The Philotis Platform: Empowering Low-Resource Languages Processing

Vivian Stamou Vasileios Arampatzakis Dimitrios Karamatskos
Vasileios Sevetlidis Nicolaos Valeontis Stella Markantonatou George Pavlidis
Institute for Language and Speech Processing, Athena R.C.
{vistamou, vasilis.arampatzakis, dkaramatskos, vasiseve,
marks, gpavlid}@athenarc.gr, {nickvaleontis}@ssl-mail.com

Relevant UniDive working groups: WG3

1 Introduction

The project Philotis has developed a web-based platform[1] and a methodology, implemented as a pipeline, for recording, analyzing, and documenting living languages in real-life contexts. The platform incorporates cutting-edge multi-modal technologies (text, image, audio). The tools have been integrated into an advanced digital platform designed for language analysis and documentation, employing machine learning techniques. The creation of resources from scratch is known to be a challenging and time-consuming task due to, inter alia, the lack of documentation, the difficulty in the accessibility of materials, the need for technological support, and the lack of complete and consistent NLP pipelines (Moran and Chiarcos, 2020).

The Philotis platform meets the needs of linguists with little technical knowledge who aim to develop plain and annotated corpora from multimodal data sources.

Users of the Philotis platform follow the procedure shown in Figure 1 that supports a wide range of settings from non-standardised oral language varieties with no writing system to varieties with the privilege of morphosyntactically annotated treebanks.

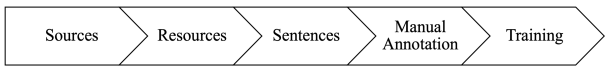

Figure 1: Pipeline Workflow.

The platform is tailored to operate in both multilingual and low-resource settings through a comprehensive adaptation in five key dimensions. First, its language-agnostic design ensures efficient functionality across linguistic variations. Second, the platform does not require a pre-specified amount or type of data, making it functional to any given settings. Third, its capability to create keyboards facilitates the handling of different writing systems, a feature of particular significance in the context of endangered languages that often lack a standardized orthography (Lin, 2021). Fourth, the tool is characterized by flexibility, allowing users to generate and choose resources customized to their needs. This adaptability extends to the training of a multi-lingual model, allowing the incorporation of resources from various languages. Thus, knowledge transfer from one language to another becomes feasible, enabling also the acquisition of an initial annotation version. Fifth, the tool aligns with the concept of 'active annotation', enabling the incremental use of text fragments and subsequent cycles of training and evaluation. This approach accelerates the generation of a gold corpus, contributing to faster and more efficient annotation processes (Anastasopoulos et al., 2018).

2 Resource development

We will present the pipeline for annotated corpus development from raw resources (recorded speech, text). The users may chose to step into (or exit from) the pipeline at some point other than its start or end, depending on the type of the available resources and on their goals.

We have already noted that Philotis facilitates the creation of new keyboards tailored to the needs of each language (Lin, 2021). The generated file can be used within the Keyman service[2].

The user starts the procedure of resource development by creating a new corpus project. Each corpus is assigned the following metadata: (i) the name of the corpus, (ii) the language, (iii) the dialect, (iv) a description, (v) the license under which it can be made available to other users (vi) the administrator, and the (vii) the contact person. Next, the 'Source' of the corpus is described, uploaded to the platform, and assigned metadata as before. As 'Source' is considered a non-empty set of raw txt/doc/pdf/audio files, and images.

In the second step, the user, through the 'Resources' component, can select texts from an avail-

[1] https://application.athenarc.gr

[2] https://keyman.com/windows/

able ‘Source’. Depending on the media type of the data (i.e., text, audio, image) two optional processes are available: (i) OCR, when the text is in an image/pdf format; the available tools are the‘tesseract-ocr’[3] and the ‘poppler-utlis’[4] packages, and (ii) speech-to-text (STT); the available tool is wav2vec2 XLS-R[5] for resources in audio format (Babu et al., 2021). The platform offers a human-in-the-loop component because the OCR and STT tools may produce output of low quality especially when language-specific training is not possible. Also, input texts may be of low quality anyway. Using the ‘File Review’ tab, results of OCR and STT can be edited manually. The corrected files are converted to the text format.

In the third step, the so-called "the ‘Sentences’ tab", allows users to choose which sentences of the raw corpus will be used for manual annotation and gold corpus creation. In a separate step, a raw corpus is used to produce word embeddings of the documented language(s), through the fastText [6] library. At this step, users can select multiple resources to create multilingual word representations or to enrich them with information from other corpora of the same or similar language variety.

2.1 The manual annotation component

The ‘Manual Annotation’ step follows, when users annotate sentences selected from the raw corpus using an in-house tool, which has adopted the functionalities of the Arborator tool (Guibon et al., 2020), as illustrated in Figure 2. The annotation schema follows the Universal Dependencies (UD) framework (de Marneffe et al., 2021); the output of the annotation is in CoNLL-U format. The annotation environment includes all the tools required to annotate a sentence. In the center of the screen is the interactive representation of the selected sentence. Each element of the representation corresponds to a form in the corresponding CoNLL-U fields. The user is also given the option to undo or redo any changes made to the graphical representation and to edit the fields (add/remove labels, i.e., PoS tags, features and dependency relations). The labels are picked from a dropdown menu that initially contains some default basic labels. The set of annotation labels can be reduced by removing labels or expanded with additional ones to meet specific language requirements (e.g., ‘obl:lmod’ or ‘obl:tmod’ for Pomak). In addition, the CoNLL-U file is validated according to the UD guidelines, ensuring that no ill-formed file will be considered for the next step.[7].

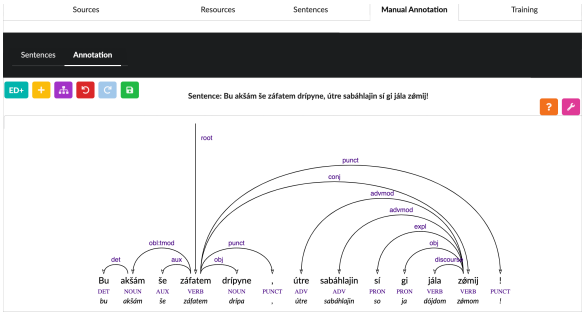

Figure 2: Corpus annotation.

2.2 Model training

The Training component comes next. It enables the users to train a model in the documented language by following a series of predefined steps. It should be noted that the steps in this case are organized according to the Stanza tool (Qi et al., 2020), as shown below:

1. Word vectorization
2. Golden corpus
3. Tokenizer training
4. Lemmatizer training
5. Part-of-Speech tagger training
6. Dependencies parser training
7. Prediction on test set
8. Model evaluation
9. Corpus prediction

It should be noted that in step 2 (‘Golden corpus’) users have the option to utilise a corpus annotated with the Philotis tools or upload other existing CoNLL-U files or any mixture of annotated corpora. Following the procedure above, a complete model can be developed. The users can select to use the model to make predictions and evaluate its performance on the selected corpus.

2.3 Backend and Frontend

At its core, the system architecture has two fundamental pillars: the Flask web framework[8] and

[3] https://github.com/tesseract-ocr/tesseract
[4] https://poppler.freedesktop.org/
[5] https://huggingface.co/facebook/wav2vec2-large-xlsr-53
[6] https://fasttext.cc/
[7] https://github.com/UniversalDependencies/tools/blob/master/validate.py
[8] https://pypi.org/project/Flask/

the Docker platform. The Flask framework provides the basis for building RESTful API endpoints that present NLP operations to users, while Docker encapsulates the entire service, including its components and environment, ensuring consistent behavior across different development environments.

The Flask service acts as the interface through which users interact with NLP functions, making requests and receiving responses through well-defined RESTful API endpoints. Leveraging the Stanza library, the Flask service orchestrates the training and updating of NLP models, ensuring that the system remains responsive and accessible even while training models.

The front end has been developed using JavaScript/PHP webpages with Axios [9] for handling asynchronous HTTP requests. To interact with the back end, the front end uses the RESTful API endpoints provided by the Flask service and integrates NLP functionalities into the user interface. PHP 8.2 has been used to develop the back end; the Laravel 9 framework has been used as an infrastructure. Access to the database is facilitated through MySQL Workbench 8.0[10].

3 Conclusions

In conclusion, the Philotis platform that is presented in this paper is a significant contribution to the field of language documentation and analysis, particularly for low-resource languages. Its user-centric design, comprehensive functionalities, and robust architecture lay a strong foundation for further advancements in linguistic research and preservation efforts. One notable feature is the platform's provision for downloading raw corpora, golden corpora, models, and predicted annotations. This capability enhances the platform's utility by enabling users to take advantage of the platform facilities at different stages of the language documentation process. In a nutshell, the Philotis platform can accommodate diverse research needs and methodologies. As this platform continues to evolve, it holds the potential to foster international collaboration among linguists in their endeavors to document and preserve linguistic diversity.

[9] https://axios-http.com/

[10] https://dev.mysql.com/downloads/workbench/

Acknowledgements

We acknowledge support of this work by the project "PHILOTIS: State-of-the-art technologies for the recording, analysis and documentation of living languages" (MIS 5047429), which is implemented under the "Action for the Support of Regional Excellence", funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

References

Antonios Anastasopoulos, Marika Lekakou, Josep Quer, Eleni Zimianiti, Justin DeBenedetto, and David Chiang. 2018. Part-of-speech tagging on an endangered language: a parallel Griko-Italian resource. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2529–2539, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. XLS-R: self-supervised cross-lingual speech representation learning at scale. *CoRR*, abs/2111.09296.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Gaël Guibon, Marine Courtin, Kim Gerdes, and Bruno Guillaume. 2020. When collaborative treebank curation meets graph grammars. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5291–5300, Marseille, France. European Language Resources Association.

Eleanor Lin. 2021. The Role of Technology in Preserving Linguistic Diversity. *Columbia Undergraduate Science Journal*.

Steven Moran and Christian Chiarcos. 2020. Linguistic Linked Open Data and Under-Resourced Languages: From Collection to Application. In *Development of Linguistic Linked Open Data Resources for Collaborative Data-Intensive Research in the Language Sciences*. The MIT Press.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.