User-Centered Design of Digital Tools for Sociolinguistic Studies in Under-Resourced Languages

Jonas Adler* **and Carsten Scholle*** **and Daniel Buschek**
University of Bayreuth, Mobile Intelligent User Interfaces
{name}.{surname}@uni-bayreuth.de

Nicolo' Brandizzi
Sapienza University of Rome, DIAG
brandizzi@uniroma1.it

Muhadj Adnan
University of Bayreuth, Arabic Studies
muhadj.adnan@uni-bayreuth.de

Abstract

Investigating language variation is a core aspect of sociolinguistics, especially through the use of linguistic corpora. Collecting and analyzing spoken language in text-based corpora can be time-consuming and error-prone, especially for under-resourced languages with limited software assistance. This paper explores the language variation research process using a User-Centered Design (UCD) approach from the field of Human-Computer Interaction (HCI), offering guidelines for the development of digital tools for sociolinguists. We interviewed four researchers, observed their workflows and software usage, and analyzed the data using Grounded Theory. This revealed key challenges in manual tasks, software assistance, and data management. Based on these insights, we identified a set of requirements that future tools should meet to be valuable for researchers in this domain. The paper concludes by proposing design concepts with sketches and prototypes based on the identified requirements. These concepts aim to guide the implementation of a fully functional, open-source tool. This work presents an interdisciplinary approach between sociolinguistics and HCI by emphasizing the practical aspects of research that are often overlooked.

1 Introduction

Researchers in sociolinguistics often use corpora for investigations of language structure and usage, identifying linguistic characteristics and patterns in different contexts. Researchers gain insights into these patterns by analyzing a collection of authentic texts (corpora) quantitatively and/or qualitatively (Biber et al., 1998). The importance of this field has particularly increased due to factors such as global interconnection and continuous increase in migration. Notably, the growing contact of speakers of different languages and varieties adds relevance to investigating and analyzing language variation and change. This research often involves collecting and transcribing natural spoken language to identify distinct linguistic features and discover patterns during analysis, though other methods, such as sociolinguistic experiments, are also employed.

Yet, the potential of this research area is frequently accompanied by many challenges that influence how research is conducted. For instance, the exponential increase of available data enhances the possibilities for research, but dealing with these large quantities of data poses new challenges for researchers and requires them to incorporate computer-assisted tools (Mair, 2018). However, transitioning to digital solutions can be difficult when faced with unfamiliar tools and a lack of knowledge about research strategies. In under-resourced languages, these issues are often compounded by the absence of assistance tools, like automatic language recognition software, leading to a time-consuming manual transcription process (Chakravarthi et al., 2019). This *transcription bottleneck* (Bird, 2021) is particularly problematic for under-resourced languages due to transcription difficulties. This raises the question of whether current research techniques can keep up with advancing technology and changing language dynamics.

In this paper, we aim to create a bridge between Human-Computer Interaction (HCI) and linguistics, fostering an interdisciplinary collaboration that leverages the strengths of both fields. By focusing on a *User-Centered Design* (UCD) approach, we investigate the practical workflows currently carried out by variationist sociolinguists working with lesser-resourced languages, using research on Arabic dialects as a case study. We aim to identify critical areas, such as data management, digital annotation, and automatic analysis, that limit the efficiency and quality of their studies. The outcomes intended to be applicable to a broader range

*These authors contributed equally to this work.

Proceedings of the 3rd Workshop on NLP Applications to Field Linguistics (Field Matters 2024), pages 12–27
August 16, 2024 ©2024 Association for Computational Linguistics

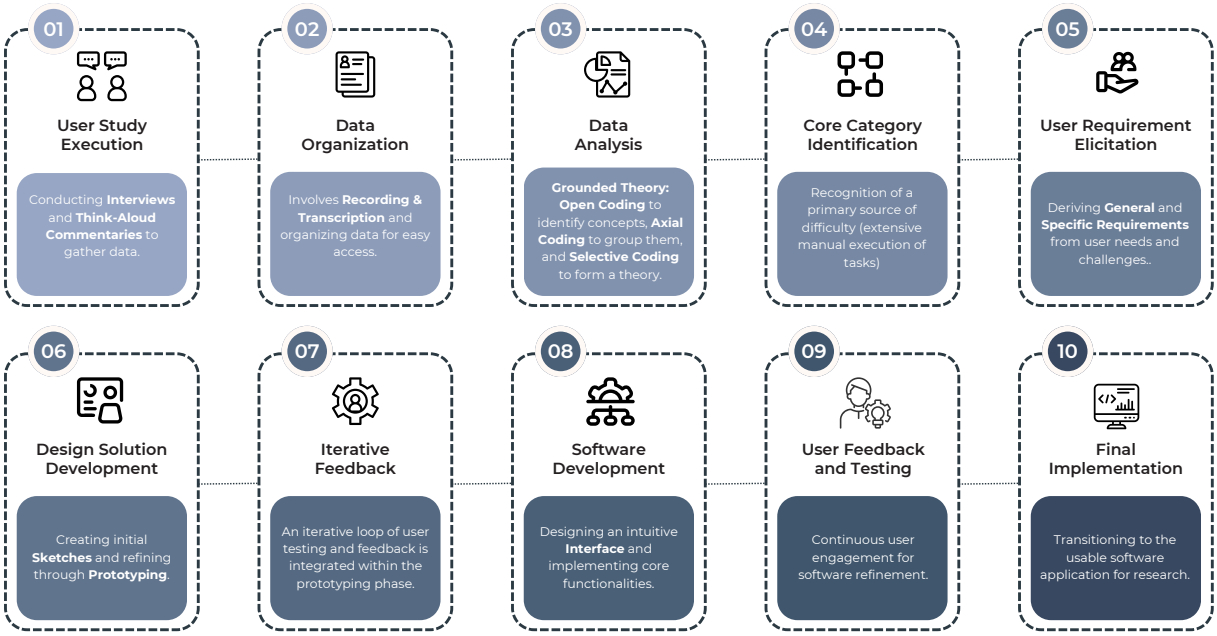

Figure 1: The *User-Centered Design Process*: steps from initial user studies and analysis to iterative design solution development, highlighting the continuous user feedback integration needed for user-friendly software interfaces.

of lesser-resourced languages. As many existing software applications invest insufficient effort in the identification of user needs for these languages, we introduce a road map for finding suitable technical solutions. Our approach enables the creation of a digital tool specifically designed to meet researchers' needs. Moreover, by actively involving researchers in the design process and valuing their feedback, we ensure that the software will be user-friendly and tailored to their requirements.

The upcoming sections outline our approach, starting with a theoretical background and overview of related works (Section 2), followed by data collection through interviews with researchers specializing in different Arabic varieties (see Section 3.1). This is followed by an in-depth data analysis (see Section 3.2 and 4.1). We then define the requirements and constraints for a user-centered software solution by considering the unique needs and challenges in this field (Section 4.2). Building on these insights, we propose a prototype that extends and enhances a previously developed tool, *CorpusCompass* (Adnan and Brandizzi, 2023), reflecting our dedication to improving the software in line with evolving research demands and user insights. Our goal is to narrow the divide between theoretical research and practical utility.

2 Theoretical Background and Related Work

This section reviews the theoretical background and relevant literature. Central to this discussion is an exploration of *User-Centered Design* principles and their various extensions (Section 2.1), which are crucial to our approach. Additionally, we present an overview of current software solutions in this domain (Section 2.5). While our work touches on language variation research, we primarily focus on UCD aspects in this section. For more detailed information on language variation research methods, please refer to Tagliamonte (2006).

2.1 User-Centered Design

User-Centered Design is the guiding principle of our research, emphasizing that software and design development should prioritize users' needs, skills, and challenges (Abras et al., 2004)(Sharp et al., 2019).

UCD proposes several key concepts and steps that can lead to a successful design process, Figure 1. One of these concepts is consulting users throughout all phases of development, especially in its early stages. This includes studying how users perform their tasks to achieve their goals, as well as understanding their preferences and characteristics. Design decisions should be informed by user research, and the process should be iterative to allow for continuous user feedback and flexible

adjustments(Lowdermilk, 2013).

Advantages of UCD The primary benefit of involving users during the development process is ensuring usability for the intended software. This is achieved by tailoring the design to address the specific problems of the users. The usability of an application is a major indicator of whether the application will be relevant for practical use or not, which makes it one of the most important factors for developing any design solution (Ritter et al., 2014).

Better usability can also impact other aspects of the users' interaction with the application. Examples include greater productivity, improved user experience, or increased accessibility (de Normalización, 2010). Consistently communicating requirements and solution concepts with target users also contributes to better expectation management. Expectation management involves clearly defining the expectations users should have regarding software functionality. This prevents failing to meet user expectations, such as not fulfilling specified requirements, which could lead to resistance or rejection of software adoption (Sharp et al., 2019).

2.2 Think-Aloud Commentaries

Think-Aloud Commentaries (TaC) are a specialized form of observations often employed in user research (Nielsen, 2012). They are used to collect user feedback within a designated research setting, for example in the context of software application design and evaluation. During TaCs, participants are asked to perform a set of representative tasks while simultaneously verbalizing all of their thoughts regarding their task execution. TaCs can be used as a data collection technique that allows for capturing subtleties and details that may go unnoticed or forgotten with alternative data collection methodologies (such as interviews and workshops). Additionally, they are also flexible and require minimal resources, which allows for easy implementation across a broad spectrum of research scenarios and online settings (Cotton and Gresty, 2006).

2.3 Grounded Theory

Grounded Theory (GT) (Corbin and Strauss, 1990) is a methodology for qualitative data analysis for text-based data sources. It enables the identification of underlying concepts in the dataset and the exploration of their relations, therefore creating a deeper understanding of the data. This is achieved by the derivation of an overarching theory, that is "grounded" in the data and explains the underlying concepts. Implementing a Grounded Theory approach usually consists of three distinct steps that help with summarizing and organizing the collected data, and therefore being able to extract valuable information from it.

The first step, *open coding*, is concerned with breaking down the data from the transcripts and notes into distinct *codes*. Each *code* is a short key phrase that precisely encapsulates an identified concept in the data. The second phase, *axial coding*, aims at grouping established *codes* that are thematically similar into different categories, as well as finding relationships between these *code groups*. Lastly, *selective coding* describes the process of formulating an overarching theory that strings all identified concepts and categories together. Core categories can be selected that serve as the foundation for this theory (Corbin and Strauss, 1990).

Additionally, it should be pointed out that these steps do not necessarily imply a fixed chronological order, but can also be performed in iterations and repetitions.

2.4 Requirements and Prototyping

Requirements dictate the necessary functionalities that a product must possess to address the previously identified issues or provide assistance in task execution (Sharp et al., 2019). After gathering sufficient amounts of data to understand the users' workflows and challenges, product (in our case, software) requirements can be specified. Over the course of this paper, product requirements will be referred to as *user requirements*. This is generally a more intuitive expression for this concept, as it implies the involvement of the user.

Requirements form the foundation for the creation of prototypes, which serve as preliminary models of the intended product or software. During prototyping, alternative design solutions are developed with the objective of identifying the most fitting design for the application context. In the context of UCD, prototyping should be integrated into an iterative process with sustained user feedback, where prototypes can be improved over different cycles (see Section 2.1). It should be pointed out that shifting the focus towards the consideration of technological possibilities should occur only at this stage of the UCD process. However, these possibilities should not serve as the driving factor for development, but rather as answers on how to fulfill

the identified requirements (Sharp et al., 2019).

2.5 Challenges in Existing Software

The study of language variation has attracted scholarly attention since the 1960s (Bayley, 2013). Early research, such as Labov's studies from that era (Labov, 2006), explored the direct relationships between linguistic and social variables without complex statistical methods. Initially, researchers primarily used simple quantitative techniques, such as percentages, cross-tabulations, and multivariate analysis (Walker, 2012; Guy, 2013). Over time, there has been a shift toward more sophisticated analytical methods. Moreover, technological advancements have led to the development of various software applications that facilitate quantitative research tasks within this domain. However, the majority of these tools are designed for a restricted subset of languages, thereby neglecting under-resourced languages (Mair, 2018).

In this field, one essential software requirement is the ability to annotate text corpora. Numerous software solutions have been developed to meet this need. Neves and Ševa (2019) conducted a comparative analysis of various annotation tools based on specific criteria. Among the tools evaluated, *WebAnno* (Yimam et al., 2013), *Brat* (Stenetorp et al., 2012), *FLAT*, and *EzTag* (Kwon et al., 2018) proved to be the best rated options. Nevertheless, none of the tools mentioned a user-centered approach during development. As a result, linguists often need to work within the limitations of these tools, rather than having tools that are flexible enough to meet their diverse requirements (Mair, 2018).

3 Methodology

This Section details the strategies for data collection (Section 3.1) and analysis (Section 3.2). It also describes how these results inform user requirements (Section 3.3), which are the core findings of this paper.

3.1 Data Collection

The data collection procedure included conducting open interviews with researchers studying language variation, as well as directly observing their workflows during a Think-Aloud Commentary (step 1, Figure 1). While TaCs are typically implemented for the evaluation of design solutions, in our study, they were used to gain detailed insights into the users' workflows and to identify the problem space.

In total, four academics from different universities participated in our user study. All of them are active researchers in Arabic linguistics and specialized in the study of different dialects (among less-resourced languages) based on oral speech (see Appendix B for users' specializations). None of the participants had prior experience with programming own solutions for their respective research tasks. The number of participants was chosen in accordance with the minimum required for discovering usability problems (Alroobaea and Mayhew, 2014; Zapata and Pow-Sang, 2012). The gathered data consists of circa four hours of interviews and two hours of observations (in the form of TaCs), where each interview took 56 minutes and each observation additional 34 minutes on average.

The interviews provided an overview of researchers' workflows, challenges and inefficiencies. This also included issues encountered with pre-existing software. The Think-Aloud Commentary on the other hand especially helped with detecting more specific difficulties, that are harder to remember during interview sessions. The interview script included questions such as the following:

- What are typical steps involved in research that deals with corpora/language variation?
- Can you tell us about the process of identifying and annotating linguistic elements?
- Do you currently use software for your work?

The interviews were recorded with both audio and video, transcribed, and finally augmented with manual notes taken during each interview session (step 2, Figure 1).

3.2 Data Analysis

We applied a Grounded Theory (GT) approach for qualitative data analysis (step 3, Figure 1).

In the first phase, we iteratively derived *codes*[1]. This iterative approach allowed us to compare *codes* with existing concepts and adjust the analysis as needed. This process repeatedly reinforced ideas and resolved conflicting concepts.

During the open coding stage, *codes* were independently extracted, then compared and reviewed in the axial coding stage. This method facilitated resolving uncertainties and conflicting *codes*, enhancing the results' quality.

[1] *Codes* are short key phrases that encapsulate singular concepts found in the data, see Section 2.3.

The analysis concluded with an *overarching theory*, formulated through the *core category* identified by the GT approach (step 4, Figure 1). This theory captures the most significant difficulties in corpus linguistics researchers' workflows and their underlying causes.

3.3 Identifying User Requirements

User requirements are derived to satisfy users' preferences, involving them continuously during the process (step 5, Figure 1). Therefore, it should be highlighted that user requirements are not to be misinterpreted as requirements held towards the user. They lay the foundation for conceptualizing and designing fitting solution ideas in later development stages.

4 Results

This section presents the findings from our Grounded Theory analysis, using open and axial coding to uncover key themes in language variation research (Section 4.1). We highlight the heavy reliance on manual processes and sparse use of software tools. A comprehensive summary is provided in Figure 2. The analysis identified central themes that guided us in understanding user requirements (see Section 4.2).

4.1 Data Analysis Results

After applying the *Open Coding* step on all of the collected data, we formulated 126 unique *codes* representing the main themes from interviews and observations. Each *code* was annotated with a participant identifier, capturing a wide variety of information for further analysis.

Grouping the *codes* for the second stage of the Grounded Theory approach (*Axial Coding*, Section 2.3) was done in two separate steps, which helped maintain a clear overview of the data. Firstly, the *codes* were classified into 12 broader groups[2], where each group contained 10-11 *codes* on average. This stage was concluded by identifying meaningful relations between the 12 general *code groups*, which enabled a comprehensive understanding of the overall concepts.

The formulated *codes* were collected in an Excel document (Microsoft Corporation, 2024) to further organize and prepare them for the next steps.

[2]A full overview of all general *code groups* that were derived from our analysis, as well the relations between them, is provided in the Appendix A.

4.1.1 Groups and Themes

The general *code groups* were formed by clustering together *codes* that share a collective theme and point to a common issue. The identified groups can be further abstracted and organized into broader themes, enabling a clearer structure and communication of our results. These themes include the common practice of *manually performing tasks*, the current *utilization of software assistance* tools, the *management of data*, and *further specific challenges* (i.e. creation, annotation, and analysis of the corpus) that occur *during* distinct steps of the *workflow*. Each of these themes covers a particular aspect of language variation research, for which the currently applied methodologies are sub-optimal or cause difficulties for researchers. The following paragraphs examine these broader themes to present the findings derived from the GT approach.

Performing Tasks Manually The implementation of manual, non-automated methodologies for performing tasks was not only prevalent throughout all interviews and observations, but it also significantly influenced and controlled every aspect throughout the progression of researchers' studies. Examples include tasks such as manually reading through the corpus and marking annotations, retrieving necessary information for the analysis by hand (i.e., by manually counting annotations), and only being able to update elements that occur multiple times in the corpus one instance at a time. Researchers also often encounter challenges with manual transcription, as exemplified by one interview-participant noting "For the transcription you sometimes need two hours to transcribe two minutes of spoken language. This makes you feel bad psychologically because you come home from work asking yourself what you have managed to do all day. Then you feel like a loser" (Interviewee #4). This reflection captures the exhaustive, slow process of manual transcription and emphasizes the psychological impact that frustrating manual work can impose. The execution of manual tasks therefore was found to be not only highly inefficient and error-prone but also placed a significant burden on the researchers who had to carry out these time-intensive activities.

Current Software Utilization Investigating current software utilization involves recognizing specific software applications that are currently used by researchers in the context of language varia-

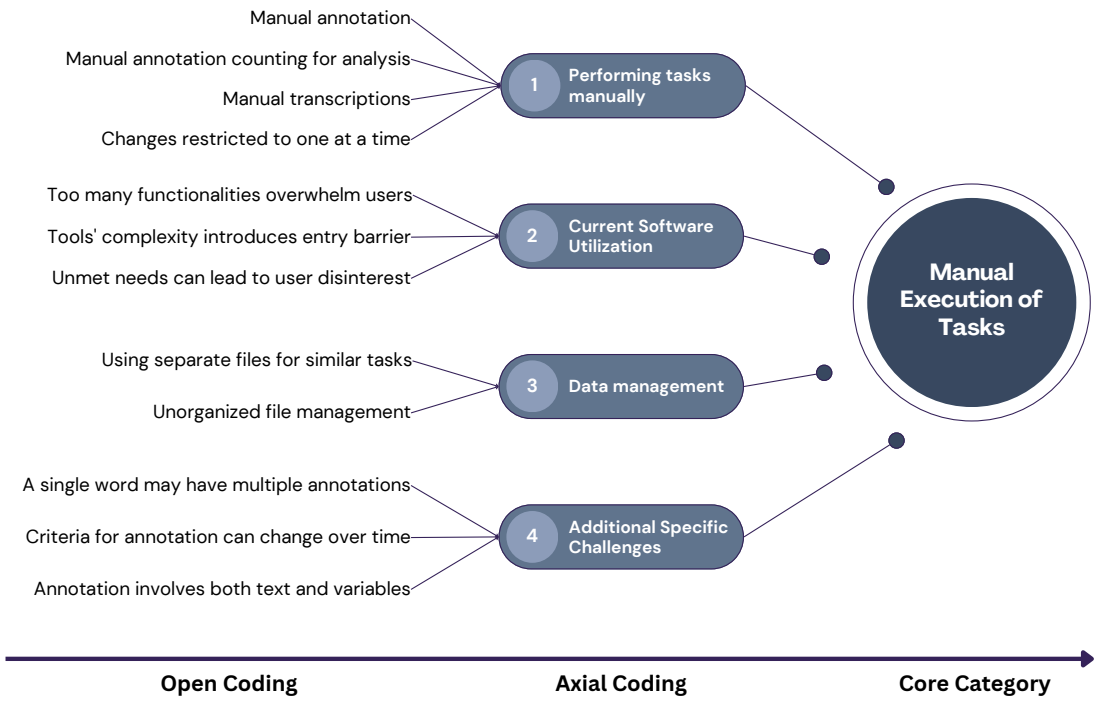

Figure 2: Stages of Analysis in the User-Centered Design Study: results from Open Coding to Axial Coding, ending with the identification of the Core Category. The process illustrates the refinement of data from initial findings to pinpointing the primary challenge of manual task execution in language research workflows.

tion studies, as well as identifying challenges they encounter while working with these tools. A representative selection of these tools was already introduced in Section 2.5. Software-related challenges primarily revolve around entry barriers that discourage the transition to digital tools. Our research indicates that these entry barriers are mainly shaped by the considerable time investment required to learn (and re-learn) the basic operations of software applications, as well as by a lack of intuitive methods for correctly importing existing data into the software. Additionally, researchers may also give up on using certain computer programs due to the software being incapable of fulfilling users' tasks and needs. One participant highlighted this issue by stating that "*Flex* felt like a software for non-linguists that need to do linguistic stuff, but it was not usable for my kind of research" (Interviewee #3). Lastly, our investigation revealed that researchers are frequently overwhelmed by tools offering an excessive amount of functionalities and interaction possibilities. This was clearly articulated by one of the participants who mentioned: "It's too much for me when programs have too many functions [...] would be good if a program is just reduced to the essentials" (Interviewee #4). This perspective highlights the discouragement they experience from either initiating or sustaining the use of a software application due to its complexity.

Data Management Our study also revealed widespread problems caused by researchers' data management. In this context, "data" includes information such as the corpus itself, speakers and their attributes, annotations in the corpus, and (intermediate) analysis results. We found that all of the interviewed researchers used different and independent files and locations for storing their data, sometimes even alternating between digital and analog environments. This practice frequently led to disorganized data structures, making navigation cumbersome and resulting in inconsistencies and critical errors in the stored data. Additionally, weak data management resulted in decreased research productivity and further demotivated researchers.

Further Challenges During Workflow The discussed themes highlighted universal challenges impacting all aspects of language variation studies, alongside unique issues specific to certain tasks. A key finding is the significant interconnection between these general and specific challenges; for example, data management problems can worsen annotation difficulties by limiting access to crucial context. Addressing these interconnected challenges is essential for developing effective design solutions and ensuring the usability of the applica-

tion.

4.1.2 Core Category

Considering all of the extracted data, challenges, and themes, our research identified the manual execution of tasks as the *core category* and primary source for existing difficulties in language variation studies. As previously mentioned, manual task execution was implemented by all researchers during a majority of their workflows and tasks in our interviews, thus negatively influencing every aspect of their research process. Given this extensive influence, we assessed that no other practice or methodology had a greater impact on its efficiency.

Identifying this core category implies the necessity of automated software solutions addressing these manual task challenges.

4.2 User Requirements

The insights obtained from the previous steps can be used to specify relevant user requirements. These requirements are derived from the specific problems and needs of the target user group and should therefore be fulfilled by the intended design solution. This section lists a selection of the most essential requirements evoked from our user study.

4.2.1 Relevant Requirements for Design Solutions

Our user research enabled the formulation of a total of 14 primary user requirements[3], with our attention directed towards reporting on the four most significant ones.

(i) *Ensuring intuitive usability* is a fundamental criterion for the design solution. The tool's user interface must provide intuitive interactions, tailored to the target users' knowledge and skills, emphasizing simplicity and focusing on essential features. This approach addresses challenges highlighted in prior user studies, guiding the requirements derivation process. (ii) Better *data-management-systems* stems from the identified data management issues. A data(base)-management system simplifies the interaction between the user and the database by ensuring consistency and managing all data-flows automatically (Dumas et al., 2018). A solution that incorporates such a system can effectively resolve data-related issues, freeing users from the responsibility of managing data storage and ensuring its consistency. (iii) *Digital Annotation* enhances the research process by automating (part of) the annotation tasks within a digital environment. This feature ensures uniform annotations across the corpus, thereby facilitating a more robust analysis. It also allows for the annotation of multiple elements simultaneously, significantly increasing productivity. Moreover, digital annotation can provide immediate feedback to users on the impact of their actions on the corpus, leading to more consistent and correct user actions. (iv) *Automatic Analysis* leverages digital annotations to enable fast, error-free counting and evaluation of data. Automatic analysis significantly facilitates research by efficiently collecting and assessing corpus annotations. This automation supports the execution of complex quantitative and statistical analyses.

[3]See Appendix C.1 for a list of the 14 primary user requirements, and Appendix C.2 for additional research directions.

4.2.2 Limitations

The limitations in meeting user requirements stem not only from technical constraints but also from the diverse personal preferences of users, leading to highly individualized approaches that make it hard to establish a set of requirements catering to all user needs. This was particularly evident in manual annotation tasks within our user study, where each participant employed a unique method for tagging linguistic features, none of which were efficient due to their manual nature. This diversity complicates the creation of uniform user requirements. While standardizing processes could offer a solution by setting expected standards, it restricts user freedom and may not fully satisfy everyone, though it could help address the broader issue more uniformly.

5 Future Directions: Engaging Users in Design and Development

Even after gathering user requirements, continuing to incorporate user feedback is crucial throughout the design and implementation phases of software development. The initial concept stage focuses on developing design solutions based on previously identified user needs, as well as employing prototypes to test and refine created design solutions. This approach ensures that the design effectively meets user expectations and informs the implementation process in later stages of development.

5.1 Concepts and Sketches

One way of starting the development of potential software solutions is by creating *sketches* (step 6, Figure 1). Sketches are essential tools for visualizing and refining ideas, serving as a bridge between

initial concepts and final designs (Tversky et al., 2003). They are encouraged to be hand-drawn, quickly made, and easily disposable, which means that each sketch has a very low cost (for an example of a sketch, see Appendix D.1). Therefore, sketching allows for rapid exploration of solution concepts, as well as evaluating and communicating these results (Greenberg et al., 2011), which makes it a powerful technique for our purpose. Easy communication through sketches allows for sharing comprehensible design ideas (i.e., with the target user group). This enables collaborative refinement of the sketches based on user feedback, which if performed iteratively (Simon, 1969) leads to converging to a specific design solution in the form of a low-fidelity-prototype (step 7, Figure 1).

5.2 Prototypes

Low-fidelity prototypes (see Appendix D.2) serve as an initial representation of the design solution concept and have been found to be extremely useful throughout the product development cycle (Virzi et al., 1996). Unlike their high-fidelity counterparts, these prototypes are not expected to replicate the final product's look or functionality fully. Instead, they can be rapidly created without losing their utility (Walker et al., 2002), facilitating the exploration of various conceptual designs and enhancing the ease of sharing these ideas for user research (Sharp et al., 2019).

Similar to the refinement of sketches, prototypes can also be refined as part of an iterative process. This process includes cycles of user feedback and fidelity enhancement that aim at ultimately creating a high-fidelity (software) prototype. High-fidelity prototypes should look and behave like the finished product, which means that they should also be close to fully functional (step 8, Figure 1). Maintaining user involvement during fidelity enhancement ensures that the resulting software remains tailored to user preferences and requirements (Sharp et al., 2019) (step 9, Figure 1).

5.3 Implementation

As a final step, our aim is to transition from a high-fidelity prototype to usable software (step 10, Figure 1). To increase the speed of development, the final software will be built on top of the functionalities presented in *CorpusCompass* (Adnan and Brandizzi, 2023). This digital tool, initially developed for corpus linguistics research, primarily focuses on automatic analysis of text-based corpora, a key component for language variation studies. Our data analysis indicates that *CorpusCompass* fulfills several user requirements identified for our project, making it a valuable technical foundation. Despite its importance, *CorpusCompass* was not developed with a focus on user needs, resulting in a user interface that is lacking in functionality and usability. To make it more useful, it is essential to conduct additional user studies and develop an interface that facilitates easy interaction. Thus better serving the needs of sociolinguists by linking advanced linguistic analysis with practical usability.

6 Conclusion

Sociolinguists studying language variation in under-resourced languages often lack supporting software tools. Addressing this requires an interdisciplinary perspective across Sociolinguistics and Human-Computer Interaction. This paper provides such a perspective and actualizes it with a UCD approach.

Our empirical work is motivated to understand, respect, and support the unique requirements of sociolinguists in their workflows. To this end, we collected rich qualitative data through interviews and observations with various academics researching language variation. Our participants were recruited from different academic institutions in Europe, and all focus on studying Arabic dialects.

This data revealed key challenges that sociolinguists encounter during their work, arising from the practice of error-prone manual text analysis and inconsistent data management approaches. The underlying root cause is a lack of software tools tailored to meet sociolinguists' specific requirements in the context of language variation research. This leads to further difficulties and inefficiencies during the research process. It is important to note that sociolinguists studying different languages, particularly those without formal writing systems, or working in different academic contexts, may face unique challenges that require tailored solutions. Thus, while our study provides valuable insights, it may not encompass all the needs of sociolinguistic researchers worldwide.

Based on these insights, we specified a set of concrete user requirements, which serve as a guideline for the design and development of better software tools. By introducing the idea of sketches and prototypes, we have illustrated how these requirements can be leveraged constructively. We plan to

implement these ideas in a functional open-source tool. Beyond our specific study here, we hope that this paper stimulates interdisciplinary perspectives to facilitate the often overlooked practical side of sociolinguistic research work.

References

Chadia Abras, Diane Maloney-Krichmar, Jenny Preece, et al. 2004. User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, 37(4):445–456.

Oliver Adams, Trevor Cohn, Graham Neubig, Hilaria Cruz, Steven Bird, and Alexis Michaud. 2019. Evaluating phonemic transcription of low-resource tonal languages for language documentation. In *11th International Conference on Language Resources and Evaluation, LREC 2018*, pages 3356–3365. European Language Resources Association (ELRA).

Muhadj Adnan and Nicolo' Brandizzi. 2023. Corpuscompass: A tool for data extraction and dataset generation in corpus linguistics. In *Proceedings of the 9th Italian Conference on Computational Linguistics*, volume 3596, pages 16–27, Venice, Italy. CEUR Workshop Proceedings.

Roobaea Alroobaea and Pam J Mayhew. 2014. How many participants are really enough for usability studies? In *2014 Science and Information Conference*, pages 48–56. IEEE.

Robert Bayley. 2013. *The Quantitative Paradigm*, chapter 4. John Wiley & Sons, Ltd.

D. Biber, S. Conrad, and R. Reppen. 1998. *Corpus Linguistics: Investigating Language Structure and Use*. Cambridge Approaches to Linguistics. Cambridge University Press.

Steven Bird. 2021. Sparse Transcription. *Computational Linguistics*, 46(4):713–744.

Bharathi Raja Chakravarthi, Mihael Arcan, and John P. McCrae. 2019. Comparison of Different Orthographies for Machine Translation of Under-Resourced Dravidian Languages. In *2nd Conference on Language, Data and Knowledge (LDK 2019)*, volume 70 of *Open Access Series in Informatics (OASIcs)*, pages 6:1–6:14, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Juliet M. Corbin and Anselm Strauss. 1990. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative sociology*, 13(1):3–21.

Deborah Cotton and Karen Gresty. 2006. Reflecting on the think-aloud method for evaluating e-learning. *British Journal of Educational Technology*, 37(1):45–54.

Organización Internacional de Normalización. 2010. *Ergonomics of Human-system Interaction: Human-centred Design for Interactive Systems*. ISO.

Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A Reijers, et al. 2018. *Fundamentals of business process management*, volume 2. Springer.

Saul Greenberg, Sheelagh Carpendale, Nicolai Marquardt, and Bill Buxton. 2011. *Sketching user experiences: The workbook*. Elsevier.

Gregory Guy. 2013. *Words and numbers: quantitative analysis in sociolinguistics*, pages 194–210. Wiley-Blackwell.

Dongseop Kwon, Sun Kim, Chih-Hsuan Wei, Robert Leaman, and Zhiyong Lu. 2018. eztag: tagging biomedical concepts via interactive learning. *Nucleic Acids Research*, 46(W1):W523–W529.

William Labov. 2006. *The Social Stratification of English in New York City*, 2 edition. Cambridge University Press.

Travis Lowdermilk. 2013. *User-centered design: a developer's guide to building user-friendly applications*. " O'Reilly Media, Inc.".

Christian Mair. 2018. *1 .Erfolgsgeschichte Korpuslinguistik?*, pages 5–26. De Gruyter, Berlin, Boston.

Microsoft Corporation. 2024. Microsoft Excel. Computer Software.

Mariana Neves and Jurica Ševa. 2019. An extensive review of tools for manual annotation of documents. *Briefings in Bioinformatics*, 22(1):146–163.

J. Nielsen. 2012. Thinking aloud: The# 1 usability tool. Last accessed 21 February 2024.

Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. 2023. Neural machine translation for low-resource languages: A survey. *ACM Computing Surveys*, 55(11):1–37.

E Frank Ritter, D Gordon Baxter, and F Elizabeth Churchill. 2014. *Foundations for designing user-centered systems: What system designers need to know about people*. Springer.

H. Sharp, J. Preece, and Y. Rogers. 2019. *Interaction Design: Beyond Human-Computer Interaction*. Wiley.

Herbert A. Simon. 1969. *The Sciences of the Artificial*. The MIT Press.

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

Sali A Tagliamonte. 2006. *Analysing sociolinguistic variation*. Cambridge University Press.

Barbara Tversky, Masaki Suwa, Maneesh Agrawala, Julie Heiser, Chris Stolte, Pat Hanrahan, Doantam Phan, Jeff Klingner, Marie-Paule Daniel, Paul Lee, et al. 2003. Sketches for design and design of sketches. *Human Behaviour in Design: Individuals, Teams, Tools*, pages 79–86.

Robert A Virzi, Jeffrey L Sokolov, and Demetrios Karis. 1996. Usability problem identification using both low-and high-fidelity prototypes. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 236–243.

James A. Walker. 2012. *Variation in Linguistic Systems*. Routledge.

Miriam Walker, Leila Takayama, and James A Landay. 2002. High-fidelity or low-fidelity, paper or computer? choosing attributes when testing web prototypes. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 46, pages 661–665. Sage Publications Sage CA: Los Angeles, CA.

Guillaume Wisniewski, Alexis Michaud, and Séverine Guillaume. 2020. Phonemic transcription of low-resource languages: To what extent can preprocessing be automated? In *1st Joint SLTU (Spoken Language Technologies for Under-resourced languages) and CCURL (Collaboration and Computing for Under-Resourced Languages) Workshop*, pages 306–315. European Language Resources Association (ELRA).

Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 2013. WebAnno: A flexible, web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6, Sofia, Bulgaria. Association for Computational Linguistics.

Claudia Zapata and José Antonio Pow-Sang. 2012. Sample size in a heuristic evaluation of usability. *Software Engineering: Methods, Modeling, and Teaching*, 37.

Appendix

A Axial Coding Results and Thematic Relationships

Figure 3 presents a visualization of the twelve *code groups* extracted from the axial coding stage, as part of Grounded Theory methodology. This illustration is designed to enhance clarity by focusing on the most critical relationships and *code groups* from the data. In the provided context, the *variables* (colored in turquoise) signify specific linguistic features (*Dependent Variables*) or speaker attributes (*Independent Variables*).[4]

As can be seen from the figure, data management is a key challenge in research workflows, directly impacting the creation of variables and the efficiency of annotation. It contrasts manual, error-prone tasks with the potential for increased efficiency and reduced errors through automated processes, underscoring our findings that automation is a desirable, though not yet fully realized, goal in language variation research. The diagram further delineates the ripple effect of data management on research output. Effective management is shown to allow for the incorporation of more variables, which can lead to richer, more nuanced research. However, this also introduces a trade-off between the potential benefits of having more variables and the additional effort required to manage them.

B Research Interests of the Users

For our study, we interviewed four participants with different academic positions, different universities, and fields of research (Table 1). The research conducted by our participants encompasses a wide range of topics within the field of Arabic sociolinguistics, primarily focusing on how language behavior varies across different social contexts, speaker backgrounds, and geographic regions. This includes for instance the study of how individuals adapt their language in response to their surroundings and interaction partners (known as language accommodation) and the differences in speech patterns between native and second language (L2) speakers. Moreover, the research focuses on the linguistic diversity found in densely populated areas, particularly examining the variation between formal and informal Arabic, the impact of identity on language use, and the influence of regional dialects on over-regional language. For example, one of the participants explores the complex environment of Morocco's multilingual setting, focusing on the diverse facets of language that such a context presents. The participants worked mainly on phonological, morphological, and lexical features occurring in their data. From a sociolinguistic perspective, these studies shed light on the complex relationship between language, society, and identity, highlighting the diverse ways in which language functions both as a tool for communication and as a marker of cultural and individual identity. The complexity of annotating, processing, and analyzing such data underscores the need for flexible tools that can accommodate the uniqueness of each research area, as every researcher's requirements differ considerably.

ID	Academic Position	Affiliation
#1	Assistant Professor	University of Bayreuth, Germany
#2	Postdoctoral Researcher	University of Bergamo, Italy
#3	Postdoctoral Researcher	Freie Universität Berlin, Germany
#4	Ph.D. Candidate	University of Vienna, Austria

Table 1: Overview of User Study Participants by Academic Position and Affiliation.

C Further User Requirements

This section documents all identified user requirements, as well as further requirements that we will not pursue but that inspire further research.

C.1 Full List of Implementable User Requirements

The following list captures the 14 user requirements that were derived from analysing the data from the user interviews and observations. Each requirement is followed by a short description detailing the expectations for a User-Centered Design solution.

1. Data/Variable-Management-System: Enables consistent data/variable-changes
2. Digital Annotation: Digitally enhanced manual annotation
3. Ensure intuitive software usability: Interactions must be relevant and intuitive
4. Automated analysis: Automatic variable and annotation counting

[4] While extralinguistic variables are used here exclusively as predictors, it is important to note that not all linguistic variables are dependent. The basic principle of the study of variation is that linguistic context often contributes significantly to variational preferences.

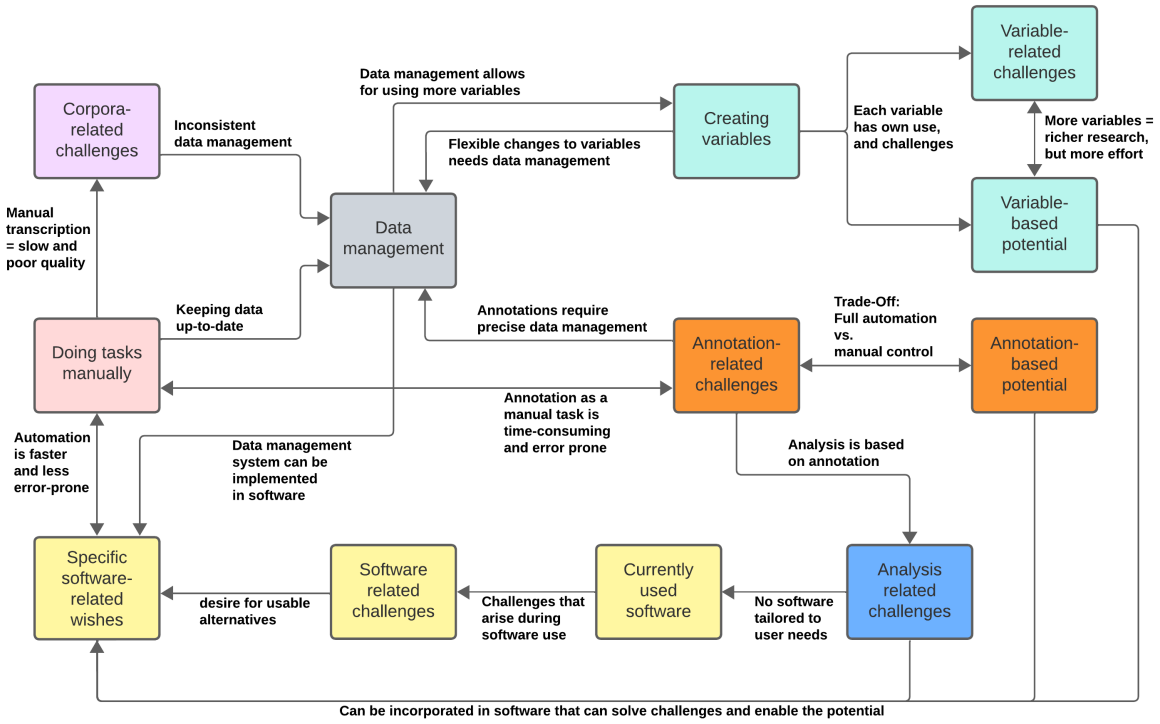

Figure 3: Illustration of *code* relationships from axial coding in Grounded Theory, focusing on data management as the core challenge in research workflows. The diagram shows its impact on variable creation, annotation efficiency, and the need for software that aligns with user needs. It highlights trade-offs between manual and automated annotation, as well as the potential for richer research through variable diversity.

5. Customizing annotation format: System detects individual annotations
6. Detect multiple annotations: Detect words with multiple annotations
7. (Partially) automatic annotation: Controlled automation of annotation process
8. Search/Highlight annotations: Enable finding annotations quickly
9. Data-Viewer: Intuitive representation of analysis results
10. Automated text-to-speaker-mapping: Detect speaker-text-correspondence
11. Corpus Management: Load and remove text files from corpus
12. Clear and intuitive navigation: Overlay that allows clear navigation
13. Corpus Exploration Section: Check whole corpus for correctness
14. Automatic variable extraction: Automatically extract data from corpus

Based on the list, Requirements 1 to 8 directly represent user needs identified during data analysis. In contrast, Requirements 9 to 14 serve as follow-up requirements, indirectly fulfilling user needs by facilitating the implementation of Requirements 1 to 8 in a technical context (for example, *10. Automated text-to-speaker mapping* enables *4. Automated analysis* by associating spoken text with speakers, thus facilitating the identification of patterns in language use).

C.2 Additional Research Directions in User Requirements

We identified additional requirements that, due to their high complexity and effort-to-benefit ratio, will not be pursued in the current project scope. Furthermore, additional user studies would be necessary to develop a sufficient design solution that fully addresses all facets of these intricate requirements. However, we documented two of them here to inform future research and highlight areas for deeper exploration.

(i) *Automatic Transcription* involves converting spoken language from audio recordings into written text. This process is traditionally labor-intensive, posing a significant time investment due to the lack of effective automation options, particularly for under-resourced languages. Despite recent advancements and growing interest in this field (Adams et al., 2019), substantial challenges (differences in phonemic inventories, phonotactic combinations, and word structure between languages, as well as limited training data for accurate transcription models) persist, as highlighted by recent research (Wisniewski et al., 2020). An intuitive and efficient design solution for automatic transcription could significantly enhance the efficiency of language variation studies by reducing manual effort and time. (ii) *Automatic and Reliable Corpus Translation* faces similar complexities, primarily relying on manual translation efforts. The challenge lies in achieving consistent and accurate translations across diverse language corpora, a task that continues to be difficult, given the complexity of linguistic variations (Ranathunga et al., 2023). Developing a design solution that ensures intuitive use, consistent processing, and reliable outcomes for corpus translation could dramatically expand the research capabilities in language variation studies, making it more accessible and less time-consuming.

D Sketches and Prototypes

While sketches and low-fidelity prototypes may appear similar initially, a difference in their purpose can be outlined. For our design process, sketching is intended for the exploration of a variety of design ideas, whereas prototyping focuses on the refinement of promising design concepts.

D.1 Sketches

Figure 4 shows an example of a sketch. It illustrates how sketches are characterized by a low level of detail and quick creation, as well as being easily disposable due to the little effort for creating them. This enables the exploration of many different design solution ideas that can be vastly different, while also allowing communication and evaluation of basic components and concepts.

D.2 Prototypes

Figure 5 portrays a (low-fidelity) prototype that is informed by the identified user requirements. It expands on earlier sketches by refining ideas and increasing the level of detail, enabling a clearer communication and evaluation, especially with target users. To incorporate functionality, a "slide-based" prototype can be employed, where each slide represents a state of the design solution (for instance, a software) by using detailed, drawn images, which

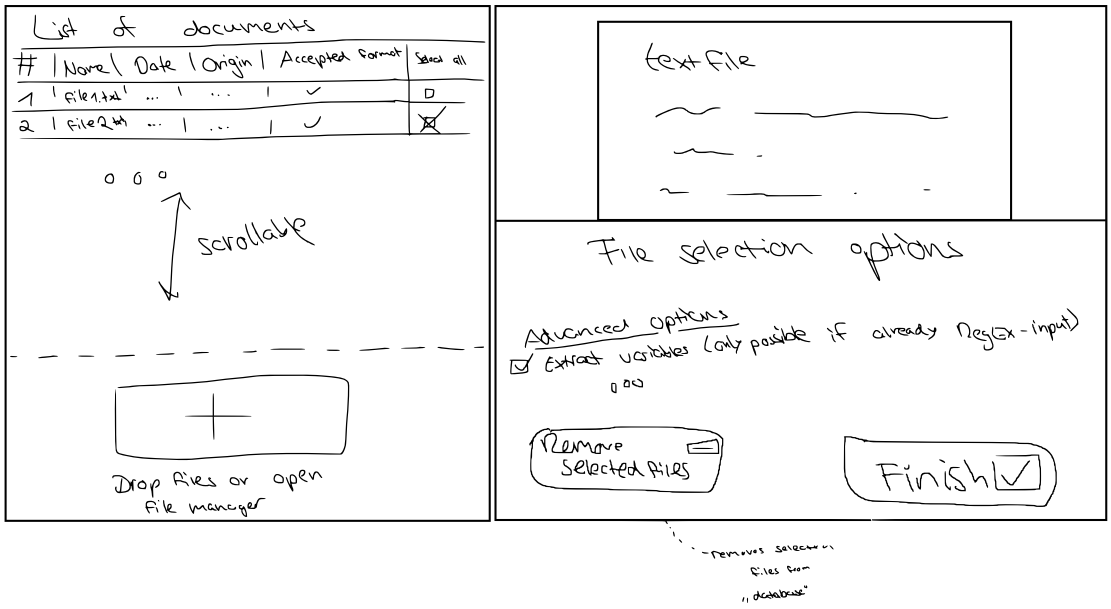

Figure 4: Illustration of a potential design solution sketch for managing corpus files, highlighting how sketching encourages the exploration of design solutions in the context of User-Centered Design.

are interconnected through linked elements in the slides.

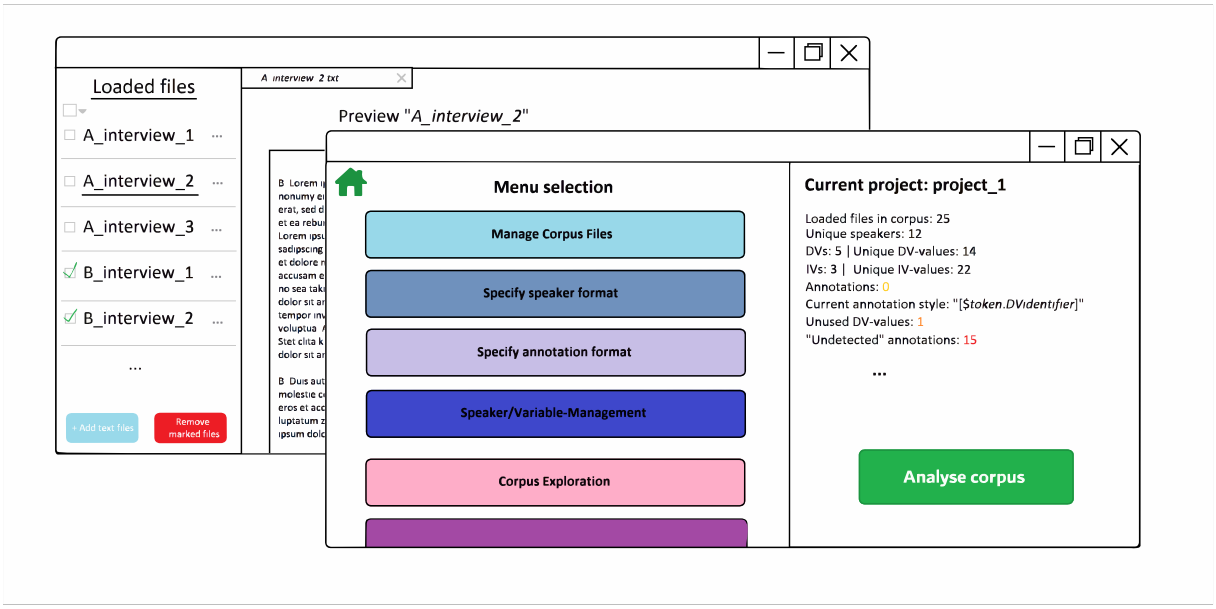

Figure 5: Refined drawing portraying a (low-fidelity) prototype, which can be used to communicate design solutions and obtain feedback during additional user studies.