and basic language processing tools like part-of-speech taggers and morphological analyzers and generators, automatic speech recognition systems, machine translation and others.

The software focuses on automating the different

Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023), pages 120–129

tasks as much as possible through a handy, intuitive interface for carrying out all the tasks. For instance, by giving initial input, the system gradually trains automatic techniques for inter-linear glossing of the dataset and subsequent production of sketch grammar as well as NLP tools for the language.

The app integrates popular machine learning model hubs such as HuggingFace Hub[3] and Universal Language Contribution API (ULCA)[4] as well as other popular models for individual tasks to provide automation for various tasks. This has enabled us to, for example, give access to the most recent unsupervised and transfer learning-based ASR models based on transformers (such as wav2vec 2.0 (Baevski et al., 2020), wav2vec-U (Baevski et al., 2021) and Whisper (Radford et al., 2022)). It has given field linguists direct access to the most recent state-of-the-art models available for language processing tasks. On the other hand, the app has also provided access to a no-code environment for training or fine-tuning models for new languages and new tasks using some of the most popular libraries such as HuggingFace Transformers and scikit-learn - this environment provides simple point-and-click options to train baseline models that could be quickly integrated into the field linguists' workflow. These two together have helped us provide an automated pipeline for transcription, inter-linear glossing and free translation of the dataset collected from the field.

2 Related Work

The growth of field linguistics and NLP has mostly taken place independently of one another. Thus the tools for speech and multimodal data collection, management and analysis used by linguists and the tools used for data collection and annotation by NLP practitioners are not developed to be interoperable and are generally used exclusively by the two communities.

For the storage, management and gathering of multimodal data as well as the creation of a lexicon, there are certain programmes and technologies designed specifically for field linguists (or community members engaged in fieldwork for their own language). One of the earliest pieces of software created by SIL (The Summer Institute of Linguistics), Toolbox (Robinson et al., 2007), formerly known as Shoebox[5], served as a forerunner to FLEx and was primarily designed for use by anthropologists and field linguists to input their text data and create dictionaries. FieldWorks Language Explorer (FLEx)[6] (Butler and Volkinburg, 2007) and (Manson, 2020), which is used for the gathering, management, analysis, and sharing of linguistic and cultural data, is one of the most well-liked tools in the field. A software called LexiquePro[7] (Guérin and Lacrampe, 2007) makes it simple to create and format lexicon databases for sharing with others. WeSay[8] was developed by SIL to assist native speakers and non-linguists in creating dictionaries for their own languages (Perlin, 2012) & (Albright and Hatton, 2008). There have been various attempts to create tools that are primarily used for data collection. (Vries et al., 2014) talks about the creation of an app called Woefzela[9]. It is a smartphone-based (Android Operating System) data-gathering tool that (Vries et al., 2014). It supports several sessions for data collection and can operate without an Internet connection. For the purpose of data collection quality control, it works well. In the South African data collection experiment, where nearly 800 hours of voice data were gathered from remote and rural locations, this technique is displayed. Similarly, SayMore[10] is designed to collect data for building dictionaries.

While these different tools provide adequate support for different tasks, there are some serious limitations -

- All of these are standalone desktop/mobile applications and most of these tools are not compatible with Linux systems, thereby, forcing users to use them on Windows /Mac.

- There are different tools for different tasks and it is expected from the users to learn all these different tools and transfer and manage their datasets across these different tools on their own. For example, FLEx and WeSay are mainly lexicon-development software (but FLEx also supports interlinear glossing), SayMore is a data collection tool and LexiquePro is mainly for dictionary distri-

[3] https://huggingface.co/docs/hub

[4] https://bhashini.gov.in/ulca/model/explore-models

[5] https://software.sil.org/shoebox/,https://software.sil.org/toolbox/

[6] https://software.sil.org/fieldworks/

[7] https://software.sil.org/lexiquepro/

[8] https://software.sil.org/wesay/

[9] https://sites.google.com/site/woefzela/

[10] https://software.sil.org/saymore/

bution (which gives a basic dictionary editing functionality). Moreover, generally, other tools such as ELAN[11] for video transcription, something like Audacity[12] or Praat[13] for slicing the sound recordings, etc are required (Wittenburg et al., 2006), (Thompson, 2014) and (Boersma and Van Heuven, 2001). Navigating through these different tools and software is a difficult task and has a long learning curve.

- The data formats used by these tools are generally non-standard and trying to use the data processed or produced through these tools with NLP systems is not feasible without significant processing of the dataset (which would require good programming skills).

- Sharing the data, in general, and in a format that works without these tools, more specifically, is not completely straightforward.

On the other side of the spectrum, NLP practitioners, make use of a different set of tools and applications for data management and annotation. For example, Label Studio[14] is a specific open-source, web-based application for data labelling. With a clear and simple user interface, it enables users to label a variety of data kinds, including speech, text, image, video, and time series, and export to several model formats. To create more accurate machine learning models, it can be used to prepare raw data or enhance current training data (Tkachenko et al., 2020-2022).

Similarly, an open-source platform named Shoonya[15] is being created with the goal of enhancing the digital presence of India's underrepresented languages. It allows users to annotate and classify data at scale. This is a crucial necessity to produce larger datasets for neural machine translation training on a wide range of Indian languages.

Some other open-source tools for token, span and document-level text annotation include BRAT rapid annotation tool (brat)[16] (Stenetorp et al., 2011, 2012), doccano[17] is an open-source text annotation tool. For text categorization, sequence labelling, and sequence-to-sequence applications, it offers annotation features. Users can produce labelled data for sentiment analysis, named entity recognition, text summarising, and other purposes (Nakayama et al., 2018), INCEpTION[18] (Klie et al., 2018), among several others.

As is evident, none of these tools even attempt to support data collection from the field and its management and analysis. Given this, LiFE attempts to provide the following -

- An integrated interface that caters to the needs of both the field linguists and NLP practitioners for their data management and processing workflows.

- Give field and documentary linguists access to an integrated workspace for the complete workflow from questionnaire preparation to data analysis and production of community-centred outputs such as grammars, lexicons, etc, without them having to wade through the convoluted workflow of different tools for different aspects of the same work. This interface also provides a user-friendly interface for putting their data in a structured format.

- Give field and documentary linguists access to the most advanced NLP models without the need for them to set up different NLP tools for their work. It also gives them a scope to train baseline models for underrepresented and undocumented languages using a no-code environment and use it for their own work as well as make it available for others.

- Give computational linguists access to a unified data collection and annotation app with support for AI-in-the-loop annotation.

- Give computational linguists access to data from endangered, low-resource, un(der)-represented languages in a structured way (if the community and researchers agree to render it accessible). Additionally, an interface allows using the interface to train and test the model on this data.

[11] https://archive.mpi.nl/tla/elan
[12] https://www.audacityteam.org
[13] https://www.fon.hum.uva.nl/praat
[14] https://labelstud.io/, https://github.com/heartexlabs/label-studio
[15] https://ai4bharat.iitm.ac.in/shoonya, https://github.com/AI4Bharat/Shoonya
[16] http://brat.nlplab.org, https://github.com/nlplab/brat
[17] https://doccano.herokuapp.com, https://github.com/doccano/doccano
[18] https://inception-project.github.io, https://github.com/inception-project/inception

We discuss the architecture and features of the app in more detail in the following sections.

3 LiFE Interface

LiFE has a full-fledged interlinked pipeline of four customisable modules for questionnaire creation, multimodal data labelling (speech, text, image etc) and validation, production of output based on data annotation and analysis (viz lexicon generation) and model training (Figure 1). All these four modules of LiFE are discussed in detail in the following subsections and their interrelationship is demonstrated in Figure 2.

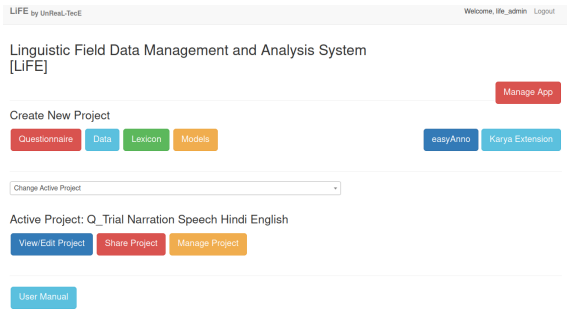

Figure 1: LiFE Interface with All Modules

3.1 Questionnaire

This module is for creating questionnaires that could be used for collecting data in the field (Figure 3). A field linguist could create a new customised questionnaire project or could derive a questionnaire project from the existing questionnaire project for a target language with the purpose of adaptation/manipulation/modification or addition of/in the existing questionnaire. This module facilitates uploading and downloading the questionnaire in multiple formats (viz JSON, CSV, XLSX, Karya JSON[19]). The questionnaire could also be downloaded in printable formats such as PDF and carried to the field for elicitation.

3.2 Data

The basic interface of this module is also the same as the questionnaire module, where a field linguist or a computational linguist could create a new customised data project or could derive a customised data project from an existing questionnaire or another data project (Figure 4). This module allows the creation of three broad kinds of projects including the following -

1. **Data Collection Projects:** The app supports data collection both from the field and through the web through two different kinds of projects.
   (a) collection of speech and text from the field.
   (b) crawling data from different sources on the web.
2. **Data Labelling projects:** The app supports the following kinds of labelling tasks -
   (a) labelling of text at both span and document level (Figure 5 (Kumar et al., 2021a)).
   (b) transcription and labelling of images.
   (c) transcription, labelling and inter-linear glossing of audio-video data.
   (d) text-to-text tasks such as translation, summarisation, etc.
3. **Data Validation Projects:** The app supports two kinds of validation projects.
   (a) giving scores to the data or annotations based on pre-defined metrics (similar to a labelling project).
   (b) arbitration and selection of the best labels among labels assigned by multiple annotators.

If the project is derived from the questionnaire or a new project is created then the module allows the upload of the data in various structured formats such as XLSX, CSV, TSV, JSON, etc. If the project is derived from another kind of data project then data and labels are copied to the new project. The module allows the data, transcriptions and annotations to be downloaded in multiple formats viz Praat Textgrid, CSV, TSV, JSON, XLSX, LaTeX, HTML and Markdown. The data could also be pushed to different platforms and repositories such as HuggingFace Hub, ULCA, GitHub, etc for public usage.

3.3 Lexicon

The data processed in different kinds of data projects is input data for this module and the field linguist could gloss transcribed/labeled data with lemma, pos category and morphological features

[19] This is the format that could be directly uploaded on the Karya Crowd-sourcing android app for data collection. https://karya.in

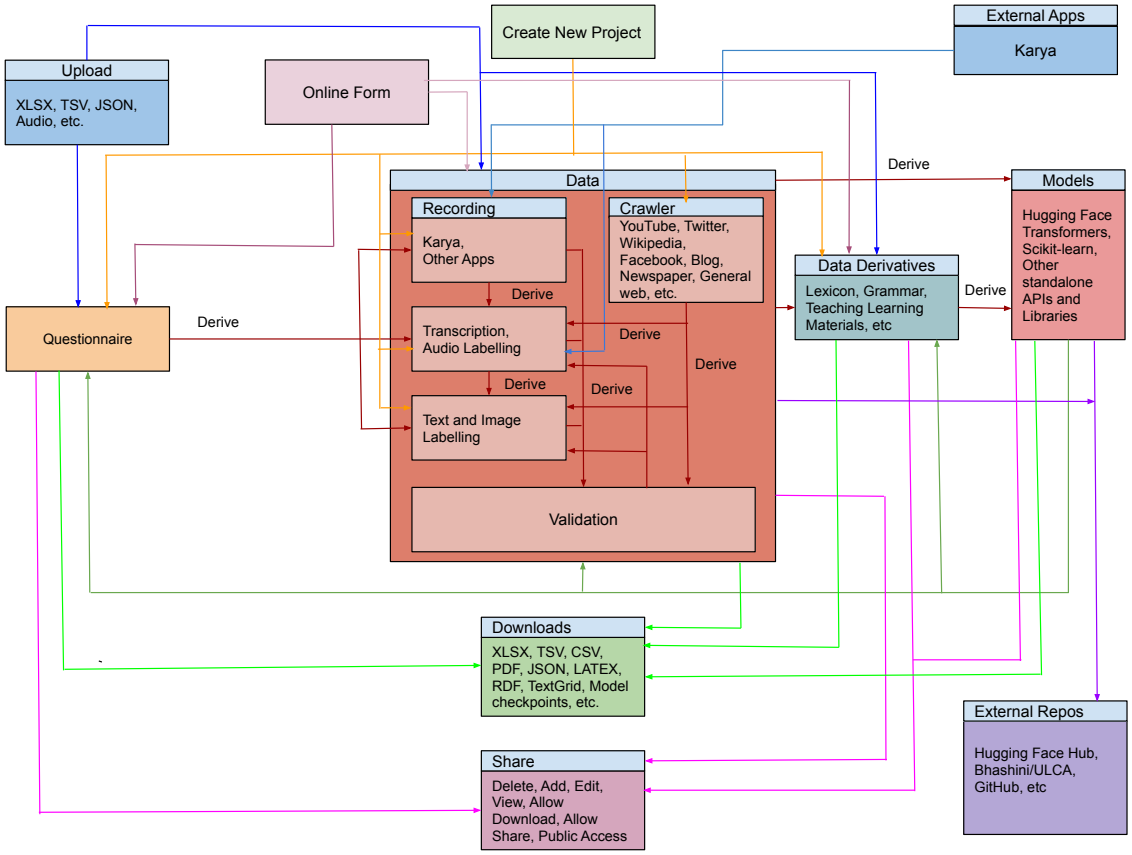

Figure 2: LiFE Workflows

to create a new lexicon (Figure 6). The information already provided during interlinear glossing or other kinds of labelling tasks (such as part-of-speech annotation) is automatically copied in the new project. A new lexicon project from scratch could also be created using the module. In that case, lexical entries could be imported from XLSX, CSV, TSV, JSON and also LIFT XML (this is the format generated by FLEx and it allows the apps' interoperability with the popular, legacy apps used by field linguists for data management and generating lexicons) formats. The data could be downloaded for further editing in multiple formats such as XLSX, CS, TSV, JSON, etc, in a publishable format for further dissemination such as Markdown or HTML or in a printable format such as PDF and LaTeX. This module also automatically generates an RDF representation of the lexicon and downloads it in different serialisation formats such as RDF/XML, Turtle, N3, etc.

3.4 Models

This module provides a no-code environment for training models for different tasks, using data from one or more projects of the same kind or different kinds of projects with similar kinds of data Figure 8. It could be derived from the projects both in the data and the lexicon modules. The models trained in this module are automatically made available across different projects in the app for immediate use. They could also be directly pushed to different platforms such as HuggingFace Hub, ULCA, GitHub, etc for public usage.

4 Supported Workflows in LiFE

LiFE does not enforce any specific workflows or pipelines. The app is currently being used by over 200 users and at least three organisations for various purposes and they all have different workflows. The app allows users to start working at any point in their data collection and analysis project. While users could define their own workflow and use different modules accordingly, the has been consciously designed to support two broad kinds of workflow -

- The Field Linguists' Workflow and
- The Computational Linguists' Workflow

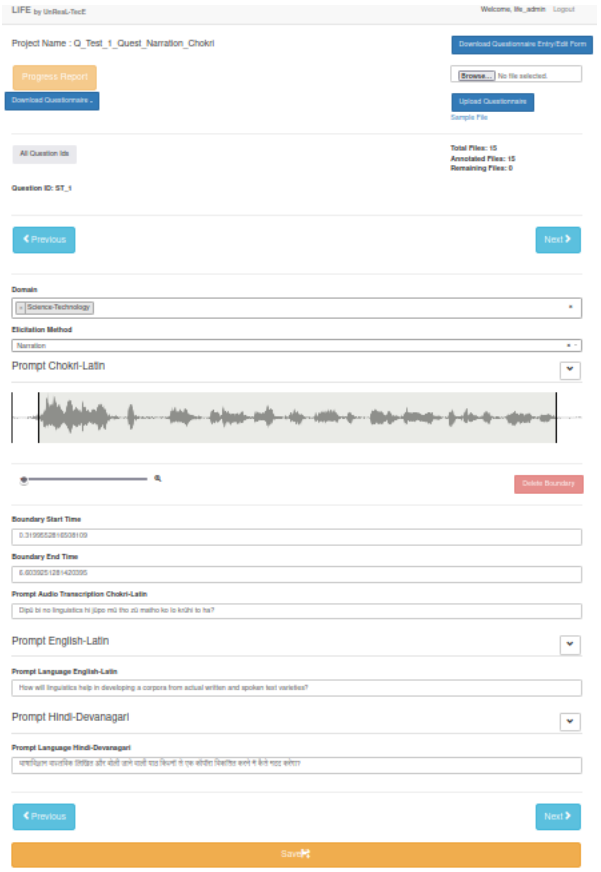

Figure 3: LiFE Questionnaire Interface

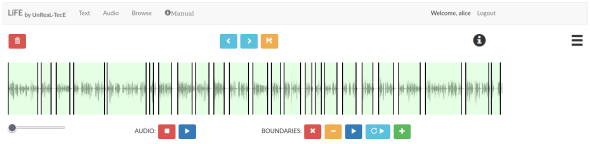

Figure 4: LiFE Data Interface

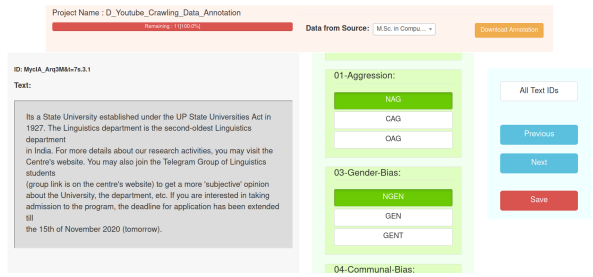

Figure 5: LiFE Text Annotation Interface

These two are discussed in the following subsections.

4.1 The Field Linguists' Workflow

A typical workflow of field linguists starts with the creation of questionnaires and other elicitation tools that could be used in the field for data collection. For this, the users have two options

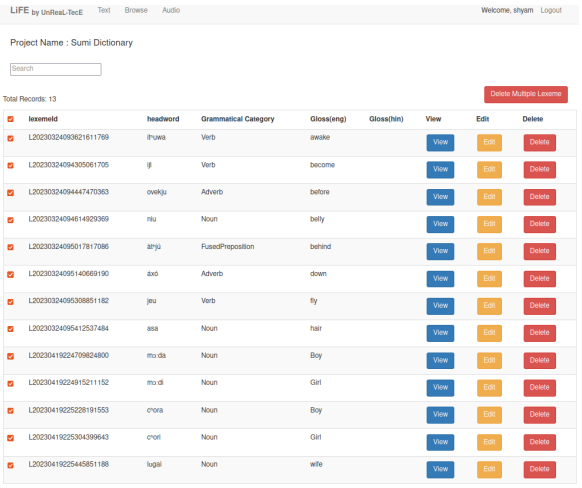

Figure 6: LiFE Lexicon Inteface

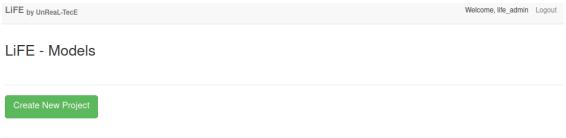

Figure 7: LiFE Models Module Form

to create questionnaires for field data collection. The first is to develop new questionnaires as per their target/goal for writing grammatical descriptions, generating lexicons, and preparing educational materials of any language. The other way is to modify and adapt some existing questionnaires to their needs. The questionnaire module in the app supports all these. After this data is collected directly from the speakers in the field using that questionnaire and imported into the data project for storage, transcription, inter-linear glossing and analysis. Once data is prepared the lexicon is generated by encoding grammatical, syntactic and semantic categories, morphological features, lexical relations, inter-linear glossing and free translation including examples. Users could also develop text and speech technologies like pos-tagger, morphological analyser and generator, automatic speech recognition (ASR), etc out of labelled data for multiple domains. Thus the app provides a simple, linear workflow for field linguists across its four modules.

The app is being used in large projects that make use of this workflow. One such project is "The Speech Datasets and Models for Indian Languages" (SpeeD-IL)[20], which is working on developing transcribed speech corpora of around 1000 hours each

[20] https://sites.google.com/view/speed-il, https://github.com/unrealtecellp/SpeeD-IL

for more than ten languages of the four major language families of India: Tibeto-Burman, Austro-Asiatic, Dravidian, and Indo-Aryan, as well as other tools and models (Kumar et al., 2022b). The project is being jointly executed by six institutions and more than twenty-five linguists are working on data collection and transcription across different languages. The other project is that of the Linguistic Data Consortium for Indian Languages (LDC-IL) [21] is an initiative by the Government of India to create all kinds of datasets across all Indian languages for technology development. The LiFE app has been recently adapted as the application for speech transcription, part-of-speech tagging and other kinds of activities in the consortium.

4.2 The Computational Linguists' Workflow

A typical workflow of computational linguists starts with data collection, generally from the web sources such as YouTube, Twitter, Facebook, Instagram, Blog, Newspaper, Wikipedia, etc. using automated crawlers A large number of such crawlers are integrated with the data module of the app and could be employed out-of-the-box for collecting multimodal data including speech, text and images. After collection, the data is annotated, transcribed, translated or processed in some way to make it suitable for training models of different kinds. A typical workflow also involves validating the data and its annotations. The data module provides support for these as well. Further, annotated data is used for developing different kinds of language technologies by training different models using the models module of the app.

Workflow like this has also been utilised in projects like "The Communal and Misogynistic Aggression in Hindi-English-Bangla-Meitei" (ComMA)[22], which was a multi-institutional project that focused on aggression identification in Hindi, English, Bangla, and Meitei ((Bhattacharya et al., 2020), (Kumar et al., 2021b) and (Kumar et al., 2022a)). Another similar project using the app is "Measuring Harm Potential of Social Media Content in India", being carried out by the Council for Strategic and Defense Research (CSDR). It aims to predict the possibility of a text (in Hindi or English) leading to some real-world harm. The annotation schema of the project is a complex mix of cross-document, single-document and span-level annotations and is handled efficiently in the app.

5 LiFE Technology Stack

The app's backend is built on the Python-based Flask[23] framework, with MongoDB (as the database) and the frontend using HTML, CSS, and Javascript. Bootsrap v3 and JQeury are used for developing the user interface in the app. Wavesurfer.js[24] is used for creating interactive, customizable waveforms which is an open-source audio visualization library. To train models for different types (audio recording and transcription, crawling and annotation) of data created in the LiFE app, Hugging Face[25] and scikit-learn[26] are used. We are also using the models hosted on the HuggingFace App to provide most of the automation facilities in the app - the app basically provides user interfaces to access these models for various tasks. The app is being developed using Agile methodology - this is ensured by keeping different modules as different Blueprints in the Flask app.

5.1 Database Architecture

Since the app contains various kinds of data and includes both structured and unstructured datasets, we are using a NoSQL database, MongoDB. It allows for storing the data entries as documents across different collections. The database of the tool contains fifteen core collections for storing different kinds of information. These are discussed below:

- **userlogin:** contains all the usernames with metadata and user profile information in the application,
- **userprojects:** contains projects that each user has developed and shared, as well as their active projects at any given point of time,
- **projectsform:** stores the forms created by users for their projects (questionnaire, data, lexicon and model) in JSON-like format. This stored information is used to render the HTML for all kinds of projects and is crucial to ensure that the interface and other properties of all projects remain completely customisable,

[21] https://www.ldcil.org/

[22] https://sites.google.com/view/comma-ctrans, https://github.com/unrealtecellp/ComMA

[23] https://flask.palletsprojects.com/en/2.0.x/

[24] https://wavesurfer-js.org/

[25] https://huggingface.co/

[26] https://scikit-learn.org/stable/

- **projects:** collection that has information about the project, its owner and project type (questionnaires, annotation, transcriptions, recordings, etc), project derivatives (other projects in the LiFE app that derive this project), project derive from (project from which this project is being derived),

- **questionnaires:** collection has a document for each prompt in the questionnaire which contains the prompt itself, a unique id, domain and elicitation information, prompt type (text, audio, image, multimedia) of the questionnaire,

- **recordings:** collection contains one document for each recorded audio and metadata of the audio (channels, sample rate, length etc),

- **crawling:** has information of data which is crawled with sources details from where the data is crawled,

- **tagsets:** has info regarding the tagset uploaded for text and image annotation, these tagsets can also be used by other projects if the user has to do a similar kind of annotation; since the app allows for using completely customised tagsets, with relatively complex structure, we have defined a structured format for uploading the tagset - this collection contains all the information provided through that structure,

- **annotation:** collection has one document for each data point to be annotated and as the same data can be annotated by multiple annotators so each annotators annotation is recorded in the same document by their username,

- **transcriptions:** collection contains the information about the transcription that has been done for each recording, speaker id, textgrid which has information about the transcription done for the audio at discourse, sentence, word, or phoneme level. The same audio could be transcribed by multiple users and the record is maintained accordingly,

- **lexemes:** collection contains information about each lexeme, with the aid of the appropriate vocabulary.

- **speakerdetails:** stores a list of metadata of all speakers,

- **sourcedetails:** is listing all the sources from where data is fetched or uploaded in the system,

- **models:** contains all the details about different models that have been trained in the app,

- **fs.files:** saves fs.chunks and the file's metadata. a file's binary portions, including pictures, videos, and audio files, are stored.

Besides these, some other collections are defined for interfacing with external apps such as Karya, interacting with external repositories like HF Hub and also for storing the app-level settings. The app stores all kinds of data and metadata in the database, without the need of storing anything in the file system.

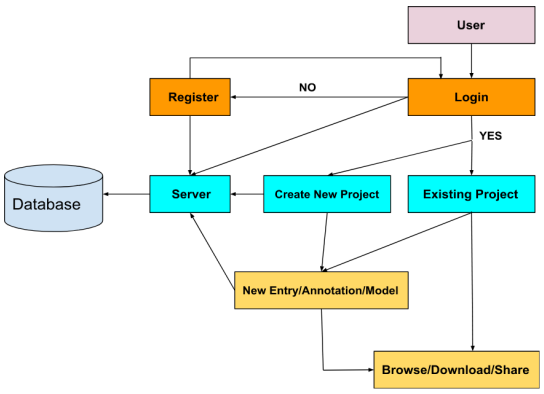

Figure 8: Model Diagram of LiFE

6 Summary

In this paper, we have presented an open-source app, `LiFE` for linguistic data management, analysis and sharing. The app intends to accelerate the development of language technologies for extremely underresourced languages by providing a link between field linguists and computational linguists. The app allows field linguists to use state-of-the-art NLP models for aiding and accelerating their work and also training baseline models for new languages and tasks in a no-code environment. At the same, it stores the data collected in the field in a structured format that could be used by computational linguists for their research. The app is

currently being actively developed and is also used by multiple teams for their research.

7 Acknowledgements

We would like to thank the Linguistic Data Consortium for Indian Languages, Central Institute of Indian Languages and Ministry of Electronics and Information Technology, Government of India for providing grants and necessary support for the development of this app. We would also like to thank Karya Inc and the Council for Strategic and Defense Research for providing the support essential for developing the app.

References

Eric Albright and John Hatton. 2008. Wesay, a tool for collaborating on dictionaries with non-linguists. *Documenting and revitalizing Austronesian languages*, 6:189 – 201.

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2021. Unsupervised speech recognition. *CoRR*, abs/2105.11084.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *CoRR*, abs/2006.11477.

Shiladitya Bhattacharya, Siddharth Singh, Ritesh Kumar, Akanksha Bansal, Akash Bhagat, Yogesh Dawer, Bornini Lahiri, and Atul Kr. Ojha. 2020. Developing a multilingual annotated corpus of misogyny and aggression. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 158–168, Marseille, France. European Language Resources Association (ELRA).

Paul Boersma and Vincent Van Heuven. 2001. Speak and unspeak with praat. *Glot Int*, 5:341–347.

Lynnika Butler and Heather Volkinburg. 2007. Review of fieldworks language explorer (flex). *Language Documentation and Conservation*, 1.

Valérie Guérin and Sébastien Lacrampe. 2007. Lexique pro. *Language Documentation and Conservation*, 1(2):293 – 300.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico.

Ritesh Kumar, Enakshi Nandi, Laishram Niranjana Devi, Shyam Ratan, Siddharth Singh, Akash Bhagat, and Yogesh Dawer. 2021a. The comma dataset v0.2: Annotating aggression and bias in multilingual social media discourse.

Ritesh Kumar, Shyam Ratan, Siddharth Singh, Enakshi Nandi, Laishram Niranjana Devi, Akash Bhagat, Yogesh Dawer, Bornini Lahiri, and Akanksha Bansal. 2021b. ComMA@ICON: Multilingual gender biased and communal language identification task at ICON-2021. In *Proceedings of the 18th International Conference on Natural Language Processing: Shared Task on Multilingual Gender Biased and Communal Language Identification*, pages 1–12, NIT Silchar. NLP Association of India (NLPAI).

Ritesh Kumar, Shyam Ratan, Siddharth Singh, Enakshi Nandi, Laishram Niranjana Devi, Akash Bhagat, Yogesh Dawer, bornini lahiri, Akanksha Bansal, and Atul Kr. Ojha. 2022a. The comma dataset v0.2: Annotating aggression and bias in multilingual social media discourse. In *Proceedings of the Language Resources and Evaluation Conference*, pages 4149–4161, Marseille, France. European Language Resources Association.

Ritesh Kumar, Siddharth Singh, Shyam Ratan, Mohit Raj, Sonal Sinha, Sumitra Mishra, Bornini Lahiri, Vivek Seshadri, Kalika Bali, and Atul Kr. Ojha. 2022b. Annotated Speech Corpus for Low Resource Indian Langauges: Awadhi, Bhojpuri, Braj and Magahi. In *Proc. 1st Workshop on Speech for Social Good (S4SG)*, pages 1–5.

Ken Manson. 2020. Fieldworks linguistic explorer (flex) training 2020 (ver 1.1 august 2020).

Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text annotation tool for human. Software available from https://github.com/doccano/doccano.

Ross Perlin. 2012. Wesay, a tool for collaborating on dictionaries with non-linguists. *Language Documentation & Conservation*, 6:181 – 186.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision.

Stuart Robinson, Greg Aumann, and Steven Bird. 2007. Managing fieldwork data with toolbox and the natural language toolkit. *Language Documentation and Conservation*, 1.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Pontus Stenetorp, Goran Topić, Sampo Pyysalo, Tomoko Ohta, Jin-Dong Kim, and Jun'ichi Tsujii. 2011. Bionlp shared task 2011: Supporting resources.

In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 112–120, Portland, Oregon, USA. Association for Computational Linguistics.

Douglas Earl Thompson. 2014. An overview of audacity. *General Music Today*, 27(3):40–43.

Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020-2022. Label Studio: Data labeling software. Open source software available from https://github.com/heartexlabs/label-studio.

Nic Vries, Marelie Davel, Jaco Badenhorst, Willem Basson, Etienne Barnard, and Alta de Waal. 2014. A smartphone-based asr data collection tool for under-resourced languages. *Speech Communication*, 56:119–131.

Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. 2006. Elan: A professional framework for multimodality research. *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*.