Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Design-Based Study of an Invisible Field Tool for Endangered Languages

Mehmet Uğur Nemlioğlu

nemerko@nemerko.com

Düzce Üniversitesi

Research Article

Keywords: Adyghe, Kabardian, Mozilla Common Voice, Design-Based Research, palochka, low-resource language technology

Posted Date: September 21st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10745242/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: Competing interest reported. Financial interests: The author declares no financial interests. Non-financial interests: The author developed the tool that is the subject of this study and holds its copyright. The tool is released under a licence that requires the author's written permission for derivative works and for commercial use. The author is also the Adyghe language representative in the Mozilla Common Voice community and represents Adyghe and Kabardian at the Unicode Consortium.

Design-Based Study of an Invisible Field Tool for Endangered Languages

Author: Mehmet Uğur Nemlioğlu
Institution/Affiliation: Düzce University, Institute of Graduate Studies, Department of Caucasian Languages and Cultures, PhD Program in Circassian Language and Literature; Düzce, Türkiye
Email: nemerko@nemerko.com, ORCID: 0000-0002-1969-2356
Date: August 18, 2026, Düzce
GitHub: https://github.com/nemerko/TXTFile-CV4MCV-ady-kbd

Abstract

This study examines the design and field impact of an Excel-VBA based validation tool, developed to serve as an auxiliary utility on the Mozilla Common Voice platform for two endangered Circassian languages, Adyghe (ady) and Kabardian (kbd). Both languages were institutionalized with distinct Cyrillic standards between 1936 and 1938 (Topçu, 2006). Diasporic distribution, complex dialectal structures, and low literacy rates make crowdsourced data collection processes particularly sensitive (Leivada et al., 2019). Framed within Design-Based Research, the tool was developed through seven iterative versions (v1.0.0–v2.1.0, April 2025–August 2026). It combines sentence topic classification, palochka (U+04C0, U+04CF) validation, variant-based automatic grouping, statistics, and ady↔tr/kbd↔tr transliteration features into a single graphical user interface. The findings reveal that the workflow—which previously required specific IT knowledge—was simplified to make it accessible for non-experts, thereby positively affecting participant morale. Both communities showed resistance to Latin transliteration; this finding demonstrates that community orthography preferences in language technology design can outweigh technical functionality. The study argues that the layer of "invisible field tools" in endangered language technologies is critical for participant sustainability and data quality.

Keywords: Adyghe; Kabardian; Mozilla Common Voice; Design-Based Research; palochka; low-resource language technology

Tehlike Altındaki Diller İçin Görünmez Bir Saha Aracının Tasarım Tabanlı İncelemesi

Öz

Bu çalışma, tehlike altındaki iki Adıge (Çerkez) dili olan Adıgece (ady) ve Kabartaycayı (kbd) Mozilla Common Voice platformunda yardımcı araç olarak kullanılması amacıyla geliştirilen Excel-VBA tabanlı bir doğrulama aracının tasarımını ve saha etkisini incelemektedir. Her iki dil de 1936–1938 yıllarında ayrı Kiril standartları ile kurumsallaşmış olup (Topçu, 2006) diasporik dağılım, çoklu lehçe yapısı ve düşük okuryazarlık oranları kitle kaynaklı veri toplama süreçlerini hassaslaştırmaktadır (Leivada et al., 2019). Araç, Tasarım Tabanlı Araştırma çerçevesinde yedi yinelemeli sürüm (v1.0.0–v2.1.0, Nisan 2025–Ağustos 2026) ile geliştirilmiş; cümle konu tabanlı sınıflandırma, palochka (U+04C0, U+04CF) doğrulaması, varyant tabanlı otomatik gruplama, istatistik ve ady↔tr/kbd↔tr transliterasyonu işlevlerini tek bir grafik kullanıcı arayüzünde birleştirmiştir. Bulgular, belirli bilişim bilgisi gerektiren iş akışının uzman olmayan kişiler için erişilebilir bir basitliğe indirgendiğini ve böylece katılımcı moralini pozitif etkilediğini ortaya koymaktadır. Her iki topluluk da Latin transliterasyonuna direnç göstermiştir; bu bulgu, dil teknolojisi tasarımında topluluk yazı sistemi tercihlerinin teknik işlevselliğin önüne geçebileceğini göstermektedir. Çalışma, tehlike altındaki dil teknolojilerinde "görünmez saha araçları" katmanının katılımcı sürdürülebilirliği ve veri kalitesi için kritik olduğunu savunmaktadır.

Anahtar Sözcükler: Adıgece; Kabartayca; Mozilla Common Voice; Tasarım Tabanlı Araştırma; palochka; düşük kaynaklı dil teknolojisi

Acknowledgments

This study could not have been conducted without the foundation provided by Bülent Özden's guidance and his circ.py[1] infrastructure, which initiated the Mozilla Common Voice Adyghe and Kabardian dataset efforts. Critical contributions were also made by Dr. Murat Topçu (MCV Kabardian representative) and Dr. Saida Abregova (Adyghe philology expert) starting from the Pontoon localization stage. The work of Elizaveta Gogunokova, the coordinator who set up and ran the sub-chat group and information channel, was decisive in the growth of the community. Gratitude is expressed to the Adyghe and Kabardian communities for their voice contributions and feedback, as well as to all individuals and organizations contributing to open-source language technology ecosystems.

1. Introduction

In linguistic classification, Adyghe and Kabardian are evaluated as two distinct languages belonging to the Adyghe-Abkhaz branch of the Northwest Caucasian language family (Arkadiev & Lander, 2021). Within the internal structure of the family, there are two main groups: Abaza-Abkhaz and Adyghe (Circassian); the Adyghe group is itself divided into Adyghe and Kabardian[2] (Arkadiev, 2022). Both languages acquired their own distinct literary standards in the first half of the 20th century and were standardized on the basis of the Cyrillic alphabet between 1936 and 1938 (Zhemukhov & Aktürk, 2015). This distinct language status is institutionally recognized in the education system, publishing, and written literature (Bakov, 2023). Typologically, both languages share distinctive features such as polysynthetic morphology, ergative alignment, flexibility in lexical categories, and left-branching syntax (Arkadiev & Lander, 2021). However, the wave of diaspora that occurred after 1864 led to the dispersion of speakers outside the Caucasus—primarily to Turkiye, Syria, Jordan, Lebanon, European countries, and America—causing both languages to face severe language transmission issues in their homelands (Zaudin et al., 2024).

In this linguistic and demographic context, digital language resources play an increasingly critical role in the preservation and development of endangered languages. Kornai emphasizes that a standardized orthography is a prerequisite for collective projects such as Wikipedia and large corpora; orthographic diversity and encoding inconsistencies directly hinder the development of machine learning-assisted language technologies (Kornai, 2013). The Common Voice project is a large-scale crowdsourcing initiative launched by the Mozilla Foundation in 2017 to create such digital resources; at the time of publication, it was active in 29 languages, in the data collection phase for 38 languages, and had compiled approximately 2,500 hours of audio data from over 50,000 participants (Ardila et al., 2020).

This article is an empirical study examining the design of an Excel-VBA based validation tool developed during the process of adding Adyghe and Kabardian to Common Voice within a DBR[3] framework,

---

[1] circ.py was developed by Bülent Özden: https://github.com/HarikalarKutusu/circassian-transliterate. The transliteration tables the tool uses are published separately and independently as part of this work: Nemlioğlu (2026), https://doi.org/10.5281/zenodo.21940474 (CC BY 4.0).

[2] In this study, the terms “Adyghe” and “Kabardian” are used in the forms in which they appear as official state languages in the constitutions of the republics of the Russian Federation. Article 5 of the Constitution of the Republic of Adygea establishes Adyghe, together with Russian, as an equal state language (НПП "Гарант-Сервис", 1995). Kabardian, in turn, is listed as one of three state languages under the name Kabardian in Article 76 of the Constitution of the Kabardino-Balkar Republic (НПП "Гарант-Сервис", 1997); it is designated under the name Circassian in Article 11 of the Constitution of the Karachay-Cherkess Republic (НПП "Гарант-Сервис", 1996), and this is the same language as the Kabardian spoken in the Kabardino-Balkar Republic.

[3] DBR (Design-Based Research): This methodology, frequently used in the integration of educational sciences and technology, aims to bridge the gap between theoretical research and real classroom or field applications (Cela-Ranilla et

proposing the concept of "invisible field tools" for sustainable language technology. The main research question of the study is: Does a validation/transliteration layer wrapped in a familiar spreadsheet environment with a graphical user interface and adapted for multi-dialectal, diasporic languages improve data quality and participant efficiency in crowdsourced projects like Common Voice?

To answer this question, the article focuses on the following sub-questions:

1. How does the DBR framework operate under conditions where the developer and operator in the field are the same person?
2. What concrete functionalities does a multi-dialectal structure and a diasporic participant base necessitate in tool design?
3. Do pre- and post-intervention workflow metrics make the impact of the tool layer measurable?
4. How do community members of endangered Adyghe and Kabardian languages (Demir Y., 2022, p. 203; Moseley, 2010) react to writing system interventions?

The remainder of the article is structured as follows: The second section summarizes the historical and sociolinguistic context of the two languages. The third section reviews relevant literature—very low-resource language speech data studies, crowdsourced language documentation, Common Voice language integration practices, and the use of DBR in contemporary language learning research. The fourth section introduces the DBR methodology, the field site, and the participant structure. The fifth section details the technical design of version 2.1.0 of the Excel-VBA tool. The sixth section presents the findings. The seventh section discusses the findings, evaluates limitations, and draws implications for future web-based tool architecture.

1.1 Materials and Method

The design of the VBA application presented in this article belongs entirely to the researcher, with partial assistance from Claude AI for code writing. Furthermore, in the preparation of this article, Gemini was used for translations required to read sources in different languages, Jenni AI was used for literature research, and Claude AI was utilized for data extraction and calculation tasks to generate certain tables; all detailed content was authored by the researcher. Table contents and calculations generated by AI were specifically verified by the author. An empirical approach was adopted for studies conducted on the community.

2. Historical and Sociolinguistic Context

2.1. Position of Adyghe and Kabardian Within the Language Family

Adyghe (ady) and Kabardian (kbd)[4] are classified within the Northwest (or Ibero-) Caucasian language family as Western and Eastern Circassian languages, respectively (Arkadiev & Lander, 2021). Another important branch of the family is the Abaza-Abkhaz group; together with this branch, both Circassian languages form the "Adyghe-Abkhaz" super-group and represent its closest members linguistically (Arkadiev, 2022). Structural similarities, phonetic correspondences, and shared vocabulary between Adyghe and Kabardian demonstrate a high degree of affinity (Arkadiev & Lander, 2021). However, having developed within distinct literary traditions over a long period, these two languages have diverged in terms of alphabet, phonetics, lexicon, and orthographic rules (Zhemukhov & Aktürk, 2015).

al., 2025). Within the DBR framework, researchers design technology (e.g., language-learning tools) in real settings, test it, continually refine it using the empirical data obtained, and produce both practical solutions and theoretical principles.

[4] Both languages have been represented by the abbreviations “ady” and “kbd” since the second edition of the ISO 639 standard (Nemlioğlu M. U., 2018). This standardization allows the two languages to be processed separately on language-technology platforms. Common Voice and similar translation and localization platforms use these codes to create separate datasets for each language (Nemlioğlu & Abregova, 2025).

2.2. Language Planning Studies in the 1920–1939 Period and Dual Language Status

Prior to the language planning efforts initiated during the Soviet period, the first notable alphabet attempt for Circassian languages was Umar Bersey's Arabic-based work titled «Адыгабзэм ибукварькIэ» (Adygabzem yibukvarych'e), published in Tbilisi in 1853. This work was used for a decade in the education of Circassian children in Stavropol, Ekaterinodar, Yeysk, and Novocherkassk (Бырсыр, 2017).

With the language planning studies started in the Soviet period, the writing systems of Caucasian languages were radically altered. Between 1920 and 1939, trials for transitioning to Arabic and Latin scripts took place for Circassian languages (Zhemukhov & Aktürk, 2015); however, separate Cyrillic alphabets were established for Eastern Circassian in 1936 and Western Circassian in 1938 (Topçu, 2006).

The project for the Adyghe Cyrillic alphabet was prepared by N. F. Yakovlev and Daut Ashkhamaf. It was approved by a decision of the Adyghe Regional Committee in 1937 and came into use in 1938 (Анчек, 2022). The foundation of the modern Kabardian Cyrillic alphabet was created by Tuta Borukayev in 1935 based on the Russian alphabet, replacing the Latin-based version initiated by B. Khuranov in 1923. The separate Cyrillic alphabet for Adyghe was finalized in 1938, while revisions for Kabardian continued until 1939 (Кумахов, 2013, pp. 7-8).

These alphabets currently form the basis of the official literary standards for both languages. This historical decision represents the most distinct turning point in institutionalizing the two languages as separate literary standards. Although initiatives to create a "single Circassian literary language" were discussed in subsequent years, maintaining two separate standards persisted in educational and publishing practices (Bakov, 2023). The necessity of re-evaluating dialect classifications was excluded from the scope of this study; however, in tool design, variant-based file organization was adopted to align with the variant structure established for these languages in the MCV project.

2.3. Post-1864 Diaspora Period and Dialect Evolution

Following the Russian-Circassian War, which ended in 1864, a major portion of Circassian communities was exiled to Ottoman territories (Demir B., 2025). This forced migration led to geographically-driven shifts in dialectal structures. Circassian communities remaining in the Caucasus and those dispersed across Türkiye, Syria, Jordan, Lebanon, Egypt, Europe, and America (Nemlioğlu M. U., 2016) maintained their languages under varying conditions, and dialectal divergence became more pronounced in this process. The same diaspora dynamic is also observed for the Abaza community, which belongs to the same language family (Karmova et al., 2023). Although dialect classification was historically made on the basis of Shapsug, Bjedug, Temirgoy, Abadzeh (West Circassian/Adyghe dialects) and Kabardian, Besleney, post-diaspora interconnected structures and population distributions in autonomous republics in the Caucasus complicated these classifications.[5] For instance, Kumakhov divides Kabardian, which was previously a single dialect, into the following dialects: Baksan, Beslaneevsky and Kuban. There are also Malkinsky, Malo-Kabardinsky, Mozdoksky and other sub-dialects (Кумахов, 2013, p. 8).

3. Related Work

3.1. Speech Data Studies in Very Low-Resource Languages

[5] In individual interviews, it was observed that some individuals of Shapsug descent living in Amman, Jordan spoke Kabardian rather than the Shapsug dialect. Similarly, while some families of Abzakh descent living in Düzce speak the Shapsug dialect, some lineages known to have lived in the pre-1864 Shapsug region identify themselves as belonging to the Abzakh clan.

Constructing speech data for very low-resource languages has long been a significant research area in language technologies. Gurtueva and Kamensky built an acoustic-phonetic database with native Kabardian-Circassian speakers, obtaining records for 10 speakers, 4,610 words, and approximately 20 minutes of audio per speaker (Gurtueva & Kamensky, 2024). Although limited in volume, such studies provide valuable evidence for model development and typological investigation. In bilingual reading studies, participants have exhibited varying dialect distributions that directly affect corpus composition; for example, in an eye-tracking study conducted by Zdorova et al., participants were distributed across Bjedug (n=22), Kabardian (n=12), Temirgoy (n=11), and Abadzeh (n=5) dialects, leading researchers to emphasize that Kabardian should be evaluated as a separate East Circassian language (Zdorova et al., 2023). This finding demonstrates the necessity of a dialect-conscious approach in corpus design.

Common Voice is the most comprehensive crowdsourced data source for low-resource languages. Nevertheless, even a medium-resource language like Turkish had a limited dataset of 68 hours and 1,228 speakers in the January 2022 release (Çöltekin et al., 2022, p. 14); this highlights how challenging it is to build initial data volumes for Adyghe and Kabardian. Godard et al. demonstrated that systematic data collection from very low-resource languages is achievable (Godard et al., 2018). Tyers and Meyer showed that speech recognition experiments can be conducted for "underserved languages" in Common Voice using even around one hour of data (Tyers & Meyer, 2021).

As a result of efforts initiated in September 2024, Adyghe and Kabardian were included in Mozilla Common Voice language lists. Upon basic completion of translations on the Pontoon tool, initial community members began audio recordings for ASR in these languages by late 2024.

The evolution of multilingual corpus releases following the launch of Common Voice in June 2017 serves as an important benchmark to frame the platform representation of endangered languages. Table 1 presents the language count, total hours, and validated hours for the initial release, the intermediate release where Adyghe and Kabardian were added (Corpus 23.0), and the current release (Corpus 26.0):

Table 1. Mozilla Common Voice Corpus Releases — Overview Metrics[6][7]

Status	Version	Date	Number of Languages	Total Hours	Valid Hours
Initial (first multilingual release)	Corpus 1	25.02.2019	**19**	1,368	1,096
ady + kbd added (during expansion of the language list from 137 to 286)	Corpus 23.0	05.09.2025	**286**	35,921	24,600
Current	Corpus 26.0	12.06.2026	**294**	42,388	28,893

3.2. Dialect-Conscious, Crowdsourced Language Documentation and Community Leadership

Multi-dialectal structures mandate dialect-conscious corpus design. The Awal platform, developed by Öktem and Boudichat for Tamazight, collects contributions through a community-oriented approach to increase Tamazight's digital representation (Öktem & Boudichat, 2025). This approach defines the

[6]The dates in the table are cutoff dates, not publication dates. According to the explicit notice in the common-voice/cv-dataset repository, the dates in the JSON file names represent data cutoff dates. For Corpus 23.0, the cutoff was 05.09.2025 while the actual release was 17.09.2025; for Corpus 26.0, the cutoff was 12.06.2026 while the actual release was 17.06.2026.

[7] The figures were compiled on 26.07.2026 from the relevant ".json" and ".md" files published by Mozilla CV at https://github.com/common-voice/cv-dataset/.

bridging role of language representatives between community leadership and technical coordination—a key actor role in this study as well. Including low-resource languages in digital environments contributes to a more equitable language technology ecosystem (Pakray et al., 2025). Mbogho et al. report that adding languages to Common Voice occurs via the Pontoon platform, requiring approximately 1,000 technical strings to be translated per language, with new terms coined if necessary (Mbogho et al., 2025). The translation set collected to date includes thousands of phrases and terms; translation completion rates vary for Adyghe and Kabardian[8], and due to data ambiguities caused by non-community contributions, the exact amount of fully approved translations for Kabardian has not been clarified[9]. This process requires language representatives to collaborate with experts and linguists possessing both technical and linguistic competencies.

Crowdsourced language documentation is another important approach in this field. The Web 2.0 ecosystem offers new possibilities for crowdsourced language documentation approaches but introduces scalability challenges (Austin, 2023, p. 3). Bettinson highlights the potential of participant-centered language documentation to produce scalable natural language corpora (Bettinson, 2015). The tool examined in this study directly addresses the problem of shortening processing cycles while maintaining data quality alongside participant engagement.

A two-tiered quality control model utilizing combined human and automated inspection was exemplified in Lange and Aznar's RefCo Checker study, where manually entered data in a digital environment is first evaluated by automated tools and subsequently reviewed by humans (Lange & Aznar, 2022). The tool presented here implements a similar two-tiered structure; however, under conditions where the operator acts as both developer and practitioner, this structure enables a tighter control loop.

3.3. Transliteration Tools and Open-Source Infrastructure

Transliteration tools for multi-script languages are fundamental language technology components. Qian et al. developed a Python toolkit for universal transliteration (Qian et al., 2010). The approach in this study forms part of an effort to make a similar transliteration infrastructure accessible to field users.

Turkish natural language processing tools served as the architectural inspiration for circ.py, which marked the starting point of this work. Turkish text cleaning practices—such as spelling corrections, character normalization, and encoding fixes—were adapted for Adyghe and Kabardian text data. Developed by Bülent Özden, circ.py is an application executing statistical functions like character/word counts as well as spelling validation, extended here for Adyghe and Kabardian. Nevertheless, transitioning from Cyrillic to Latin or different transliteration systems adds an extra layer of complexity for these languages.

3.4. Design-Based Research Framework and CALL Practices

Design-Based Research is an iterative research approach in educational research where intervention development and implementation processes are intertwined. Reeves and McKenney argue that DBR yields three main contributions: (a) field intervention design and refinement, (b) development of theoretical knowledge, and (c) professional development of the researcher (Reeves & McKenney, 2013). Core characteristics of this approach include being "pragmatic, grounded, interactive, iterative,

[8] Source: https://pontoon.mozilla.org/ady. This data was compiled based on information obtained during a visit on 2026-07-30.

[9]Example: string #312523 (“Sentences are collected for people to read aloud.”) was approved for the Kabardian (kbd) field with a translation written in Russian, “Собираются предложения для чтения вслух.” This and similar entries—texts actually written in Russian rather than Kabardian but marked as approved—make it difficult to determine the true translation-completion percentage for Kabardian: https://pontoon.mozilla.org/kbd/common-voice/web/locales/common-voice/en/pages/about.ftl/?string=312523 (visited on 2026-08-07.)

flexible, integrative, and contextual" (Wang & Hannafin, 2005). These characteristics align with the evolution of the tool developed in this study across seven iterative versions (v1.0.0–v2.1.0).

In Computer-Assisted Language Learning (CALL), DBR has gained increasing acceptance as a research methodology in recent years (Reeves & McKenney, 2013). Gaffas noted in a DBR-based study on using social media platforms for language learning that the approach allows for iterative testing and design refinement (Gaffas, 2024). Levy argues that for DBR to normalize within CALL, tools must transition from a research focus to practitioner integration, reflecting how these tools emerge (Levy, 2013). The Excel-VBA application developed in this study embodies this transition by moving the console structure of circ.py—which required "deep" technical knowledge—into an ordinary and accessible environment.

4. Method: Design-Based Research Framework

4.1. Research Design

This study was conducted within a Design-Based Research framework (Kılınç & Özkök, 2025). The impact of the Excel-VBA tool—developed as an intervention across seven iterative field versions (v1.0.0, v1.0.1, v1.1.0, v1.1.1, v1.2.0, v2.0.0, v2.1.0)—on both data quality and participant sustainability was examined. Combining the roles of researcher, developer, and operator in one person represents a concrete instance of DBR's "researcher in the field" principle.

4.2. Field Site and Participant Structure

The field site for this study is a multilingual crowdsourced project initiated in late 2024 via Pontoon to add Adyghe and Kabardian to Common Voice. The division of roles among language representatives is as follows: Adyghe (ady) representative [the author]; Kabardian (kbd) representative Dr. Murat Topçu[10]; advisor for Adyghe Dr. Saida Abregova[11]; advisor for MCV Bülent Özden[12]. The first Telegram group was established on October 14, 2024, for administrative purposes, alongside the initiation of Pontoon translations for both languages. On the same date, two official Telegram groups were created, and volunteers were invited for voice contributions and recordings. However, actual voice recordings began after Pontoon translations crossed the 35% threshold, enabling the languages to be listed on the MCV interface.

The community structure grew organically. The management group initially comprised four individuals, later expanding to six, though active participants remained at five. In January 2025, a community member joined the management team as a coordinator and set up a sub-chat group and an information channel. The group started with a Kabardian focus and still has one, but from the first days it also included people who work on Adyghe. The chat group reached 734 members and the channel reached 326 members. Additionally, a dedicated task group (26 members) was formed for the MCV Spontaneous Speech initiative, and a separate working group (25 members) was organized by the researcher for Adyghe sentence preparation and special tasks. A small WhatsApp group of journalists and broadcasters living in Maykop also worked autonomously to meet the Spontaneous Speech target duration before joining Common Voice recordings to continue contributing.

Table 2. Community Structure *(Snapshot values as of July 31, 2026; members continuously join or leave groups)*

Group	Function	Members

[10] Faculty member, Department of Russian Translation and Interpretation, Okan University

[11] Faculty member, Division of Circassian Language and Literature, Department of Caucasian Languages and Cultures, Düzce University

[12] Senior Computer Engineer, member of the Mozilla Common Voice Software Development group

CV-Circassian-Team (EN/TR/RU)	Management, coordination, and decision-making	6 (5 active)
CV-Circassian-Global (EN/ru/ady/kbd)	Participants from all countries	107
CV-Çerkesçe	Participants from Turkey	97
MCV-Адыгабзэ (ady)	Adyghe-specific tasks and sentence production	25
MCV:SS-Circassian	Spontaneous Speech special task	27
ДЫЗЭКЪУЭТМЭ, ДЫЛЪЭЩЩ 🤝	Kabardian-focused, mixed	724
Maykop, WhatsApp Group	Local journalists/broadcasters (SS → CV)	Undocumented
NOTE:	Arithmetic sum of the 6 measured groups is 986; unique members: 863; duplicate entries: 123. Measurements were conducted using the Telegram API and the WTelegramClient NuGet library.	

This setup displays a "core-periphery" model common in crowdsourced participation (Crowston & Shamshurin, 2017): a small core coordination group (2 active managers, 1 technical advisor + linguists), a middle layer of task-focused working groups, and a periphery of general participants. Members of these groups overlap significantly and may be active across multiple channels. This overlap demonstrates the scalability of the language representative role and the distributed nature of community leadership.

4.3. Pre-Intervention Observation and Problem Identification

In the default Common Voice model, sentences are collected through open contribution: anyone can write a sentence and donate it under CC0, the sentence enters the reading list after two speakers of that language approve it, and it is removed after two rejections. This model works for languages such as Turkish, Russian or English, because it assumes a large pool of speakers who can also read and write the language. Adyghe and Kabardian have a limited number of speakers, and the number of people who are literate in them is much smaller. Two contributors who share the same misconception may well approve an incorrect sentence, and writing a Latin character in place of the palochka is the most common example. For this reason the community chose to build a review path that works before upload; the workflow described below is that path.

Prior to tool development, the sentence collection-validation-upload cycle for Adyghe proceeded as follows:

1. Sample sentences produced by community members were compiled into shared Google Sheets tables.
2. Once sentences reached a target volume (at least 500+ items), they were submitted to an expert linguist for review. Since only a single linguist conducted reviews, this phase took between 1 and 7 days depending on workload and volume.
3. Orthographic and variant checks (palochka accuracy, script consistency) were completed in 1–5 days by the ady representative (the researcher) and the kbd representative (subject to other work commitments).

4. Validation was executed via command line using circ.py, and output files were manually copied and pasted in batches into the Common Voice portal.
5. Post-upload, two language experts conducted final error checks on the MCV system before approval. If an overlooked issue was identified, the sentence was reported or rejected.

Several critical issues were observed in this workflow:

- Sentences for each variant had to be manually collected in separate planned spreadsheets.
- Grouping sentences by domain required preparing separate files for each domain and validating them individually with circ.py.
- Participant attrition: contributors who prepared sentences and recorded audio became reluctant to continue for two separate reasons. First, it took a long time for the sentences they had prepared to reach the platform, so a contributor could not see their own work in the system. Second, when the reading queue ran out, a contributor who wanted to record found nothing to contribute to. Both were tied to the same factor: the delay between preparation and upload.
- Data quality issues: Community members frequently made digital orthography errors: double spaces, incorrect spacing before commas, improper quotation marks, and substituting or confusing Cyrillic Letter Palochka “I” (U+04C0) and Small Palochka “I” (U+04CF) with lookalike characters such as Latin capital "I" (U+0049), lowercase "l" (U+006C), or "a" (Latin: U+0061, Cyrillic: U+0430).
- Character encoding issues: Historically, standard Russian keyboard layouts were widely used for Adyghe, which lacked encodings for Cyrillic Palochka “I” (U+04C0) and Small Palochka “I” (U+04CF). Consequently, the Windows ady & kbd keyboard layout[13] developed under the author's Master's thesis (currently v1.2) was recommended and adopted by the community. Nevertheless, sentences copied from open sources occasionally contained simulated palochka characters using Latin script.

4.4. Intervention: Development of the Excel-VBA Tool

In light of these observations, an Excel-VBA tool was initiated in April 2025. The rationale for developing the tool was three-tiered:

1. Technical layer: Transitioning circ.py's complex console parameters and error-prone structure to a user-friendly form interface.
2. Workflow layer: Automatically grouping sentences by domains and language variants, reducing tasks that manually took hours down to minutes.
3. Sustainability layer: Eliminating the requirement for external Python installations and running circ.py as a subprocess, turning the utility into an "invisible field tool" that any Excel user could open and execute.

Development progressed through seven iterative releases, reaching v2.0.0 in July 2026 and v2.1.0 in August 2026. The tool was in actual use in the field from its first version onward, and development ran alongside live use; therefore each change in a version responds to friction met during real operation rather than to a hypothetical need. The version history is summarised in Table 3.

Table 3. Version History of the Excel-VBA Tool

[13] The keyboard layout is distributed at https://www.nemerko.org/adyghe-circassian-keyboard-for-windows-7-11/

Version	Date	Key Change
v1.0.0	2025-04-30	.xlsx → .xlsm conversion, VBA form added
v1.0.1	2025-07-21	circ.py pausing: file-to-file flow
v1.1.0	2025-08-15	Improved error handling and user warnings
v1.1.1	2025-09-09	Minor bug fixes
v1.2.0	2025-11-09	Automatic grouping by variant added
v2.0.0	2026-07-19	circ.py's entire validation and transliteration logic transferred to five VBA modules; external Python, circ.py, and settings.ini dependencies removed; 12-code domain LUT system and two-tab form added; platform support clarified and limited to Windows and Excel 2021/365.
v2.1.0	2026-08-15	The three dialectal letters (гь, кь, кIь) added to the mapping tables of both languages; three inaccuracies in the Adyghe table corrected

This version evolution demonstrates DBR's "iterative" and "flexible" qualities; the first six releases directly addressed field observations, while v2.1.0 came from the checks made for publication.

The tool was engineered from inception with domain-based grouping capability, eliminating the need to maintain distinct sentence lists per domain. Sentences compiled on a single sheet are automatically grouped into files based on domain codes specified in the Domain column.

Prior to v1.2.0, community members drafted all sentences in a single variant table. The burden of maintaining separate sheets for each variant resulted in low adoption of variant-specific sentence generation. Because all sentences were aggregated into one table, they defaulted to global Cyrillic tags (ady-Cyrl and kbd-Cyrl). This caused Russian loanwords and terms in the corpus to trigger frustration among diaspora participants. Similarly, transcriptions of spontaneous speech recorded by Adyghes residing in Türkiye proved difficult or error-prone for evaluators in the Republic of Adygea. This underscored the necessity of proper variant tagging. Thus, v1.2.0 introduced a variant column, enabling automated grouping based on variant tags.

With the release of v2.0.0, Python dependencies were removed, eliminating setup hurdles like installing Python or circassian-translate. This made the tool fully self-contained, enabling community sentence contributors to group and upload validated sentences directly to MCV following linguistic review.

The current version, v2.1.0, did not come from the field. It came from the preparation of this study for publication. During the review conducted to publish the mapping tables as an independent dataset, three dialectal letters that were not previously defined were added to the tables, and three errors in the Adyghe table were corrected (see 5.4 and 7.1).

4.5. Data Sources and Ethical Considerations

Findings in this study were collected from diverse data sources: Excel-VBA tool source code, VBA modules (v2.1.0), and README documentation; Telegram group structures and member metrics; Common Voice platform statistics (recorded hours, speakers, validation rates for ady and kbd); community feedback on transliteration (indirect observation); direct observation during MCV and MCV-SS sentence/audio verification; and Pontoon translation statistics.

This study reports findings from a crowdsourced initiative based on open, voluntary participation. Participants contributed under Mozilla Common Voice's standard terms of use. The tool does not collect user data and operates strictly locally on sentence files. Real group names are cited in Table 2 as these public channels are open to all interested contributors.

5. Excel-VBA Validation Tool (v2.0.0–v2.1.0)

5.1. Architectural Overview

As of v2.0.0, the TXT File Creator & Validator tool consists of two primary components: The Excel workbook (v2.3.0), which labels and groups sentences by variant and domain using a Look-Up Table (LUT)[14] sheet containing MCV domain tag codes, and the VBA application (form and macros, v2.0.0). The key architectural decision in v2.0.0 was eliminating external Python dependencies by porting all validation, statistics, and transliteration logic from circ.py directly into five VBA modules (mdlCircConst, mdlCircConstAdy, mdlCircConstKbd, mdlCircLib, mdlCircMain). Detailed architecture descriptions are provided in Appendix A. The interface of the tool and the file organisation it produces are shown in Figs. 1-3 in Appendix B.

In versions v1.0.0–v1.2.0, circ.py was invoked as an external subprocess via the Python interpreter—a structure that posed accessibility barriers for non-technical community members. In v2.0.0, this dependency was removed (see Appendix A). This integration relies on redistributing Bülent Özden's circ.py logic with explicit permission.

5.2. Palochka Validation

One of the most distinctive features of Circassian orthographies is the palochka character (I, U+04C0 uppercase; I, U+04CF lowercase), a character uniquely represented in the Cyrillic alphabet (Nemlioğlu M. U., 2018). Phonetically, the palochka represents the glottal stop [ʔ] on its own, while in letter combinations it marks ejective consonant modification (Höhlig, 2021). Because standard Russian keyboards omit this letter, users frequently substitute Latin "1", "l", or "I" (Nemlioğlu & Abregova, 2025, p. 7). This degrades corpus quality and introduces noise into datasets intended for AI model training (Nemlioğlu & Abregova, 2025, p. 7). When running in validation mode, the Excel-VBA tool executes palochka verification, ensuring non-standard character substitutions are automatically normalized. This validation complements the Adyghe keyboard layout previously developed for Windows and macOS by the researcher: while the keyboard handles input at the user level, the validation tool acts as a final platform-level gatekeeper.

5.3. Variant-Based Automatic Grouping and Transliteration

The Common Voice platform supports multiple variants per language for Adyghe and Kabardian (Mozilla Common Voice, 2021). The variant tables used for both languages are shown in Tables 4 and 5.

Table 4. Adyghe (ady) Variant Table

Language Variant	BCP-47 Tag	Script	Short Description

[14] LUT (Look-Up Table): Multilingual domain code table

Adyghe/West Circassian (Cyrillic)	ady-Cyrl	Cyrillic	Indicated for all literate speakers.
Adyghe/West Circassian (Cyrillic, Russia)	ady-RU	Cyrillic	Specified for literate speakers in Russia.
Adyghe/West Circassian (Cyrillic, Turkey)	ady-Cyrl-TR	Cyrillic	Specified for literate speakers in Turkey.
Adyghe/West Circassian (Latin, Turkey, transliteration)	ady-Latn-TR-t-ady-cyrl-tr	Turkish transliteration	Specifies for non-literate speakers in Turkey.
Adyghe/West Circassian (Cyrillic, Jordan)	ady-Cyrl-JOR	Cyrillic	Specified for literate speakers in Jordan.
Adyghe/West Circassian (Cyrillic, Syria)	ady-Cyrl-SY	Cyrillic	Specified for literate speakers in Syria.

Table 5. Kabardian (kbd) Variant Table

Language Variant	BCP-47 Tag	Script	Short Description
Kabardian/East Circassian (Cyrillic)	kbd-Cyrl	Cyrillic	Indicated for all literate speakers.
Kabardian/East Circassian (Cyrillic, Russia)	kbd-RU	Cyrillic	Specified for literate speakers in Russia.
Kabardian/East Circassian (Cyrillic, Turkey)	kbd-Cyrl-TR	Cyrillic	Specified for literate speakers in Turkey.
Kabardian/East Circassian (Latin, Turkey, transliteration)	kbd-Latn-TR-t-kbd-cyrl-tr	Turkish transliteration	Specified for non-literate speakers in Turkey.
Kabardian/East Circassian (Cyrillic, Jordan)	kbd-Cyrl-JOR	Cyrillic	Specified for literate speakers in Jordan.
Kabardian/East Circassian (Cyrillic, Syria)	kbd-Cyrl-SY	Cyrillic	Specified for literate speakers in Syria.

The developed tool automatically groups sentences according to the variant column in the Excel sheet, producing discrete output files for each variant. The file types and the naming scheme are given in Table 6.

Table 6. Output File Formats

File Type	Format	Naming Scheme
Main grouping file	.txt	yyyy-mm-dd-[BCP47]-{domain code(s)}.txt
Valid sentences	.tsv	yyyy-mm-dd-[BCP47]-{domain code(s)}_val.tsv
Invalid sentences	.tsv	yyyy-mm-dd-[BCP47]-{domain code(s)}_inval.tsv

Character count	.tsv	yyyy-mm-dd-[BCP47]-{domain code(s)}_chars.tsv
Word count	.tsv	yyyy-mm-dd-[BCP47]-{domain code(s)}_words.tsv
Transliteration file	.txt	yyyy-mm-dd-[BCP47]-{domain code(s)}_latin.txt

Because filenames encode date and variant metadata, traceability across processing periods is maintained.

TXT File Creator & Validator tool integrates ady ↔ tr and kbd ↔ tr transliteration tables developed by the researcher, incorporated into circ.py as functions by Bülent Özden. These tables offer reversible Cyrillic-to-Latin conversions based on phonetic closeness between Circassian and Turkish phonemes, designed primarily to facilitate participation among diaspora speakers in Türkiye who lack Cyrillic literacy.

5.4. Key Features of v2.0.0 and v2.1.0

While preserving v1.2.0's capabilities, v2.0.0 represents a significant architectural leap. Core advancements include:

- Two-tab interface: "Create TXT" and "Validate/Transliterate TXT Files" tabs separate operational workflows so users can validate files without losing context. Action button labels dynamically update based on language selection (e.g., "validate ady TXT files").
- Three output modes: validate, stats, and trans modes produce dedicated files with suffixes (_val, _inval, _chars, _words, _latn).
- Platform requirements: Windows 10/11 and Excel 2021/365 requirements are explicitly stated; Excel for Mac and Excel Online are unsupported. (Excel 2016 compatibility is expected but unverified).
- Byte-level verification: as of v2.0.0, the VBA engine produced exactly the same output as the original circ.py (2026-07-19). This match ended on purpose with v2.1.0: the corrections described below, and the dialectal letters that were added, are not present in the source script.
- v2.1.0 does not change the architecture of v2.0.0. It only changes the mapping tables. The defects that made this release necessary were not found in the field. They were found during the checks made while this study was prepared for publication (see 7.1 and 7.2). There are four changes. First, three dialectal letters (гь, кь, кІь) were added to the mapping tables of both languages; because these letters were not previously defined, a rule that deletes trailing ь characters at the end of transliteration was triggered, causing these letters to silently disappear if they were used in the text. Second, in the Adyghe table the uppercase palochka was mapped to an empty string, so a word-initial I lost its glottal mark and Іоф came out as of instead of 'of. Third, in the same table the Latin form of кІо ended with a Cyrillic о (U+043E). Fourth, a rule that could never match was removed; this last change does not affect the output.
- The tool itself is also archived as versioned, citable software (Nemlioğlu M. U., TXT File Creator & Validator for MCV (ady, kbd) (Version 2.1.0) [Computer software], 2026). With this release the tool also gained something else. The transliteration tables that the tool carries inside it are now published separately, as a citable data set. The tables can therefore be read, checked and reused without the tool. Versions v2.0.0 and v2.1.0 are still new, and how they perform in the field will be seen in the coming period.

6. Findings

6.1. Workflow Time Metrics

Table 7. Pre- and Post-Intervention Workflow Durations

Stage	Before	After
Sentence preparation ady/kbd (team capacity)	~700 /~1,800 sentences/month	~700 /~1,800 sentences/month
Linguist review (ady)	3-7 days	2-5 days
Orthography/variant check	1-3 days	1 day
Grouping + validation	Multiple days	~20-50 minutes[15] / 1,000 sentences
Upload + approval	1.5-4 hours/1,000 sentences due to CV standard upload limits	1-3 hours/1,000 sentences due to CV standard upload limits

As Table 7 shows, while sentence generation capacity remains the primary bottleneck (~700 / ~1,800 sentences per month per team), the time required for sentences to reach the platform dropped substantially. This compressed the data production cycle from days to hours and reduced the loss of motivation it caused. Because fully controlled timing was not possible under field conditions, these values are given as approximations (see 7.5). Analyzing barriers to community tech adoption confirms a direct link between literacy levels and tool usability (Öktem & Boudichat, 2025, p. 12).

6.2. Data Quality Metrics

Table 8 presents the current platform status for ady and kbd, and Table 9 presents the release-by-release change from Corpus 23.0 to Corpus 26.0.

Table 8. Common Voice Platform Statistics
(As of 2026-07-29; Source: commonvoice.mozilla.org/ady/languages)

Language	Total Hours	Speakers	Validation Rate	Total Sentences
Adyghe (ady)	95	168	84%	12602
Kabardian (kbd)	304	382	90%	31824

Table 9. Common Voice — Adyghe and Kabardian Release Comparison
Source: https://github.com/common-voice/cv-dataset, (datasets/scripted-speech, full non-delta releases)

Adyghe (ady)						
Metric	**CV 23.0-2025-09-05**	**CV 24.0-2025-12-05**	**CV 25.0-2026-03-09**	**CV 26.0-2026-06-12**	**Change (23.0→26.0)**	**Increase %**
Total clips	34,241	45,460	60,343	63,508	29,267	85.5%

[15] The "~20–50 minutes" value is an estimated duration range observed across multiple sessions by the developer-operator. This timeframe varies based on subgroup batch size (500–2,000 sentences), concurrent tasks, and computer performance. Strict chronometric measurements were not conducted; data reflects empirical reporting.

Validated clips	24,954	40,812	48,689	53,002	28,048	112.4%
Rejected clips	883	1,524	2,585	3,261	2,378	269.3%
Unreviewed clips	8,404	3,124	9,069	7,245	-1,159	-13.8%
Number of speakers	109	137	163	168	59	54.1%
Validated sentences	9,370	10,309	11,364	11,484	2,114	22.6%
Unvalidated sentences	2,218	2,725	2,734	2,774	556	25.1%
Reported (flagged) sentences	580	633	1,425	1,435	855	147.4%
Total duration (hours)	47.79	63.23	87.44	94.26	46.47	97.2%
Valid duration (hours)	34.83	56.76	70.55	78.67	43.84	125.9%
Average clip duration (s)	5.025	5.008	5.217	5.344	0.319	6.3%
Archive size (GB)	0.91	1.21	1.68	1.82	0.90	99.0%

Kabardian (kbd)						
Metric	**CV 23.0-2025-09-05**	**CV 24.0-2025-12-05**	**CV 25.0-2026-03-09**	**CV 26.0-2026-06-12**	**Change (23.0→26.0)**	**Increase %**
Total clips	70,522	112,462	168,564	174,133	103,611	146.9%
Validated clips	64,812	102,361	157,434	158,156	93,344	144.0%
Rejected clips	1,426	2,499	2,763	2,770	1,344	94.2%
Unreviewed clips	4,284	7,602	8,367	13,207	8,923	208.3%
Number of speakers	214	289	347	380	166	77.6%
Validated sentences	20,914	27,620	30,253	31,823	10,909	52.2%
Unvalidated sentences	2,376	2,376	2,391	2,397	21	0.9%
Reported (flagged) sentences	2,390	3,359	3,995	4,015	1,625	68.0%
Total duration (hours)	119.77	194.74	291.40	302.06	182.29	152.2%
Valid duration (hours)	110.07	177.25	272.16	274.34	164.27	149.2%

Average clip duration (s)	6.114	6.234	6.223	6.245	0.131	2.1%
Archive size (GB)	2.26	3.64	5.52	5.73	3.47	153.8%

Error distribution of sentences processed using the tool was calculated as shown in Table 10:

Table 10. Sentence Error Analysis for MCV Adyghe (ady) and Kabardian (kbd)
Calculated across all .txt and .tsv files generated via the TXT File Creator & Validator tool.

Error Category	**Adyghe (ady)**	**Kabardian (kbd)**
Mixed Latin/Cyrillic characters	58	125
Punctuation error	12	32
MCV constraint violation (length/allowed characters)	95	141
Total invalid sentences (line count in _inval files)	**115**	**159**
Total sentence count (source .txt files)	4898	20408
Invalid rate within total sentences	**2.35%**	**0.78%**
Sentences rejected by MCV	**4**	**14**

- Source: *.txt files in each directory contain all submitted sentences; *_inval*.tsv files contain sentences rejected by the local validator alongside reasons.
- 'Mixed Latin/Cyrillic characters': Lines containing Latin characters (e.g., D, n, v) OR Ukrainian/Belarusian Cyrillic İ/i (U+0406/U+0456) mistakenly used instead of I (palochka).
- 'Punctuation error': Disallowed/misplaced punctuation marks such as commas, quotes (", “, ”), dashes (–, —), or parentheses.
- 'MCV constraint violation': Violations of character limits (MAX_LEN), word counts (MAX_WORDS[n]), capitalization rules (UPPERCASE), numeric digits, hidden BOM (\ufeff) characters, or forbidden symbols (−, ¬, –, etc.).
- As a sentence may contain multiple error reasons, individual category sums may not equal the 'Total invalid sentences' row.
- 'Sentences rejected by MCV': Lines in rejected-sentences*.tsv files actually rejected by Mozilla Common Voice (mostly due to TOO_LONG), independent of local validator results.

Evaluations indicate that higher local validation detection rates directly reduced rejection rates on the MCV platform. Additionally, because the tool automatically fixes Latin capital "I" (U+0049), Latin lowercase "l" (U+006C), and inline digit "1" (U+0031) characters, no such character errors occurred in invalid files.

6.3. Participant Base and Variant Organization Evolution

The community layout is detailed in Table 2. Growth from 3 to over 700 active users was driven by three factors: sub-chat groups established by the Kabardian coordinator in January 2025 (734 + 326 members), the crowdsourcing participation culture among journalists in Maykop, and parallel

sentence generation by the Adyghe working group. This multi-centric structure confirms the scalability and decentralization of community leadership.

Before the intervention, all sentences were uploaded under a single default variant, precluding dialectal filtering and statistical analysis. Following intervention, variant-based organization was adopted across both languages. Automated Latin transliterations were initially executed in bulk via circ.py, but community pushback led to a focus on Cyrillic variants, causing Latin variant submissions to decline over time.

6.4. Transliteration: Functionality and Community Resistance

Observations revealed that participants who lacked Cyrillic reading habits struggled equally when reading Latin transliterated text. Conversely, community members who already had a reading habit — living both in Türkiye and in Russia, including people who are not literate in Turkish — read these sentences easily.

The wide visibility of the Latin transliterated sentences in the community came from the way the platform presents sentences. Common Voice orders the sentences it offers to a contributor by giving priority to those with the fewest recordings, and a sentence stays in the queue until its recording count reaches an upper limit. For languages without a separate limit, such as Adyghe and Kabardian, this limit is 15 recordings.[16] Because part of the Cyrillic sentences had already been read and recorded, the Latin variants uploaded later had the lowest recording counts and were therefore offered to all contributors first. Everyone who read and recorded these sentences was literate in Adyghe or Kabardian; they made these recordings so that the sentences would leave the queue and not appear again.

A significant finding of this study is the community resistance observed in both Adyghe and Kabardian groups regarding transliteration. For the first ~4,000 sentences per language, the Cyrillic variant was uploaded first, and then the Latin transliteration of the same sentences was uploaded as well (about 8,000 sentences per language in total); however, community opposition led to abandoning Latin variants in favor of pure Cyrillic.

7. Discussion

7.1. DBR Contributions in This Case

This study presents a concrete case of how DBR operates within endangered language technology contexts. The three contributions defined by Reeves and McKenney (Reeves & McKenney, 2013) materialized as follows:

- Intervention design: The Excel-VBA tool emerged as a direct response to field bottlenecks, maturing across seven iterative releases.
- Theoretical knowledge: The concept of "invisible field tools" in endangered language technologies represents the core theoretical takeaway.
- Professional development: Blending developer, operator, and researcher roles allowed field insights to immediately inform system design.

[16] In the Common Voice server code, the default upper limit of recordings per sentence is defined as MAX_CLIP_PER_SENTENCE = 15. Lower language-specific limits are also defined: 1 for German, French, and Kinyarwanda, and 2 for English, Spanish, and Kabyle. Adyghe and Kabardian are not in this list, so the default value applies. The sentence queue is ordered with ORDER BY s.clips_count ASC, which gives priority to the sentences with the fewest recordings. Source: common-voice/common-voice, server/src/lib/model/db.ts: https://github.com/common-voice/common-voice/blob/main/server/src/lib/model/db.ts (visited on 2026-08-13).

Kılınç and Özkök's (Kılınç & Özkök, 2025) characterization of DBR as "pragmatic, grounded, interactive, iterative, and flexible" is evidenced by how each version (v1.0.0 form additions, v1.0.1 circ.py pause, v1.1.0 error handling, v1.1.1 bug fix, v1.2.0 automated variant grouping, v2.0.0 removal of the external Python dependency, v2.1.0 correction of the defects found during the checks made for publication) was driven by concrete feedback or data: field observation for the first six versions, and the publication checks for the last one.

These seven versions do not carry equal weight; the iteration rhythm worked in three different phases. The first phase (v1.0.0–v1.2.0) consists of short cycles: because the tool was already in use, its shortcomings became visible during real operation, and since the developer and the operator were the same person, there was neither a reporting delay nor a loss of meaning between the observation and the technical response. At the end of this phase, v1.2.0 confirmed the functional solution of the tool — automatic grouping by variant and domain — in the field. The second phase (from v1.2.0 to v2.0.0) is different in kind and took clearly longer: the question was no longer “does the tool solve the task?” but “can someone other than the researcher run it?”. This distinction shows that DBR iterations are not all the same: cycles that validate function and cycles that build transferability work on different time scales.

The third phase (from v2.0.0 to v2.1.0) came from a different source. It did not start with a field observation, and it was not about transferability. It started while the study was being prepared for publication. Publishing the tool's tables as a separate data set, and writing this paper, forced the same logic to be built a second time in another context. That second construction revealed defects that had stayed invisible in the field for two years (see 7.2). In the DBR literature, iteration is usually described as a feedback loop that runs from practice to research. In this case a loop in the opposite direction was also observed: the research activity itself — preparing for publication and writing — became a way of checking the intervention and made its defects visible.

The methodological gain of the DBR framework in this case is that it required a record of which field observation each version responded to. This produced a traceable chain of design reasons that is independent of the tool itself. The code of this tool cannot be taken over without conditions: the tool depends on Excel and is released under a license that requires the written permission of the author for derivative works (see 7.3). However, what is transferable is not the code but the order in which the problems were solved: first build something that works in the hands of a single expert operator, validate the data model in the field, and only then remove the expertise barrier. The reverse order — designing the tool for everyone from the start — carries the risk of freezing a data model that has not yet been validated. This ordering is offered as a hypothesis drawn from a single case and needs to be tested in other communities.

The combination of the researcher, developer and operator roles in one person acted in this case as both an engine and a risk. The concentration that made fast iteration possible also tied the sustainability of the tool to a single person (see 7.5). What is notable is that this limitation was not only reported but also shaped the design: the architectural decision behind v2.0.0 — removing the external Python dependency — aims to dissolve the role concentration by lowering the technical knowledge needed to run the tool. The result of this decision can be tested: the tool is planned to be distributed to the sentence-collection and review volunteers in the community (see 7.6), and this distribution will show whether the task can be spread from one person to many.

7.2. Significance of the "Invisible" Tool Layer

Levy's assertion that CALL tools must transition "from extraordinary to ordinary" to achieve normalization (Levy, 2013) applies directly to this work. Where circ.py's console interface demanded "extraordinary" technical expertise, the Excel-VBA tool abstracted this complexity into an accessible format. As the tool became "invisible" in practice, the researcher's role naturally shifted toward an operational function. Here, "invisibility" refers not only to hiding technical mechanics, but also to the

tool becoming a natural workflow component while allowing the researcher to step into the background.

How that line is held in practice is a separate design question. Hiding the mechanics of a tool and hiding its decisions are not the same thing: the first makes the tool usable, the second makes it impossible to check. To make the difference concrete, the linguistic decisions the tool carries inside it — the correspondences between Cyrillic letters and Turkish Latin letters — were taken out of the tool and published as a separate, citable data set (Nemlioğlu M. U., Adyghe & Kabardian Language Resources (Version 1.0.0) [Data set], 2026). A user can now see and discuss which linguistic decisions the tool rests on without having to know how it works. The tool layer stays invisible while the decision layer becomes visible.

The risk that invisibility weakens oversight did not stay abstract in this study; it was observed in the tool itself. While the transliteration table was being prepared for separate publication, a cell-by-cell comparison showed that three letters for dialectal sounds that are not part of the literary language or the alphabet (гь, кь, кIь) were not defined in the mapping table. Because these letters were not consumed as single units, the rule that deletes a stray ь at the end of the conversion removed the sign and the letter disappeared without warning; distinct words such as гьэ and гэ came out as the same Latin string. The defect produced no error message and the output looked well formed, so it was not noticed in use. This is the argument about invisible tools seen from the other side: the defect of an invisible tool is invisible too. The correction was applied in v2.1.0, adding the three letters to the mapping tables of both languages.

One reason why the defect went unnoticed for so long lies in the structure of the corpus. The sentences prepared for Common Voice were written in the literary language, not in a dialect, and the variant labels were based on region rather than dialect (see 7.4). These dialectal letters therefore almost never passed through the transliteration path, and the defect left no visible trace in the output.

The same thing had happened earlier, while v2.0.0 was being prepared. Porting the logic of circ.py into VBA required a line-by-line comparison of the two implementations, and this revealed three defects in the source script: the uppercase palochka was mapped to an empty string, so a word-initial I lost its glottal mark; the Latin form of кIо contained a Cyrillic о (U+043E), which is exactly the error class the tool's own validator is meant to catch; and one rule in the list could never match. These corrections were applied in the VBA version used in this study, and a review report was sent to the developer of the source script.

Taken together, these two events point to a single phenomenon, and the finding is more general than the defects themselves. The defects were not created by growing complexity. All three had been in the simpler source script for a long time. What brought them to light was the same thing each time: the logic was derived a second time, independently. Porting it to another language, publishing the table as a separate data set, and writing this paper all did the same work. Each forced the same knowledge to be rebuilt by another route. In an invisible tool such a step does not happen by itself. As long as the tool runs and the output looks well formed, nothing calls for a second reading. This is the cost of invisibility.

There is one point where complexity does play a part, and that is duplication. The same mapping table was kept in three places: in the working spreadsheet, in the source script and in the VBA port. Most of the defects that were found are divergences between these copies. The remedy addresses exactly that: the table is now published as a single citable source, and the codepoints are generated from the characters instead of being typed by hand. The copies now have one common origin that can be checked.

7.3. Data Sovereignty and Licensing Strategy

The Excel-VBA tool is released under the NEMERKO Limited Open License (NLO License). The license allows the tool to be used freely for personal, educational, academic-research and non-commercial language-documentation purposes — including the sentence preparation and validation work of community volunteers. In return, modifications, forks and derivative works, as well as commercial use, require the written permission of the author; requests are assessed on the basis of their purpose and may be granted free of charge or for a fee, may be granted with conditions, or may be refused. In this respect the license works as a controlled gate rather than a closed wall. The principle of data sovereignty — as formulated in the Indigenous Data Governance literature — holds that language communities must retain control over their own data (Carroll et al., 2020). This principle concerns the data a tool touches, not the tool itself: a field tool weakens the community's control the moment it claims rights over the data that passes through it. For invisible field tools this is a design condition — a tool must not interfere with data sovereignty. The licence used here was written so that this condition is not broken: the restriction applies only to the design and the code of the software, and it does not apply to any data the tool processes.

The scope of the restriction is decisive here: the limitation applies only to the tool layer, not to the language data. The sentences produced by the community are collected under the CC0 (public domain) license and delivered to Common Voice on that condition; because of this requirement, only sentences written by the contributors themselves or taken from openly licensed sources are accepted, and texts protected by copyright are filtered out during collection. The resulting corpus remains publicly available without any restriction. The same holds for the data the tool consumes: the transliteration tables that the tool carries inside it have also been published separately under CC BY 4.0 (see 7.2). In other words, both the data the tool produces and the data it uses are completely free; only modifying and deriving from the tool requires permission.

This strategy is open to debate in terms of open-source software ethics. By the definition of the Open Source Initiative, the NLO License is not an open-source license but a source-available one, and it may create friction for reuse within the language technology community. On the other hand, the license aims to protect the tool layer that grows around endangered language data from being separated from the community and turned into a commodity. The choice is a deliberate balance between these two costs: while the corpus is released freely, the infrastructure that produces it remains open to community-oriented control over who develops it and for what purpose.

7.4. Dialectal Complexity and Multi-Dialectal Design

While the tool does not automatically resolve dialectal nuances, its variant-based file organization provides infrastructure for future dialect-conscious data analysis. The community-driven dialect contribution model advocated by Öktem and Boudichat (Öktem & Boudichat, 2025) for Tamazight represents a long-term design target for this project. Nevertheless, resistance to transliteration underscores that technical optimization must balance community expectations. The "invisible tool" metaphor has limits: as tools become invisible, the community's capacity to critique underlying design decisions may inadvertently decrease.

One distinction should be made clear here. In this study the term variant refers to region, not to dialect. The sentences were written in the literary language, and the variant label was used to separate words that enter the language differently in different countries. For example, speakers living in Russia use еджакIо for a school pupil and студент for a university student, while in the Turkish variant еджакIо covers both. Institution names and some other terms also change by region in the same way. Older dialect categories such as Shapsug, Abzakh or Temirgoy are not represented as separate variants in the corpus. The variant-based file organisation of the tool therefore provides the infrastructure that a dialect-aware corpus would need, but it has not yet been used for that purpose.

This demonstrates that language technology interventions carry social, identity, and orthographic policy implications beyond technical utility. In endangered language communities, native

orthographies represent cultural identity, leading speakers to resist external alternatives (Simpson, 2025). Viewed alongside claims that digital inclusion fosters equitable language technologies, this resistance highlights that "equitable participation" requires community consensus, not merely technical access.

7.5. Limitations

Several limitations should be noted:

Single-case limitation: This study represents a DBR case study focused on a single language pair and a single developer-operator; findings may not generalize directly to different language communities or multi-developer settings.

- Single-developer dependency: The primary developer served as the main operator, creating a sustainability risk despite aligning with DBR's "researcher in the field" model. Future plans aim to distribute operational responsibilities across multiple team members.
- VBA platform lock-in: Dependency on Microsoft Excel limits cross-platform accessibility. Transitioning to a web-based architecture for macOS/Linux users remains a planned future step.
- Nature of time metrics: Reported workflow durations rely on observational estimates across sessions rather than strict chronometric logging.
- Validation rate isolation: Platform-level validation rates reflect broader community dynamics, making it difficult to isolate the tool's independent statistical impact.
- Community constraints on variant usage: Although automated validation and grouping were designed for multi-dialectal workflows, community resistance to transliteration limited active variant usage, concentrating submissions primarily in Cyrillic variants.

7.6. Future Work: Transition to a Web-Based Architecture

The Excel-VBA tool served as an interim field solution. Future work will build upon these architectural insights to develop a web-based platform. Zinn's Language Resource Switchboard model (Zinn, 2018) offers a framework for integrating utility tools under a single interface. Utilizing direct APIs or bulk upload endpoints could eventually eliminate manual copy-pasting. However, because Common Voice rate limits rapid automated submissions for security reasons, full automation will require platform policy adjustments. Web tool development will systematically incorporate community approval mechanisms into the design process.

Until the web-based tool is available, the present tool is planned to be given to the community volunteers who collect and review sentences, so that the immediate need is met. This distribution is also a test of the hypothesis stated in 7.1: does lowering the technical knowledge needed to run the tool actually spread the task from one person to many? The outcome — how many volunteers can use the tool on their own, and where they need support — will also set the design requirements for the web tool.

8. Conclusion

This study investigated an Excel-VBA validation tool developed to support the sustainable inclusion of two endangered languages, Adyghe (ady) and Kabardian (kbd), on Mozilla Common Voice, using a Design-Based Research (DBR) framework.

Key findings include:

1. Workflow optimization: for a batch of 1,000 sentences, the time needed for character validation and upload to Common Voice dropped from several days to about 20–50 minutes (developer-operator estimate, observed over several sessions).
2. Quality control: Automated palochka verification and variant-based file organization successfully supported multi-dialectal and multi-domain data management.
3. Community dynamics: Community resistance to Latin transliteration demonstrated that orthographic identity and community preferences can outweigh pure technical functionality in language technology design.
4. Where the checking came from: the defects of the tool did not become visible during field use. They became visible when its logic was derived again, independently — when it was ported to another language, when its tables were published as a separate data set, and when this paper was written.

This study argues that an "invisible field tools" layer is critical for participant sustainability in endangered language technologies. In DBR settings where the developer and operator are the same individual, field feedback directly informs system design, producing tangible gains across iterations. However, "invisibility" must not undermine the community's oversight; community consent must be continuously re-confirmed.

Future work will focus on transitioning the utility into a web-based application, fully deploying multi-dialectal contribution models, investigating deeper integration with Common Voice policies, and embedding structured community authorization workflows into technical development. In these approval mechanisms, topics that need a technical or scientific background should be grouped and given their own permission levels, so that expert support inside the community is used more efficiently.

Appendix A: Technical Architecture Details

This appendix details the four architectural layers composing v2.1.0 of the TXT File Creator & Validator tool:

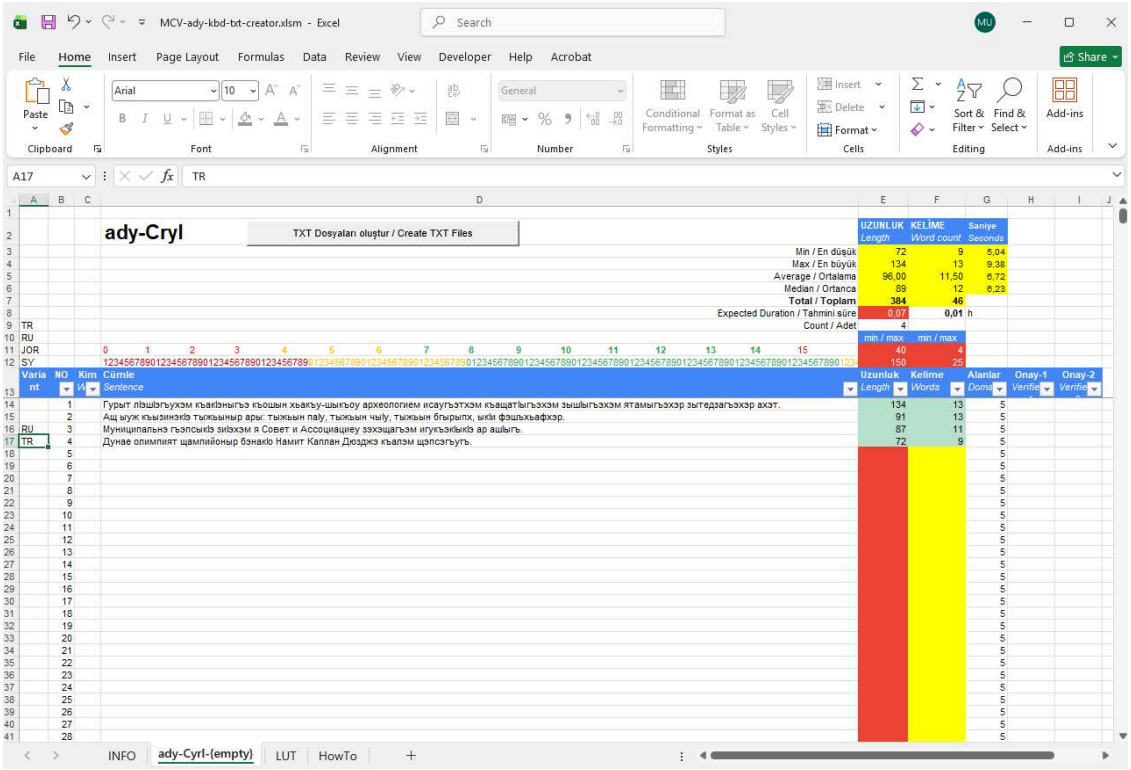
1. Data Collection Layer: Sentences are stored in a 9-column Excel sheet (Variant, No, Who, Sentence, Length, Words, Domains, Verify-1, Verify-2). The Domains column (G) references a 12-coded LUT; sentences are assigned 1–3 codes separated by commas (defaulting to code 5 [General]).
2. Domain LUT Layer: A dedicated LUT sheet maintains numerical codes alongside English, Turkish, Adyghe, and Russian domain labels. While sentences may carry multiple domain tags, file generation writes each sentence to a single output file reflecting its tag combination (e.g., tags 5, 8 yield a file named with 5-8).
3. Classification and Generation Layer: Sentences are parsed via the VBA form based on sheet selection, a file limit (recommended maximum: 650 sentences/file), target folder, variant tags (Column A), and domain codes (Column G), and exported as UTF-8 .txt files.
4. Validation, Statistics, and Transliteration Layer: Output files are processed in execution modes (validate, stats, trans) using the local VBA engine without invoking external Python

interpreters. Modules utilize late binding (CreateObject("ADODB.Stream")) to execute UTF-8 file I/O operations in-process.

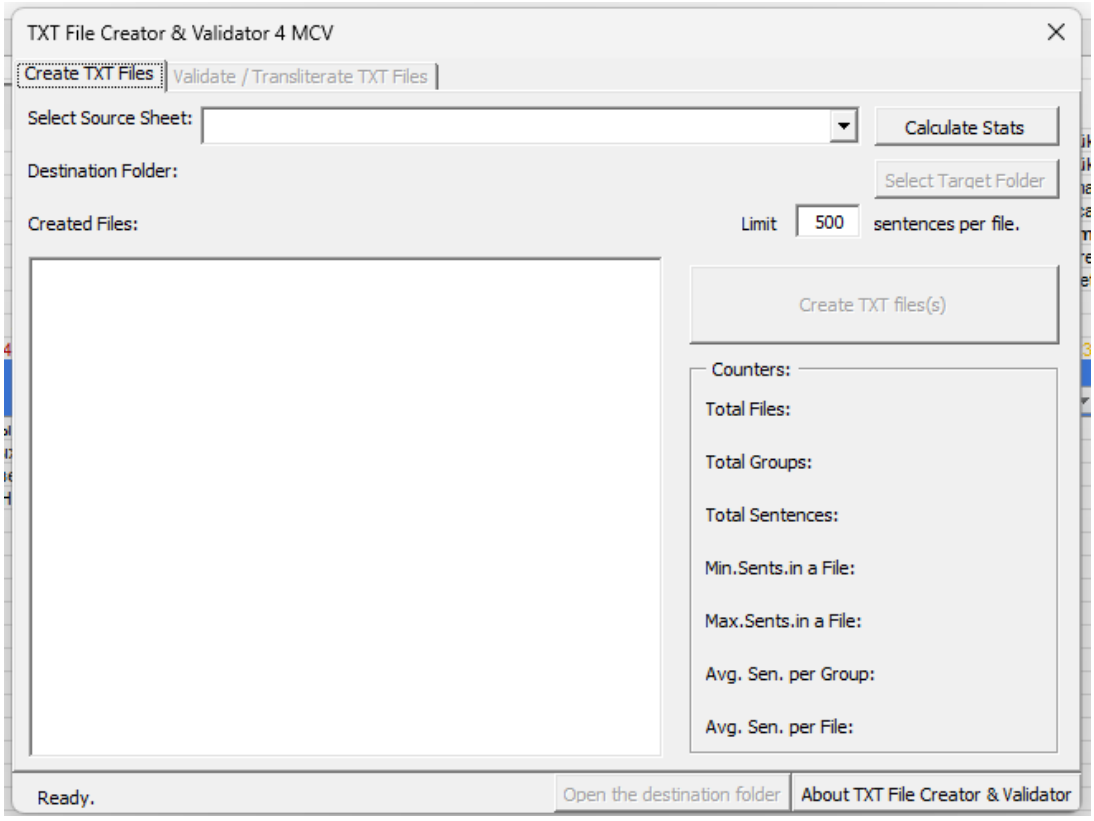
Appendix B: Application Screenshots

This appendix shows the stages of use of the tool: the Excel sheet where sentences are collected (Fig. 1), the main VBA form interface (Fig. 2), and the file organisation produced after variant-based grouping (Fig. 3).

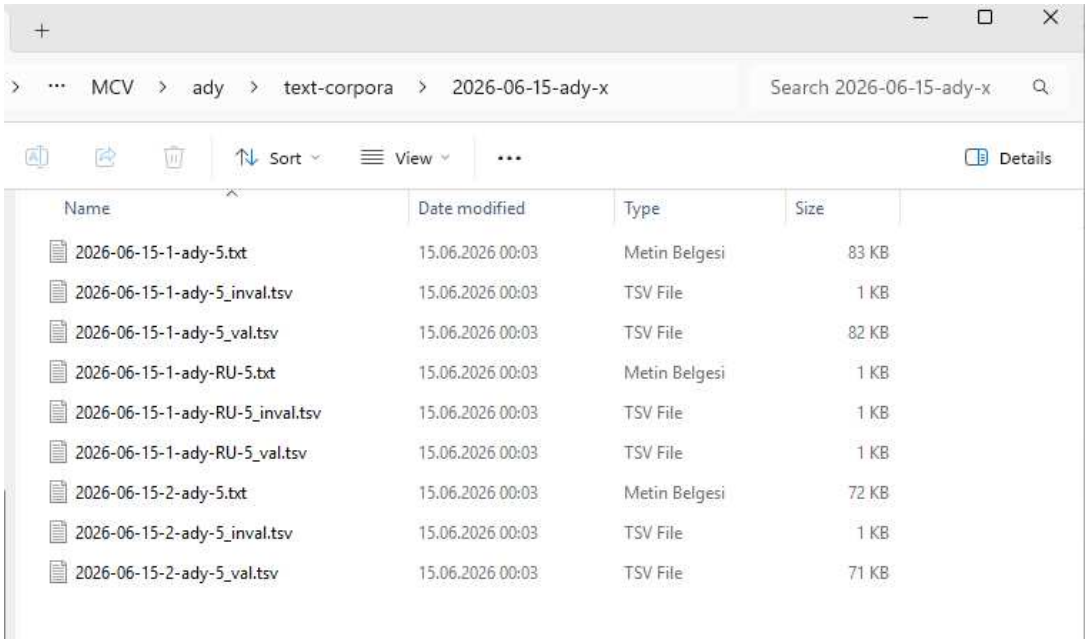
- Fig. 1 Sentence collection Excel sheet



- Fig. 2 Main VBA form interface (upload, grouping, and validation controls)



- Fig. 3 File organization following automated variant-based grouping



Statements and Declarations

Funding: The researcher received no external funding for this study or for the development of the tool it describes.

Competing interests: The author developed the tool that is the subject of this study and holds its copyright. The tool is released under a licence that requires the written permission of the author for derivative works and for commercial use. The author is also the Adyghe language representative in the Mozilla Common Voice community. The author declares no financial competing interests.

Ethics approval: The transliteration test carried out within this study was approved by the Ethics Committee of Düzce University (decision dated 27.02.2025, no. 2025/84). The development and field use of the tool did not require separate ethics approval, because the tool collects no personal data and participation was public and voluntary.

Consent to participate: Informed consent was obtained from the people who took part in the transliteration test. The community data reported in the study consist of aggregate figures from public groups and contain no individual-level data; no participant is identifiable in the findings.

Data availability: The transliteration tables used by the tool are published under CC BY 4.0 (Nemlioğlu, 2026a). The sentence corpus produced in this study is publicly available through the Mozilla Common Voice platform under CC0.

Code availability: The tool is archived as citable software (Nemlioğlu, 2026b). It is released under the NEMERKO Limited Open License v2.0, a source-available licence.

Author contributions: The study was designed, carried out and written by the sole author.

References

Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F. M., & Weber, G. (2020, March 5). *Common Voice: A Massively-Multilingual Speech Corpus.* Retrieved 06 30, 2026, from arXiv [cs.CL]: https://arxiv.org/pdf/1912.06670

Arkadiev, P. (2022). Morphology of the Caucasian Languages. *Jezikoslovni zapiski*, 7-38. Retrieved from https://ojs.zrc-sazu.si/jz/article/view/11118

Arkadiev, P., & Lander, Y. (2021). The Northwest Caucasian languages. In *The Oxford Handbook of Languages of the Caucasus, Oxford Handbooks* (pp. 368-446). Oxford Academic. https://doi.org/10.1093/oxfordhb/9780190690694.013.3

Austin, P. (2023). Citizens, speakers, and documentation of (endangered) languages and cultures. *Journal of Endangered Languages, 13*(23), 7-19. Retrieved from https://dergipark.org.tr/tr/pub/tdd/issue/82059/1412197

Bakov, H. I. (2023). On the problem of the general Adyghe literary language. *Adyghe International Scientific Journal*, 64-73. https://doi.org/10.47928/1726-9946-2023-23-l-64-73

Bettinson, M. (2015). Towards Language Documentation 2.0: Imagining a crowdsourcing revolution. *4th International Conference on Language Documentation and Conservation (ICLDC).* Manoa: University of Hawai‘i. Retrieved from http://hdl.handle.net/10125/25302

Carroll, S. R., Garba, I., Figueroa-Rodríguez, O. L., Holbrook, J., Lovett, R., Materechera, S., Parsons, M., Raseroka, K., Rodriguez-Lonebear, D., Rowe, R., Sara, R., Walker, J. D., Anderson, J., & Hudson, M. (2020). The CARE Principles for Indigenous Data Governance. *Data Science Journal, 19*(1), 43. https://doi.org/10.5334/dsj-2020-043

Cela-Ranilla, J., Esteve-Mon, F., & Sánchez-Caballé, A. (2025). Design-based research in higher education: A systematic literature review between 2019 and 2023. *Australasian Journal of Educational Technology, 41*(4), 67-83. https://doi.org/10.14742/ajet.10380

Crowston, K., & Shamshurin, I. (2017). Core-periphery communication and the success of free/libre open source software projects. *Journal of Internet Services and Applications, 8*(1), 10. https://doi.org/10.1186/s13174-017-0061-4

Çöltekin, Ç., Doğruöz, A., & Çetinoğlu, Ö. (2022). Resources for Turkish natural language processing: A critical survey. *Language Resources and Evaluation, 57*, 449-488. https://doi.org/10.1007/s10579-022-09605-4

Demir, B. (2025). May 21, 1864: The Great Circassoan Exile: Causes and Consequences. *Troyacademy | International Journal of Social Science*, 98-108. https://doi.org/10.31454/troyacademy.1749202

Demir, Y. (2022). A Review of Key Concepts in Language Maintenance: Directions for Maintaining Endangered Languages in Turkey. *JofEL - Journal of Endangered Languages, 12*(21), 201-210. Retrieved from https://dergipark.org.tr/tr/pub/tdd/issue/71566/1119427

Gaffas, Z. (2024). Using social media platform X to enhance student medical English learning: an attempt based on design-based research (DBR) in a medical English for specific purposes (ESP) course. https://doi.org/10.1186/s40862-024-00281-1

Godard, P., Adda, G., Adda-Decker, M., Benjumea, J., Besacier, L., Cooper-Leavitt, J., Kouarata, G.-N., Lamel, L., Maynard, H., Mueller, M., Rialland, A., Stueker, S., Yvon, F., & Zanon-Boito, M. (2018). A Very Low Resource Language Speech Corpus for Computational Language Documentation Experiments. *Eleventh International Conference on Language Resources and Evaluation.* Miyazaki, Japan: European Language Resources Association (ELRA). https://doi.org/10.63317/52v32ouunucq

Gurtueva, I., & Kamensky, M. (2024). Interfered Russian Speech of Native Speakers of the Kabardian-Circassian Language: An Experimental Study. *Science Journal of Volgograd State University. Linguistics, 23*(4), 119-141. https://doi.org/10.15688/jvolsu2.2024.4.10

Höhlig, M. (2021). *Adıge Edebi Dilindeki Ünsüz ve Ünlü Seslerin Fonetik Betimlemeleri.* (M. U. Nemlioğlu, Trans.) Düzce.

Karmova, M., Khachmafova, Z., Хабекирова, З., Khachetsukova, Z., & Шхумишхова, А. (2023). Digitalization: Advantage or disadvantage in the issue of preserving the languages of minority ethnic groups (on the example of the Abaza language). *SHS Web of Conferences, 172*, 1005-1005. https://doi.org/10.1051/shsconf/202317201005

Kılınç, A., & Özkök, G. (2025). Design-Based Research in teaching Turkish to foreigners: Models, procedure and a design example. https://doi.org/10.5281/zenodo.15154147

Kornai, A. (2013). Digital Language Death. *PLOS One, 8*(10). https://doi.org/10.1371/journal.pone.0077056

Lange, H., & Aznar, J. (2022). RefCo and its Checker: Improving Language Documentation Corpora's Reusability Through a Semi-Automatic Review Process. https://doi.org/10.63317/4b52btqk88mq

Leivada, E., D'Alessandro, R., & Grohmann, K. (2019). Eliciting Big Data From Small, Young, or Non-standard Languages: 10 Experimental Challenges. *Frontiers in Psychology, 10*, 313-313. https://doi.org/10.3389/fpsyg.2019.00313

Levy, M. (2013). Design-based research and the quest for normalization in CALL. In J. C. Rodríguez, & C. Pardo-Ballester (Eds.), *Design-Based Research in CALL* (pp. 31-40). San Marcos, TX: CALICO. Retrieved from https://espace.library.uq.edu.au/view/UQ:321300

Mbogho, A., Awuor, Q., Kipkebut, A., Wanzare, L., & Oloo, V. (2025). *Building low-resource African language corpora: A case study of Kidawida, Kalenjin and Dholuo.* https://doi.org/10.48550/arxiv.2501.11003

Moseley, C. (2010). *UNESCO Atlas of the World's Languages in Danger.* Retrieved 03 5, 2016, from UNESCO: http://www.unesco.org/languages-atlas/index.php?hl=en&page=atlasmap&lid=1064

Mozilla Common Voice. (2021, December 2). *How we're making Common Voice even more linguistically inclusive*. Retrieved from Mozilla Foundation: https://www.mozillafoundation.org/en/blog/how-we-are-making-common-voice-even-more-linguistically-inclusive/

Nemlioğlu, M. U. (2016, Aralık). Adıge (Çerkes) Diasporasının Bugünü. In M. Khachemizova, F. Altın, & Y. Şen (Ed.). *1*, pp. 350-365. Düzce: APRA Yayıncılık Ltd.

Nemlioğlu, M. U. (2018). *Bilişim ve İnternette Adıge Dilinin Kullanımının Yaygınlaşmasının Sağlanması (Yüksek Lisans Tezi).* Düzce Üniversitesi Sosyal Bilimler Enstitüsü, Düzce. Retrieved from https://tez.yok.gov.tr/UlusalTezMerkezi/

Nemlioğlu, M. U. (2026). *Adyghe & Kabardian Language Resources (Version 1.0.0) [Data set].* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.21940474

Nemlioğlu, M. U. (2026). *TXT File Creator & Validator for MCV (ady, kbd) (Version 2.1.0) [Computer software].* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.21959549

Nemlioğlu, M. U., & Abregova, S. K. (2025). Адыгабзэкlэ клавиатурэр гъэфедэным lэрышl акъылым (искусственный интеллект) игъэсэныгъэ мэхьанэу щыриlэр. *Сохранение и развитие родного языкав условиях глобализации:современные методы и технологии* (pp. 148-154). Maykop: Адыгейский республиканский институтгуманитарных исследований им. Т.М. Керашева. Retrieved from https://www.researchgate.net/publication/395242067_ADYGABZEKE_KLAVIATURER_GEFEDENYM_ERYS_AKYLYM_ISKUSSTVENNYJ_INTELLEKT_IGESENYGE_MEHANEU_SYRIER

Öktem, A., & Boudichat, F. (2025). Awal -- Community-Powered Language Technology for Tamazight. *International Conference on Information and Communication Technologies for Amazigh (TICAM 25).* https://doi.org/10.48550/arxiv.2510.27407

Pakray, P., Gelbukh, A., & Bandyopadhyay, S. (2025). Natural language processing applications for low-resource languages. *Natural Language Processing, 31*(2), 183-197. https://doi.org/10.1017/nlp.2024.33

Qian, T., Hollingshead, K., Yoon, S., Kim, K., & Sproat, R. (2010). A Python Toolkit for Universal Transliteration. https://doi.org/10.63317/5b4rq3qiukg3

Reeves, T. C., & McKenney, S. (2013). Computer-assisted language learning and design-based research: increased complexity for sure, enhanced impact perhaps. In J. C. Rodríguez, & C. Pardo-Ballester (Eds.), *Design-Based Research in CALL* (pp. 9-22). San Marcos, TX: CALICO.

Simpson, L. (2025). From Icons to Identities: Analysing Visual Cultural Elements in Emerging Scripts. *Visible Language, 58*(2). https://doi.org/10.34314/9ecfxc51

Topçu, M. P. (2006). Çerkes-Adığe Yazısının Tarihçesi. *Nart, İki Aylık Düşün ve Kültür Dergisi, 51*, 25-30. Retrieved from https://www.academia.edu/2762087/Çerkes_Adığe_Yazısının_Tarihçesi

Tyers, F., & Meyer, J. (2021). What shall we do with an hour of data? Speech recognition for the un-\n and under-served languages of Common Voice. *arXiv:2105.04674 [cs.CL]*. https://doi.org/10.48550/arxiv.2105.04674

Wang, F., & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development, 53*(4), 5-23. https://doi.org/10.1007/BF02504682

Zaudin, K., Mukhamedovna, A., & Khautievich, U. (2024). The Adyghe (Kabardian-Circassian) Diaspora Language in Interlingual Interaction. *Cadernos de Educação Tecnologia e Sociedade, 17*(1), 293-301. https://doi.org/10.14571/brajets.v17.n1.293-301

Zdorova, N., Parshina, O., Ogly, B., Bagirokova, I., Krasikova, E., Ziubanova, A., Unarokova, S., Makerova, S., & Dragoy, O. (2023). Eye movement corpora in Adyghe and Russian: an eye-tracking study of sentence reading in bilinguals. *Frontiers in Psychology, 14*. https://doi.org/10.3389/fpsyg.2023.1212701

Zhemukhov, S., & Aktürk, Ş. (2015, September). The movement toward a monolingual nation in Russia: The language policy in the Circassian republics of the Northern Caucasus. *Journal of Caucasian Studies (JOCAS), 1*(1), 33-68. https://doi.org/10.21488/jocas.46077

Zinn, C. (2018). The Language Resource Switchboard. https://doi.org/10.1162/coli_a_00329

Анчек, С. Х. (2022, February 15). *О становлении адыгейского языка.* Retrieved 08 18, 2026, from Советская Адыгея: https://sovetskaya-adygeya.ru/2022/02/15/o-stanovlenii-adygejskogo-yazyka-material-doktora-filologicheskih-nauk-zavedujushhej-otdelom-yazyka-arigi-im-t-kerasheva-suret-anchek/

Бырсыр, Б. М. (2017). Адыгэ алфавитымрэ тхыбзэмрэ къакIугъэ гъогур (История развития адыгского алфавита и письменности). *Вестник науки Адыгейского республиканского института гуманытарных исследований имени Т.М. КЕРАШЕВА, 11*(35), 12-15. Retrieved from https://www.elibrary.ru/item.asp?id=30677369

Кумахов, М. А. (2013). *Кабардино-Черкесский язик.* Nalchik: Институт гуманитарных исследований Правительства КБР и КБНЦ РАН.

НПП "Гарант-Сервис". (1995). *Раздел 1. Основы конституционного строя (ст.ст. 1 - 17) - Статья 5*. Retrieved 04 19, 2026, from Конституция РФ - Конституция Республики Адыгея: https://constitution.garant.ru/region/cons_adig/chapter/5633a92d35b966c2ba2f1e859e7bdd69/

НПП "Гарант-Сервис". (1996). *Раздел I (гл.гл. 1-10) / Глава 1. Основы конституционного строя (ст.ст. 1-13) - Статья 11.* Retrieved 04 19, 2026, from Конституция РФ - Конституция Карачаево-Черкесской Республики: https://constitution.garant.ru/region/cons_karach/chapter/9d78f2e21a0e8d6e5a75ac4e4a939832/

НПП "Гарант-Сервис". (1997). *Раздел I (гл.гл. I-X) / Глава III Государственное устройство (ст.ст. 67-77) / Статья 76*. Retrieved 04 19, 2026, from Конституция РФ - Конституция Кабардино-Балкарской Республики: https://constitution.garant.ru/region/cons_kabardin/chapter/ef67419dbaa01e4d228acc1d3cf42314/