SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Information Systems

Optimizing User-centered Design for CreateData4AI

Nick Mathis Hoffmann

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Information Systems

Optimizing User-centered Design for CreateData4AI

Optimierung eines nutzerzentrierten Designs für CreateData4AI

Author:	Nick Mathis Hoffmann
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Stephen Meisenbacher
Submission Date:	21.10.2025

I confirm that this master's thesis in information systems is my own work and I have documented all sources and material used.

Munich, 21 October 2025

Location, Submission Date

Author

AI Assistant Usage Disclosure

Introduction

Performing work or conducting research at the Chair of Software Engineering for Business Information Systems (sebis) at TUM often entails dynamic and multi-faceted tasks. At sebis, we promote the responsible use of *AI Assistants* in the effective and efficient completion of such work. However, in the spirit of ethical and transparent research, we require all student researchers working with sebis to disclose their usage of such assistants.

For examples of correct and incorrect AI Assistant usage, please refer to the original, unabridged version of this form, located at this link.

Use of *AI Assistants* for Research Purposes

I have used AI Assistant(s) for the purposes of my research as part of this thesis.

Yes [x] No [ ]

Explanation: GitHub Copilot (GPT-4.1 and Claude Sonnet 3.7 Thinking) was used to support the implementation of features described in this thesis, including the ideation of technical designs, root-cause analysis of defects, and the AI-based review of manually written code. ChatGPT's web search and deep research features were utilized for knowledge discovery, including exploring literature on specific niche topics, in addition to traditional non-AI methods. ChatGPT and Grammarly were used to enhance manually written text drafts in terms of grammatical correctness, fluency, conciseness, and academic tone. ChatGPT was used to assist with the formatting of LaTeX code for tables. ChatGPT was used to translate the Abstract into German language. No text was copied directly from AI tools without thorough manual review and, where necessary, further refinement.

I confirm in signing below, that I have reported all usage of AI Assistants for my research, and that the report is truthful and complete.

Munich, 21 October 2025

Location, Date

Author

Abstract

CreateData4AI (CD4AI) is a novel human-in-the-loop (HITL) data labeling system designed to assist data scientists and domain experts by making the annotation of large text corpora efficient and scalable. A prototype web application, previously developed in a lab-based environment, aims to integrate the underlying HITL engine into a fully functional and user-friendly labeling tool that supports team collaboration. Prior work in the CD4AI research project, however, has primarily focused on advancing the underlying HITL framework, with limited attention given to evaluating and improving the interface from a user-centered perspective. Therefore, this thesis investigates how the functional design and usability of the CD4AI web application can be optimized to transition it from a sandbox environment to real-world use, and how principles from Human-Computer Interaction research can guide this process. Following the User-Centered Design framework outlined in ISO 9241-210, four iterative cycles of design validation, requirements engineering, and functional implementation were conducted. Evaluation included internal design inspection and external user testing with 28 participants, and usability was assessed using the System Usability Scale (SUS). The analysis resulted in 37 technical adaptations and a final SUS score of 72.8, indicating the above-average usability of CD4AI. Differences between technical users (SUS 80.0) and non-technical users (SUS 58.5) highlight distinct user personas and the need for tailored support. Future research should examine the system's effectiveness in authentic, real-world usage scenarios and further enhance usability for non-technical users, particularly in terms of workflow intuitiveness and learnability. This study contributes to the democratization of Artificial Intelligence by advancing CD4AI from a laboratory prototype to a mature system and discusses design principles that may generalize to other HITL applications

Kurzfassung

CreateData4AI (CD4AI) ist ein innovatives Human-in-the-Loop-(HITL)-Data-Labeling-System, das Datenwissenschaftler:innen und Domain-Expert:innen unterstützt, indem es die Annotation großer Textkorpora effizient und skalierbar gestaltet. Eine prototypische Web-Applikation, die bisher in einer laborbasierten Umgebung entwickelt wurde, zielt darauf ab, die zugrunde liegende HITL-Engine in ein voll funktionsfähiges, nutzerfreundliches Labeling-Tool zu integrieren, das kollaboratives Arbeiten im Team ermöglicht. Bisherige Arbeiten im CD4AI-Forschungsprojekt konzentrierten sich primär auf die Weiterentwicklung des HITL-Frameworks, während die Evaluation und Optimierung der User Interface aus einer nutzerzentrierten Perspektive nur begrenzt berücksichtigt wurde. Vor diesem Hintergrund untersucht diese Arbeit, wie das funktionale Design und die Usability der CD4AI-Webapplikation optimiert werden können, um den Übergang von einer Sandbox-Umgebung in reale Einsatzszenarien zu ermöglichen, und inwiefern Prinzipien der Human-Computer-Interaction-Forschung diesen Prozess unterstützen können. Unter Anwendung des User-Centered-Design-Frameworks gemäß ISO 9241-210 wurden vier iterative Zyklen aus Design-Validation, Requirements Engineering und funktionaler Implementierung durchgeführt. Die Evaluation umfasste sowohl interne Design-Inspektionen als auch externe User-Tests mit 28 Teilnehmer:innen; die Usability wurde mithilfe der System Usability Scale (SUS) gemessen. Die Analyse führte zu 37 technischen Anpassungen und einem finalen SUS-Wert von 72,8, was die überdurchschnittliche Usability von CD4AI bestätigt. Unterschiede zwischen technischen Nutzern (SUS 80,0) und nicht-technischen Nutzern (SUS 58,5) unterstreichen die Existenz unterschiedlicher User Personas und den Bedarf an maßgeschneiderter Unterstützung. Zukünftige Forschung sollte die Effektivität des Systems in realen, authentischen Nutzungsszenarien evaluieren und die Usability für nicht-technische Nutzer:innen weiter verbessern, insbesondere hinsichtlich Workflow-Intuitivität und Learnability. Diese Arbeit leistet einen Beitrag zur Demokratisierung von Artificial Intelligence, indem sie CD4AI von einem Laborprototyp zu einem ausgereiften System weiterentwickelt, und diskutiert Designprinzipien, die potenziell auf andere HITL-Anwendungen übertragbar sind.

Contents

1. Introduction

Every day, large amounts of unstructured and unlabeled text are generated. Making this data usable for Artificial Intelligence (AI) applications requires careful annotation, yet manual labeling is slow, labor-intensive, and difficult to scale. Dependence on expert annotation is particularly critical in specialized domains, creating a major bottleneck that limits the speed, efficiency, and accessibility of AI development. This challenge is especially pressing for organizations with limited AI resources, such as small and medium-sized companies.

The CreateData4AI project (***C**ontext **R**ule **E**mbedding-Assisted **A**nnotation of **T**extual **Data for AI** Applications*), or *CD4AI*, addresses these challenges by introducing a hybrid human-in-the-loop (HITL) framework. Instead of manually labeling each data point individually, the framework facilitates collaboration between a human expert and Natural Language Processing (NLP) techniques to iteratively define so-called *context rules*, which capture the semantic meaning and intent of class labels. Once defined, these context rules are automatically applied to the entire text corpus, enabling scalable and efficient data annotation. The framework is implemented as a pipeline-based NLP engine. User interaction with the NLP engine is facilitated through the CD4AI prototype web application, which is designed to make the labeling process intuitive and user-friendly. The web application further provides functionalities for project organization, data import and export, team collaboration, and workflow management, thereby supporting an integrated and efficient annotation process.

Prior research on CD4AI mainly focused on advancements of the HITL framework. The prototype web application, however, was developed under laboratory conditions, with little attention given to evaluating and improving the interface from a user-centered perspective. Additionally, recent scientific developments from research on the HITL framework have not yet been incorporated into the system. Considering the goal of a public deployment and open-source release, the system currently lacks the maturity required for real-world use. The Technology Acceptance Model (TAM) posits that users' intention to adopt a system is primarily influenced by its perceived usefulness and ease of use [1]. Research further emphasizes that AI system usability is crucial for adoption and user trust, and that many systems fail when algorithmic performance is prioritized over human-centered design [2, 3]. Building on these insights, this thesis addresses the following research questions (RQs).

RQ1: *What efforts can be taken to reduce manual effort in the CD4AI web application?*
Manual effort is a particular usability concern in the labeling workflow, given the project's goal of making annotation as efficient as possible. Since minimizing user workload is central to CD4AI's philosophy, the first research objective is to analyze inefficiencies in the current design and explore strategies that further support users. Improvements may occur at both the overall workflow level and within specific user interactions.

RQ2: *How can the CD4AI web application be iteratively optimized for usability?*
In addition to reducing manual effort, this thesis aims to systematically improve the overall usability of the CD4AI application. Usability encompasses not only the efficiency and effectiveness of task completion but also the learnability and satisfaction experienced by users. Ensuring that users can independently understand and operate the system is crucial for successful adoption. To achieve this, established usability evaluation methods will be employed, combining heuristic analysis and user testing in both in-person and remote settings. Through iterative cycles of validation, ideation, and improvement for both interface and functionality, the objective is to reach above-average usability, measured by the System Usability Scale (SUS) [4].

RQ3: *What technical adaptations are required to transform the CD4AI application from a research prototype into an externally usable system?*
The current prototype lacks several essential features for public deployment. Many mechanisms exist only as mock implementations and are not yet ready for production. This primarily concerns accessibility, scalability, and the integration of recent advancements that underlie the HITL framework. The third research objective is therefore to identify and implement the technical adaptations necessary to transition CD4AI into a stable and usable system.

These research questions are addressed through the application of the User-Centered Design (UCD) process, as defined in ISO 9241-210 [5]. The UCD framework, established in Human-Computer Interaction (HCI) research, provides an iterative methodology for developing systems that prioritize user needs throughout the design cycle. Over four UCD iterations, established evaluation methods were applied to identify usability flaws, implement requirements, and assess improvements. The flexibility and adaptability of this process make it suitable for addressing all three research questions.

This thesis makes key contributions to the CD4AI research project, advancing the usability, maturity, and practical applicability of the system. The web application was systematically developed into a user-centered system, based on 37 targeted adaptations implemented through iterative evaluation and refinement. Above-average usability was attested, as reflected in a SUS score of 72.8, with technical users rating the system even higher (SUS 80.0). The system was advanced from a research prototype to a mature, externally usable application. Furthermore, functional and methodological recommendations were derived from this work, providing guidance for future development and supporting the broader adoption of CD4AI in real-world environments.

2. Background

The evaluation and further development of the CreateData4AI web application take place at the intersection of technical system design and usability research. A brief overview of the project's scientific background and architectural principles, together with relevant foundations from Human-Computer Interaction, helps establish the basis for the methodological approach of this thesis.

2.1. CreateData4AI (CD4AI)

The CreateData4AI (CD4AI) research project introduces a hybrid framework designed to reduce the time and effort required for generating labeled datasets from unstructured text corpora. The framework is operationalized through a web-based application that serves as an accessible interface to the underlying process.

2.1.1. Foundations of the CD4AI Framework

The CD4AI framework consists of a hybrid multi-stage *human-in-the-loop* pipeline that integrates domain expertise with state-of-the-art Natural Language Processing (NLP) techniques. Human-in-the-loop (HITL) describes systems in which human input is integrated into the training, evaluation, or operation of AI models, allowing iterative refinement and improved reliability [3].

The proposed pipeline consists of four interdependent components: keyword extraction, context window extraction, context rule creation, and extrapolation. Each stage contributes to the gradual transformation of unstructured text into structured, labeled data.

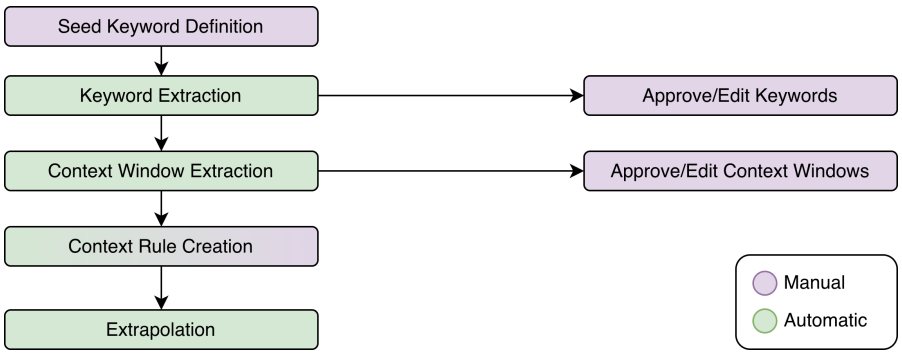

Figure 2.1.: The human-in-the-loop CD4AI pipeline.

The proposed pipeline takes a corpus of unstructured text documents and a set of classes or categories defined by a domain expert as input. To obtain a dataset of fully labeled documents, the CD4AI pipeline must be performed for each class separately.

The first stage involves the identification of class-specific *keywords* that are representative of each class's semantic characteristics. In this context, Meisenbacher, Schopf, Yan, et al. [6] have proposed a dedicated class-specific keyword extraction approach by extending the widely used KeyBERT[1] framework. The method accepts a predefined set of class-specific seed keywords and iteratively expands this set based on a scoring scheme that measures semantic similarity between the KeyBERT-extracted candidates and the seed terms. The quality and scope of the resulting keyword set can be adjusted by specifying a percentile threshold for inclusion, the maximum number of new keywords to add per iteration, and the total number of iterations. A subsequent review and filtering of keywords by a domain expert ensures consistent quality in the loop.

Building upon the extracted keywords and keyphrases, the next stage extracts surrounding text segments, or *context windows*, from the focal text corpus. These segments capture the linguistic environment in which these terms are used. The domain expert filters out context windows in which the focal keyword is not relevant to the class. Thus, the remaining context windows serve as candidate examples illustrating how a concept is expressed in context and form the basis for subsequent interpretation and rule definition.

In the next stage, *context rules* are derived from the selected context windows. These rules embed domain-specific understanding into the data transformation process, effectively bridging the gap between purely statistical NLP methods and expert-driven semantic interpretation. While the concept of context rules itself is abstract and can be implemented using various methods, recent approaches in the CD4AI project rely on a semantic archetype-driven framework. In this approach, the relevant context windows are first clustered semantically. Using a Large Language Model (LLM), these clusters are then distilled into archetypes, i.e., short and concise summarizing phrases that capture the semantic essence of the class.

Finally, during the extrapolation phase, the system applies the derived context rules (archetypes) to larger text corpora. Leveraging modern NLP models, such as embedding-based similarity measures, enables automatic classification and annotation of documents.

Through this structured process, CD4AI enables the creation of interpretable and reproducible datasets that directly link abstract class definitions to concrete text segments. The resulting datasets can serve as reliable inputs for downstream AI applications such as model training, information retrieval, or knowledge discovery.

2.1.2. Core User Journey of the CD4AI Web App

The CD4AI web application is accessed through a standard web browser. Users register and log in using an existing GitLab account, as authentication is handled via the OAuth[2] protocol. Figure 2.2 illustrates the core workflow followed by a newly registered user to create a labeled dataset in CD4AI.

[1] `https://github.com/MaartenGr/KeyBERT`
[2] `https://oauth.net/2/`

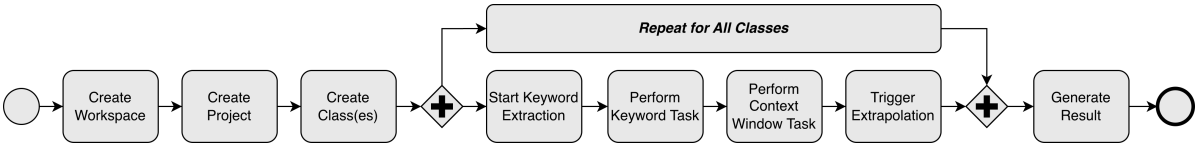

Figure 2.2.: Business process model of the initial CD4AI core user journey.

The process begins with setting up a workspace, which involves providing a name and description to organize subsequent projects. Within the workspace, the user creates a project by specifying its name, description, a raw data file (in `.csv` format uploaded from the local device), the data language (English or German), and the embedding model (either *sentence-transformers/all-MiniLM-L6-v2* for English or *deutsche-telekom/gbert-large-paraphrase-cosine* for German). The user can invite collaborators to workspaces and projects via the *Invitees* tab and assign them one of three roles (admin, contributor, or viewer). Workspace invitees have access to all projects within the workspace. Users can access the workspaces and projects they are invited to via respective tabs on the CD4AI dashboard. Each project requires at least one class, defined by a class name and description. All workspaces, projects, and classes are hierarchically organized and can be edited (including name, description, and project files) and deleted.

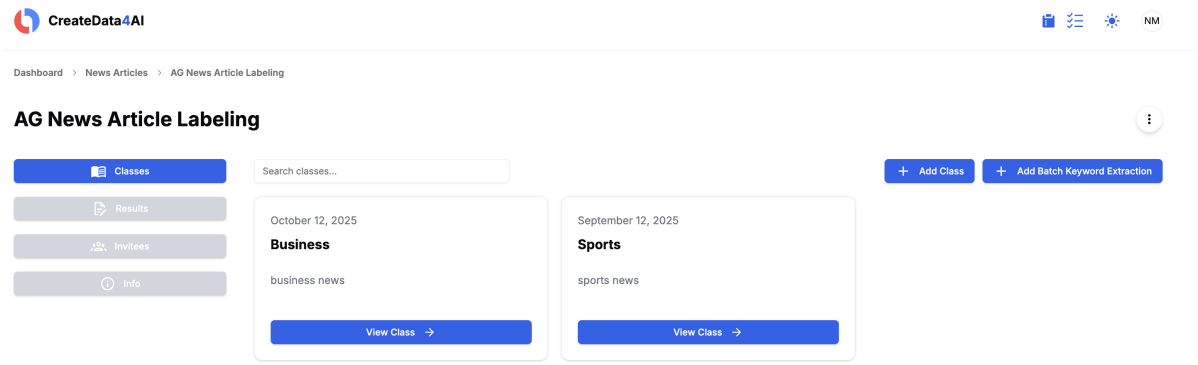

Figure 2.3.: Initial CD4AI project dashboard.

Labeled datasets are created at the project level. To create a labeled dataset, the user has to proceed through the CD4AI HITL pipeline for each of the project's classes, as outlined in Section 2.1.1: initiating keyword (KW) extraction, completing the KW task, conducting the context window task, and triggering extrapolation. In fact, the CD4AI prototype does not yet include the concept of context rules.

Class-specific KW extraction can be initiated on the respective class's dashboard page or from the project page. The latter offers a batch-creation feature, where KW extractions for one or more classes can be started simultaneously (see Figure 2.5). The batch-creation option is designed to provide a smoother and potentially faster way to initiate tasks, utilizing the same algorithm as if a KW extraction were performed for a single class alone.

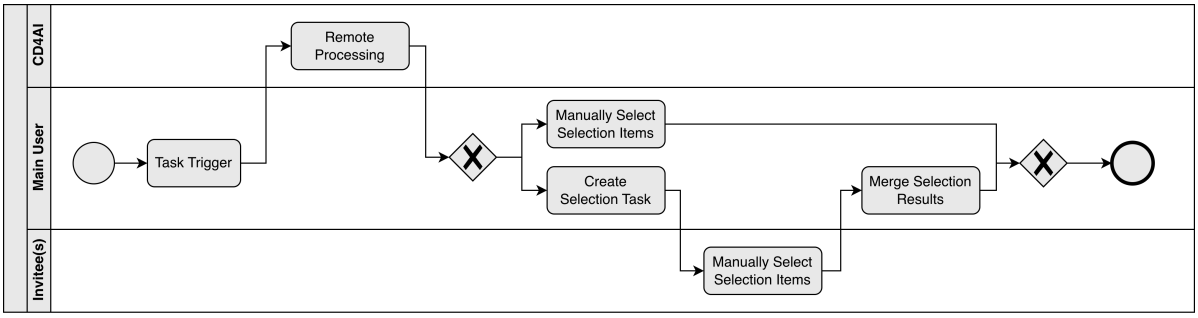

Figure 2.4.: Business process model illustrating the mechanism of tasks in CD4AI.

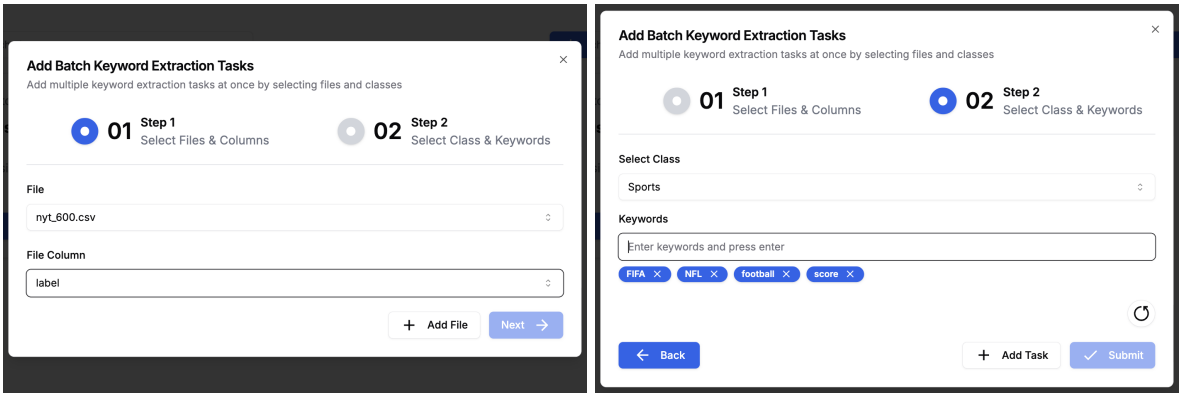

Figure 2.5.: Users can create multiple keyword tasks in a batch. Step one (left): the user selects the file-column combination that stores the raw data. Step two (right): the user defines initial seed keywords for a specific class.

For KW evaluation, CD4AI provides a review panel (see Figure 2.6), where all extracted KWs can be categorized as *not relevant* (default, left side) or *relevant* (right side). Several action buttons support the user in that process by providing options to search, sort, and batch-select KWs.

Context windows (CW) are evaluated one by one by moving the displayed CW to the left (reject) or right (accept), as shown in Figure 2.7. A tracker in the middle of the screen indicates the number of remaining CWs to be classified. After ten CWs have been accepted and ten rejected, the *Auto-Assign* feature is activated. When triggered by the user, the function automatically reviews all remaining CWs by computing their semantic similarity (cosine similarity between the respective CW embeddings) to those that have already been classified. Each unclassified CW is then assigned to the same category as a previously labeled CW, provided that the similarity exceeds a user-defined threshold. Technically, it is not necessary to classify all CWs to proceed to the extrapolation.

Both the KW and the CW task panels have several features in common. The task status (i.e., *in progress*, *completed*, *failed*) is indicated via an icon and a banner at the top of the panel. Tasks are assigned a default name, which can be renamed by clicking an icon button at the

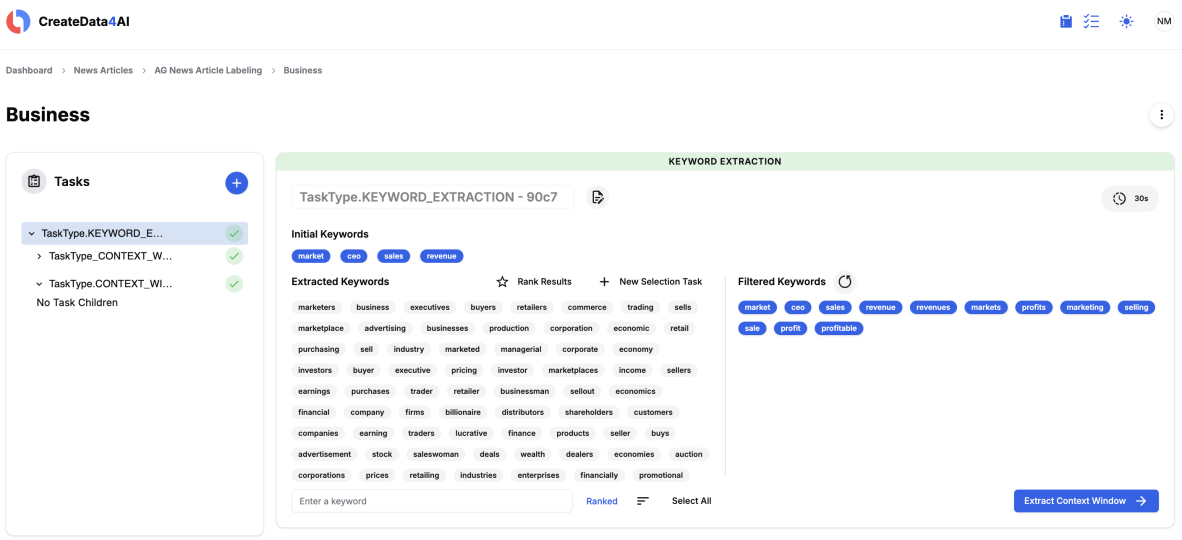

Figure 2.6.: The keyword task panel enables the user to review extracted keywords and select relevant ones.

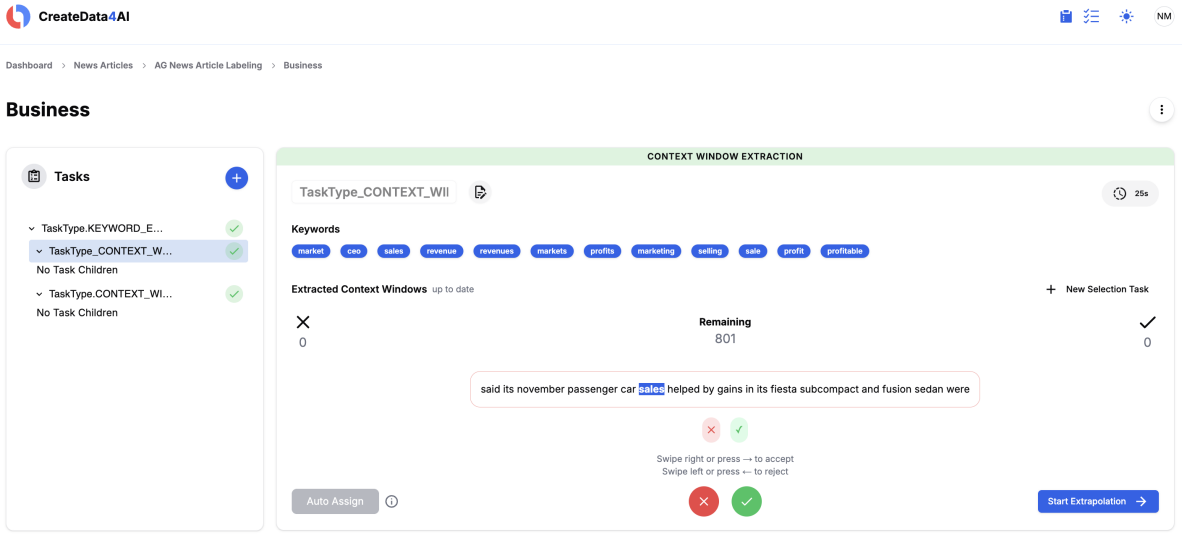

Figure 2.7.: The context window task panel enables the user to review extracted context windows and judge their relevance.

top of the panel. A task timer indicates the duration of the extraction. The task input (i.e., the initial user-defined KWs for a KW extraction, or the input KWs for the CW extraction) is displayed right below the task name and above the action panel. The option to save the results and proceed with the next task is placed in the bottom right corner. Starting the next task creates a new task object, which appears as a task child in the class's task tree on the left side of the screen.

Both task types can also be completed collaboratively via a so-called *selection task*, which can be initiated by clicking the button labeled *New Selection Task*. In this case, the extracted items are divided into subsets assigned to a user-defined group of workspace or project collaborators, each of whom independently assesses their subset. Once such a selection task is created, the button mentioned above is replaced with an analogous button that directs the user to the selection task dashboard, where the task can be managed (such as assigning new collaborators, tracking progress, and merging the results). Invited users can access and complete their assigned selection task via the *ToDo* icon in the global app header bar, located at the top right of the screen. Once all collaborators have submitted their results, the creator can merge the corresponding batches into a unified outcome.

Key to CD4AI's design is the repeatable character of the task flow. Even after dataset creation, KW and CW selections can be modified, allowing for the start and performance of multiple KW, CW, and extrapolation tasks based on different task inputs. This enables users to experiment with CD4AI by evaluating the extrapolation results and iteratively exploring the effect of varying task inputs on labeling quality.

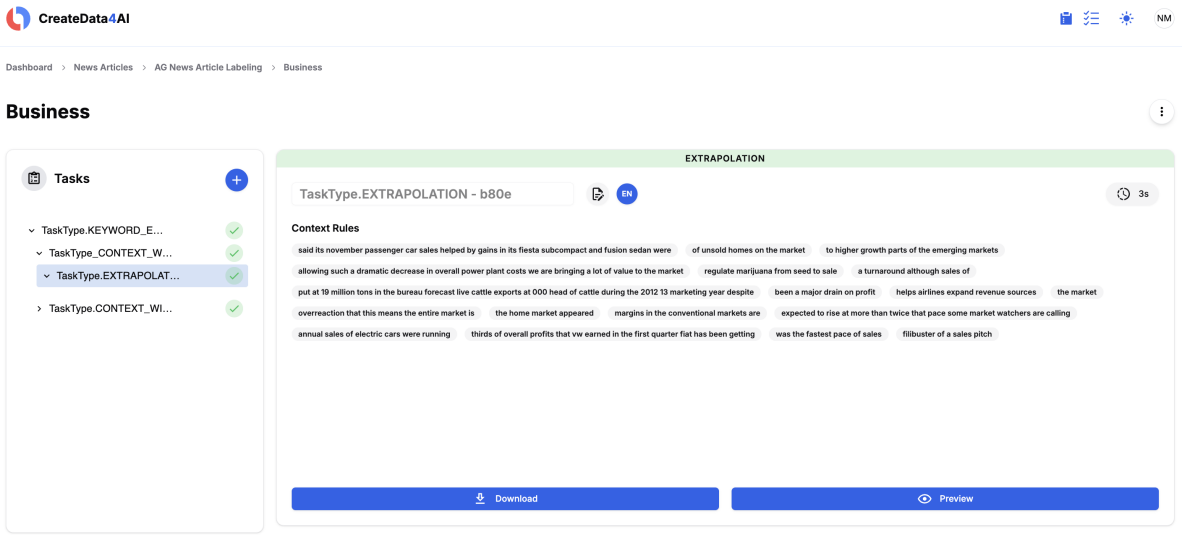

Figure 2.8.: Context rules have not yet been implemented in CD4AI. In its current form, the extrapolation task relies on the relevant context windows directly as context rules.

At present, the CD4AI application does not include the concept of context rules. Consequently, users move directly to the extrapolation step, which is currently realized only as a mock implementation. The mock works by assigning a class label C to all data points (i.e., documents) that contain any CW that was marked as *relevant* for class C. The extrapolated results from individual classes can then be combined to generate a labeled dataset, either from all classes or a selected subset. After at least one successful extrapolation, users proceed by generating a result. CD4AI aggregates all extrapolation results into a single file. The resulting file can be previewed or downloaded.

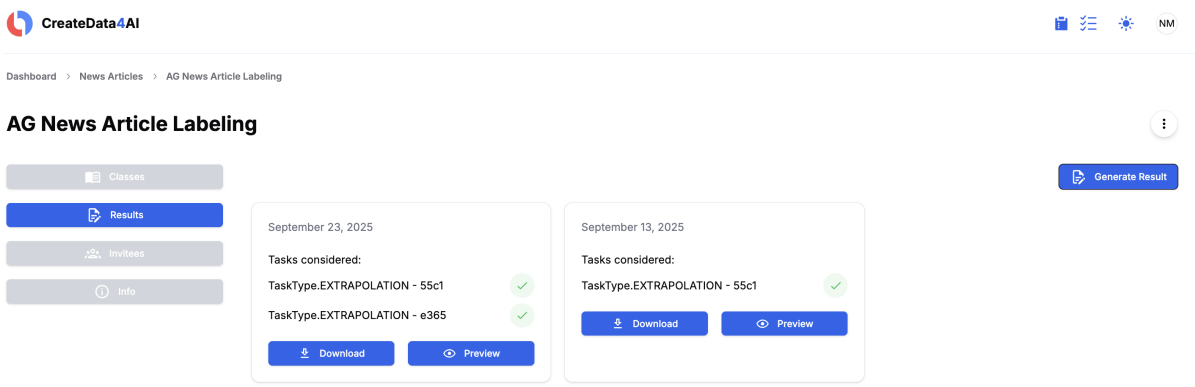

Figure 2.9.: Users can generate labeled datasets based on class-specific context rules.

2.1.3. Technical Architecture of the CD4AI Web App

The CD4AI web application is implemented following a frontend–backend paradigm using a modern technology stack, as illustrated in Figure 2.10. Authentication is managed via the OAuth protocol, and no custom registration or password handling is implemented within CD4AI. The backend service consists of a single REST API with two primary responsibilities. First, the API performs standard create, read, update, and delete (CRUD) operations on business objects and persists data in the database. Second, it provides the NLP engine, which implements the CD4AI pipeline presented in Section 2.1. Keyword extraction is based on a modified version of the KeyBERT framework, as proposed by Meisenbacher, Schopf, Yan, et al. [6].

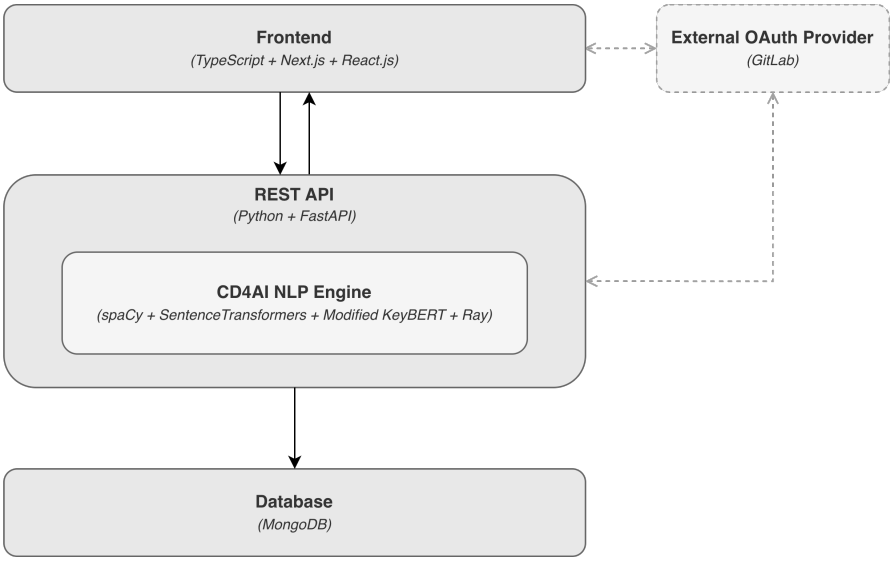

Figure 2.10.: High-level technical architecture of the CD4AI web application.

Data is stored in a non-relational document database implemented as a locally running MongoDB service. Figure 2.11 presents the conceptual data model of CD4AI. As it is typical for document-based structures, several attributes use custom document-like types. *AccountSettings* stores user preferences. In its current form, it contains only a user's preferred default embedding model for each language. *ProjectFile* stores the file path to an uploaded raw dataset in `.csv` format, together with relevant metadata. *TaskType* is an enumeration defining the task category (keyword task, context window task, or extrapolation). The *LanguageModel* attribute in the Task document specifies which embedding model should be used for processing. *TaskInput* holds the input data for a given task, with a structure that depends on the task type: keyword tasks store seed keywords, context window tasks store keywords, and extrapolation tasks store context windows. Additionally, a reference to the corresponding input data file is stored in this object. Similarly, *TaskResult* stores the type-specific output, such as the extracted or user-selected keywords and context windows. The *SelectionTask* and *SubSelectionTask* documents implement the collaboration logic. Within *SelectionTask*, *Invitee* represents an invited user, while *SplitOption* and *MergeOption* define how data should be split and recombined in collaborative task splits. A *SubSelectionTask* represents the assigned portion of selection items for one invitee. Two enumeration attributes indicate the progress of selection tasks: *SelectionTaskStatus* represents the overall task status (in progress or merged), and *SubSelectionTaskStatus* indicates the status of individual collaborators (in progress or submitted).

2.2. Human-Computer Interaction and Usability Research

Human–Computer Interaction (HCI) is the interdisciplinary study of how people engage with digital systems and how those systems can be designed to support human goals effectively. The principles and concepts of this research field provide guidelines on how to conduct user-centered design in practice.

2.2.1. Principles of Human-Computer Interaction

Originating from the convergence of computer science and cognitive psychology, HCI focuses on optimizing the dialogue between users and technology to make interaction efficient, intuitive, and meaningful. From its foundations, HCI has drawn on psychological insights, particularly concerning how people perceive, interpret, and act on information [7, 8]. Users rely on mental models, which are internal representations of how systems work, to anticipate outcomes and guide their actions [8]. Interfaces that emphasize recognition over recall, provide clear feedback, and offer consistent mappings help reduce cognitive effort and support smooth task performance [8]. These principles illustrate how understanding human cognitive constraints forms the theoretical basis for practical design guidelines in HCI.

Building on these cognitive foundations, HCI research has formulated a set of core design principles that translate human capabilities into actionable design practices. Among the most influential are the principles articulated by Norman [8], which emphasize that good

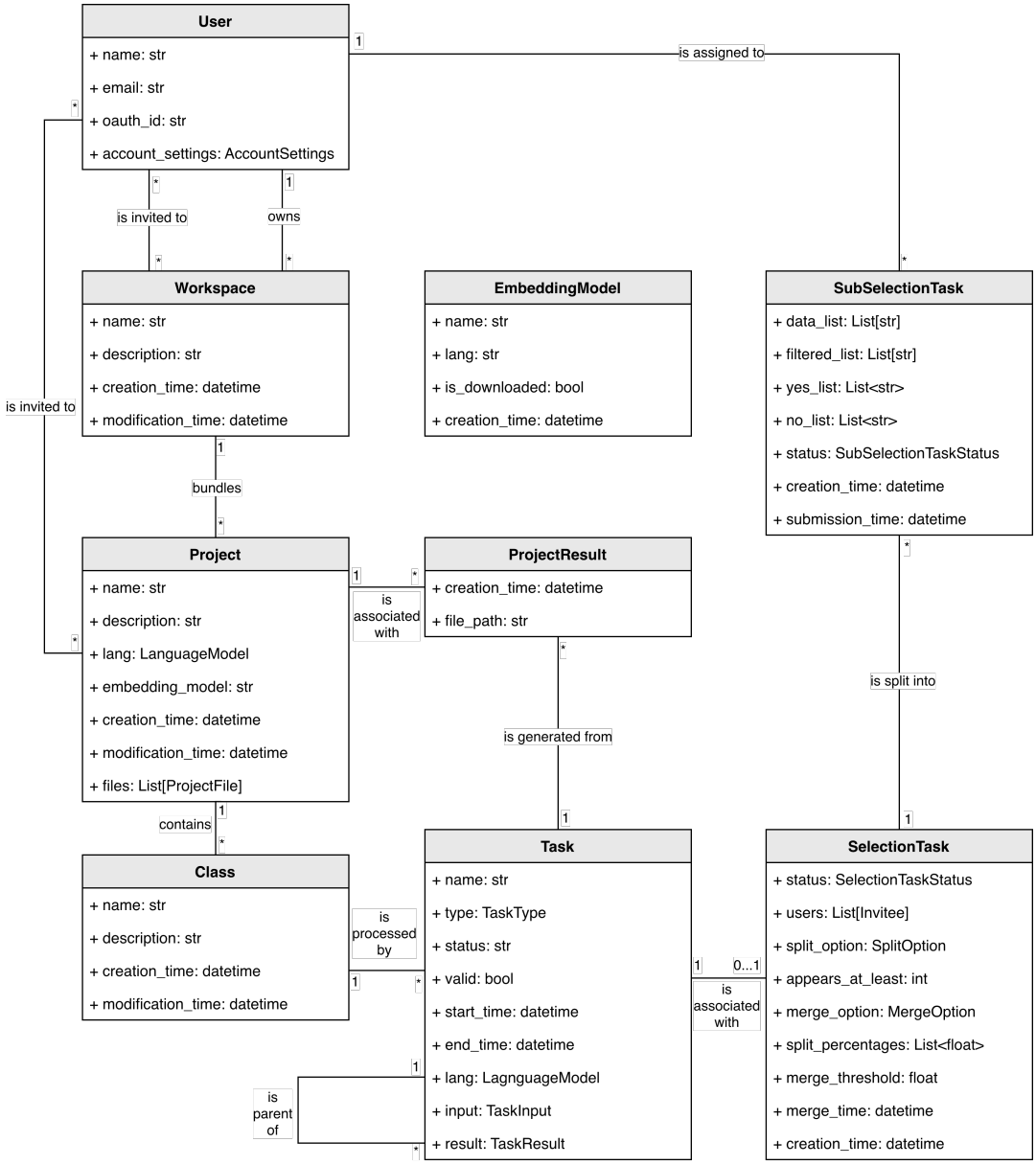

Figure 2.11.: Conceptual data model of CD4AI.

design makes possible actions visible and system responses understandable. The concept of *Discoverability and Affordances* postulates that users should be able to infer what actions are possible by perceiving the system's available functions. Clear affordances and signifiers communicate how an element can be interacted with, reducing uncertainty and guesswork. The idea of *Feedback and Visibility of System Status* postulates that user action should produce immediate and informative feedback. Visibility of system status keeps users aware of progress and helps them evaluate whether their goals are being met. *Consistency and Mapping* refers

to consistent language, layout, and interaction patterns that allow users to transfer prior knowledge across contexts. Good mapping ensures a natural relationship between controls and their effects, reducing learning time and error. The concept of *Constraints and Error Prevention* covers the idea that effective design limits the possibility of incorrect actions through physical, semantic, or logical constraints. When errors occur, systems should support recovery through undo functions or clear corrective guidance. The principle of *Conceptual Models and Simplicity* means that interfaces should help users form coherent mental models of how the system works. Simplifying complex functionality and maintaining conceptual clarity increases predictability and user confidence [8].

Building on these principles, HCI researchers have developed heuristics and design patterns as practical tools that capture invariant solutions to recurring design problems [7]. Among the most widely cited are Nielsen's 10 Usability Heuristics, as shown in detail in Table 2.1, which offer practical guidelines for evaluating and improving interface design [9]. Besides that, other influential frameworks have contributed to defining general principles of human–system interaction. Shneiderman's Eight Golden Rules [10] represent an earlier, design-oriented set of guidelines emphasizing consistency, feedback, and user control in graphical interfaces. The ISO 9241-110 standard [11] formalizes seven high-level dialogue principles, such as suitability for the task, self-descriptiveness, and controllability, which are intended for broad application across interactive systems. All three frameworks share a common cognitive and human-centered foundation.

While classical HCI principles focus on usability and cognitive alignment, the rise of AI-driven systems introduces new concerns about shared control, explainability, and ethical interaction, leading to extended design frameworks for intelligent interfaces. Horvitz [13] defined design guidelines for so-called *mixed-initiative* user interfaces, i.e., interfaces where tasks can be either performed by autonomously acting *agents*, or by the human. The guidelines propose principles for balancing automation and user control by reasoning about uncertainty, attention, and context, enabling systems to act autonomously when appropriate while remaining responsive and deferential to the user. Amershi, Cakmak, Knox, and Kulesza [3] emphasized the importance of studying real users to understand their needs, behaviors, and challenges when engaging with interactive machine learning systems. Greater user control and system transparency can empower users, but must be carefully evaluated to ensure they truly support user goals and lead to more effective human–machine collaboration [3]. Amershi, Weld, Vorvoreanu, et al. [14] have identified 18 guidelines for the design of AI applications. The guidelines, for example, provide recommendations on how to design system behavior during use interaction, how to deal with AI model bias, or how to maintain and refine the system over time. Pailian and Li [15] identified guidelines that concern the system design, emphasizing the respect of ethical principles, building a relationship between the user and AI, and how to build or restore human trust in the AI system. Margetis, Ntoa, Antona, and Stephanidis [16] proposed a methodological framework on how UX can be systematically integrated into the development process of AI applications.

Table 2.1.: Nielsen's 10 Usability Heuristics [9] with original descriptions from Nielsen [12]

No.	Heuristic	Description
1	Visibility of System Status	"The system should always keep users informed about what is going on, through appropriate feedback within reasonable time."
2	Match Between System and the Real World	"The system should speak the users' language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms."
3	User Control and Freedom	"Users often choose system functions by mistake and need a clearly marked "emergency exit" to leave the unwanted state without an extended process."
4	Consistency and Standards	"Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions."
5	Error Prevention	"Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. Either eliminate error-prone conditions, or check for them and present users with a confirmation option before they commit to the action."
6	Recognition Rather Than Recall	"Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed."
7	Flexibility and Efficiency of Use	"Shortcuts — hidden from novice users — may speed up the interaction for the expert user so that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions."
8	Aesthetic and Minimalist Design	"Interfaces should not contain information that is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility."
9	Help Users Recognize, Diagnose, and Recover from Errors	"Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution."
10	Help and Documentation	"It is best if the system does not need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks."

2.2.2. Usability and User Experience

According to ISO 9241-11, *usability* is defined as the "extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" [17]. This definition holds two key observations. On the one hand, usability is not absolute, but rather specific to the targeted user group with specific goals, in a specific environment, the so-called *Context of Use* (more on this in Section 2.2.3). Second, usability can be sliced into and analyzed in three subcategories. *Effectiveness* refers to the "accuracy and completeness with which users achieve specified goals". *Efficiency* refers to the "resources used in relation to the results achieved", such as time, human effort, costs and materials. *Satisfaction* is the "extent to which the user's physical, cognitive and emotional responses that result from the use of a system, product or service meet the user's needs and expectations" [17].

Another widespread definition of usability [18] breaks the idea down into a total of five more fine-granular characteristics: *Learnability* captures how quickly new users can begin to interact with the system effectively. *Efficiency* in this definition relates to how productively users can perform tasks once they are familiar with the interface. *Memorability* concerns the ease with which occasional users can return to the system and resume use without needing extensive relearning. A *low error rate* is also essential. Systems should minimize user mistakes, ensure that errors are recoverable, and prevent critical failures. Finally, *satisfaction* in this definition reflects the user's overall comfort and positive engagement with the system [18, 19]. Many other usability definitions, despite having slight differences, propose similar extensions, especially regarding the aspect of learnability [20].

The ISO 9241-210 standard defines the concept of *user experience* (UX), which is not the same as usability, but also is a popular term in HCI. In fact, UX has a broader meaning and refers to the "user's perceptions and responses that result from the use and/or anticipated use of a system, product or service" [5]. Furthermore, UX "is a consequence of brand image, presentation, functionality, system performance, interactive behavior, and assistive capabilities of a system, product or service" [5].

2.2.3. The User-Centered Design Framework

User-Centered Design (UCD), as defined in ISO 9241-210 [5], is a systematic approach that ensures systems are designed around users' needs, tasks, and contexts to achieve usability and positive user experience. The historical foundations of the framework are laid out in early HCI research [21, 8].

The UCD approach is grounded in six key principles. First, design decisions must be based on a clear and explicit understanding of users, their tasks, and the environments in which they operate. Second, users should not only be considered but also actively involved in the design and development process. Third, design must be driven and continuously refined by user-centered evaluation. Fourth, the process is inherently iterative, meaning that insights from evaluation inform successive design cycles. Fifth, the scope of design should extend to the overall user experience. Ultimately, effective user-centered design necessitates a

multidisciplinary team, ensuring that diverse expertise and perspectives are integrated into the process [5].

The UCD process is typically described as a sequence of iterative activities, beginning with the analysis of the Context of Use, followed by specification of requirements, production of design solutions, and iterative evaluation (Figure 2.12). These stages should not be understood as a linear sequence but rather as a cycle in which evaluation plays a central role and informs all preceding activities.

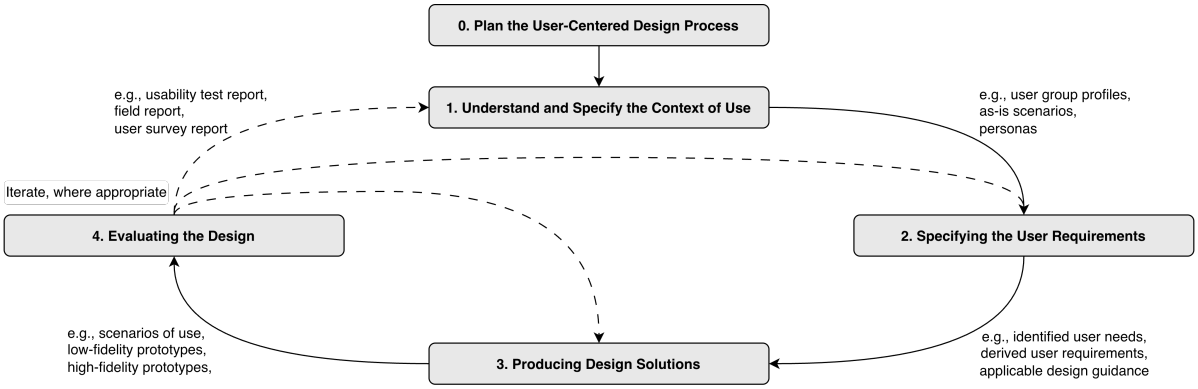

Figure 2.12.: Activities in the User-Centered Design process, as proposed in ISO 9241-210. [5]

Step 1: Understanding and Specifying the Context of Use. A central starting point in UCD is the thorough specification of the so-called *Context of Use* of the system. This requires identifying the relevant user and stakeholder groups, as well as understanding their key goals, constraints, and interrelationships. Particular attention is paid to the characteristics of the user groups, including prior knowledge, skills, physical and cognitive capabilities, preferences, and accessibility needs. Beyond user characteristics, designers must analyze the users' goals and tasks. This includes how tasks are typically performed, their frequency and duration, as well as any associated risks or potential consequences if tasks are completed incorrectly. Part of the Context of Use is also the environment in which the system is deployed. This encompasses the technical environment (hardware, software, and infrastructure), as well as the physical, social, and cultural settings. Factors such as lighting, workspace ergonomics, organizational practices, and attitudes can significantly affect usability and adoption [5]. Maguire [22] proposed a systematic template that supports system designers to define the Context of Use.

Step 2: Specifying User Requirements. The specification of user requirements involves identifying and documenting user and stakeholder needs within the intended Context of Use, ensuring alignment with organizational objectives. These requirements should include usability goals, contextual constraints, and measurable performance criteria, serving as a foundation for system design and evaluation. Potential conflicts between requirements

must be resolved transparently, and the specification should remain testable, consistent, and regularly updated throughout the project lifecycle [5].

Step 3: Producing Design Solutions. Based on the contextual analysis, design solutions are developed. These may range from low-fidelity sketches to high-fidelity prototypes and are gradually refined over multiple iterations. Importantly, these solutions remain provisional until validated through user-centered evaluation, which ensures alignment with user needs and contexts of use [5].

Step 4: Evaluating the Design. Evaluation is the cornerstone of the UCD process. It provides empirical evidence of how well a design supports intended tasks and user needs, driving iterative improvement [5]. Methods of evaluation can broadly be divided into *inspection methods* and *test methods*.

Table 2.2.: Comparison of usability evaluation techniques (adapted from Holzinger [19]).

Required Resources	**Inspection Methods**			**Test Methods**		
	Heuristic Evaluation	**Cognitive Walkthrough**	**Action Analysis**	**Thinking Aloud**	**Field Observation**	**Question-naires**
Time	low	medium	high	high	medium	low
Users	none	none	none	3+	20+	30+
Evaluators	3+	3+	1–2	1	1+	1
Equipment	low	low	low	high	medium	low
Expertise	medium	high	high	medium	high	low
Intrusive	no	no	no	yes	yes	no

Inspection methods rely on expert analysis rather than direct user participation. Holzinger [19] summarized the three most common types of inspection methods: heuristic evaluation, cognitive walkthrough, and action analysis. In heuristic evaluation, specialists systematically compare interface elements against established usability principles (e.g., Nielsen's 10 Usability Heuristics). Cognitive walkthroughs simulate user interaction with the system step-by-step to assess learnability and cognitive load. Action analysis focuses on the efficiency of interaction by breaking tasks into individual physical or cognitive actions and estimating the effort required for their execution. The strength of inspection methods lies in their ability to identify issues early in the design process without requiring end-user involvement. However, they risk overlooking the real-world needs and behaviors of users.

In contrast, test methods require the active involvement of end users and are considered the most fundamental form of usability evaluation. Holzinger [19] distinguishes three central types: thinking-aloud sessions, field observations, and questionnaires. Thinking-aloud sessions involve participants verbalizing their thoughts as they interact with the system. This technique offers valuable insights into user reasoning, misconceptions, and areas of confusion.

While highly informative, thinking-aloud studies can be time-intensive and sometimes alter the natural flow of interaction. For such qualitative usability testing, Nielsen [23] proposed a model to estimate the proportion of usability issues identified based on the number of participants:

$$P_{\text{found}} = 1 - (1 - L)^n$$

where L represents the proportion of usability problems discovered by a single user (typically 0.31), and n denotes the number of test participants. According to this model, testing with five users reveals over 80% of usability issues. Consequently, Nielsen [24] suggests that five participants offer an optimal balance between cost and benefit, recommending that further effort be directed toward iterative design improvements.

Field observations involve studying users in their natural work environments. By observing how systems are integrated into everyday practices, evaluators gain an authentic understanding of usability challenges and context-specific constraints. The method is particularly effective for uncovering major usability breakdowns but requires careful attention to minimize observer interference [19].

Finally, questionnaires provide a systematic means to capture user perceptions, preferences, and satisfaction. They are especially useful when large numbers of participants are involved or when quantitative data is needed to complement qualitative findings. Questionnaires can address overall usability, specific features, or general user experience dimensions [19]. It is generally recommended to conduct such quantitative studies with approximately 20 to 30 participants, as this number provides a good balance between cost and benefit [25, 19]. In this context, the System Usability Scale (SUS) is a widely used standardized questionnaire comprising ten five-point Likert-scale questions that ask users to indicate their level of agreement with specific statements. The SUS is valued for its easy applicability (due to its brevity and low-effort evaluation), technology-agnostic nature, and non-proprietary status [26]. The SUS items are alternately framed positively (e.g., "I thought the system was easy to use") and negatively (e.g., "I found the system unnecessarily complex"), meaning that the highest score of agreement (five) alternately is a positive or a negative signal for usability [4]. The specifics of the SUS survey items are presented in Section 3.3. Based on a user's responses to the ten SUS questions, the *SUS score* can be computed, yielding a value between 0 (worst) and 100 (best). Let $i \in \{1, \ldots, 10\}$ denote the question number, and let R_i represent the user's rating for question i on a five-point Likert scale (1 = strongly disagree, 5 = strongly agree). The contribution of each question, $ScoreContribution_i$, is calculated as follows:

$$ScoreContribution_i = \begin{cases} R_i - 1, & \text{if } i \text{ is odd (positively worded items)} \\ 5 - R_i, & \text{if } i \text{ is even (negatively worded items)} \end{cases}$$

The overall SUS score is then obtained by scaling the sum of all contributions:

$$Score = 2.5 \times \sum_{i=1}^{10} ScoreContribution_i$$

The resulting SUS score can be interpreted by comparing it to established benchmarks from the literature. A score around 68-70 is generally regarded as the average baseline for usability. Scores above 70 are typically considered to reflect good usability, and scores above roughly 80-85 are commonly associated with excellent usability [27, 26, 28].

In practice, effective UCD often combines both inspection and test methods. Inspection provides rapid, cost-effective identification of potential issues, while user-based test methods deliver empirical grounding and validation. Together, they ensure that usability evaluation remains a continuous and integral activity throughout the design process.

2.2.4. User-Centered Design in Practice

Numerous studies have demonstrated the effectiveness of user-centered design methods in improving the usability of specific tools. This trend is increasingly extending to applications in the AI domain [29, 30, 31, 32]. Inspection methods are frequently combined with heuristic evaluation, and Nielsen's 10 Usability Heuristics are among the most widely applied frameworks in this context [33]. User-centered design studies typically achieve a SUS score around 73 to 88, representing improvements of between 2 and 39 points [34, 35, 36, 37, 38, 39].

Two studies stand out in terms of particular challenges. He, Zhang, and Bian [34] conducted the UCD process on *STAT*, a web-based semantic text annotation tool designed to crowdsource information extraction from scientific mental health literature. To improve usability and workflow efficiency, the authors carried out four iterations involving both internal focus groups and external evaluations via Amazon Mechanical Turk[3] . Each iteration combined usability testing, heuristic evaluation, and SUS assessments to identify and resolve interface issues. SUS scores fluctuated across iterations: after rising from 70.3 to 81.1 during the first two internal rounds, the score dropped to 55.7 in the third round when the study was opened to external participants. The decrease was attributed to task complexity and insufficient user guidance. Following further improvements, the final SUS reached 73.8.

Marien, Legrand, Ramdoyal, et al. [35] performed iterative development of *MedRec*, a digital tool supporting medication reconciliation in healthcare settings. Across three iterations involving 48 participants, the researchers employed a mixed-methods approach combining observations, questionnaires, and follow-up discussions. Each iteration informed design improvements to enhance usability and workflow integration. SUS scores improved slightly between the second and third iterations (from 73 to 75). The authors attributed this modest gain partly to the change in study setting: unlike earlier tests, the final evaluation was conducted in a near real-world context, likely introducing additional system requirements related to workflow integration.

User-centered design research on HITL systems remains limited. Smith, Kumar, Boyd-Graber, et al. [31] and Fang, Alqazlan, Du Liu, et al. [32] applied user-centered design methods to a HITL topic modeling tool. HITL topic modeling combines machine learning-based topic modeling, which automatically identifies hidden themes in large text collections, with human input to refine and improve topic quality and interpretability [31]. Although both studies

[3] `https://www.mturk.com/`

focused primarily on usability issues specific to topic modeling, they also revealed insights that may be transferable to other HITL contexts. Smith, Kumar, Boyd-Graber, et al. [31] found that users did not experience practical issues related to trust or control in the HITL system; instead, some participants placed excessive trust in the system, while others showed hesitation or limited confidence in their refinements. Fang, Alqazlan, Du Liu, et al. [32] conducted a follow-up user test after implementing previously suggested usability improvements and found that participants frequently revised and branched their topic models during refinement, highlighting the need for systems to support iterative and exploratory interaction with models.

3. Methodology

This study followed the User-Centered Design (UCD) process introduced in Chapter 2.2.3. In total, four UCD iterations and one preliminary preparatory phase were conducted, as shown in Figure 3.1. In the subsequent chapters, these iterations are referred to as *UCD-1*, *UCD-2*, *UCD-3*, and *UCD-4*. A minor adaptation to the standard framework was made by initially omitting the third UCD step (*Producing Design Solutions*), as a mature prototype was already available at the beginning of the study. This adjustment enabled an immediate evaluation of the existing system and facilitated progressive refinement across subsequent iterations. Each UCD cycle thereafter followed the conventional sequence, as illustrated in Figure 3.1: evaluation, refinement of user requirements, and implementation of the revised requirements. This structured and iterative approach ensured a continuous alignment between user needs, system capabilities, and the overarching research objectives.

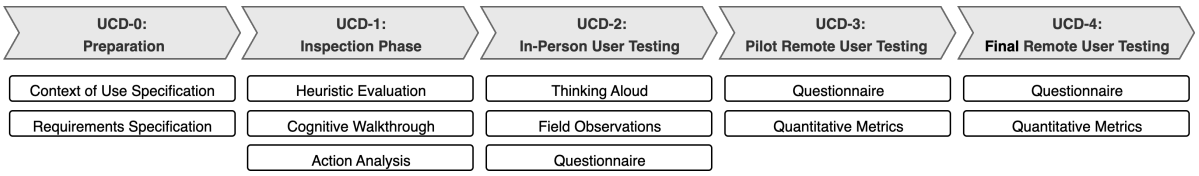

Figure 3.1.: Overview of the UCD process as applied in this thesis.

3.1. UCD-0: Preparation

The initial stage of the UCD process served as a preparatory phase and was conducted in close collaboration with the CD4AI product owner, who supervises the overall CD4AI research project and oversees its strategic direction. Both explicit methods, such as structured interviews, and implicit methods, such as validating preliminary assumptions with the product owner, were applied to establish a solid foundation for the subsequent design activities. The Context of Use was defined according to Maguire's framework [22], which distinguishes four key components: the system and stakeholder analysis, the users (including their skills, knowledge, and attributes), the tasks they aim to accomplish (characterized by their goals, frequency, and outputs), and the environment in which these tasks occur (covering technical, physical, and organizational aspects). In addition, non-functional requirements were retrospectively specified to provide a baseline of design principles serving as benchmarks for prototype evaluation

3.2. UCD-1: Inspection Phase

The design inspection phase constituted the first iteration of the UCD process and focused on systematically evaluating the existing prototype before involving end users, with the goal of uncovering design flaws, workflow inefficiencies, and usability issues through lab-based inspection methods.

Heuristic Evaluation, Cognitive Walkthrough, and Action Analysis. To maximize the quality of findings, the three inspection methods introduced in Section 2.2.3 were applied in a continuous and iterative manner until convergence was reached, i.e., no new usability issues emerged. Heuristic evaluation was conducted using Nielsen's 10 Usability Heuristics. This method was applied statically to the screens and modals of the CD4AI application, enabling the identification of usability violations at the component level. Cognitive walkthrough was used to examine multi-step processes, focusing on how first-time or infrequent users would complete tasks. This approach proved well-suited to analyzing higher-level workflows within the application and uncovering potential points of confusion, unnecessary complexity, or inconsistencies in task progression. Finally, action analysis was employed to decompose individual workflow steps into fine-grained action sequences, such as filling out a form. This method provided a detailed account of micro-interactions (e.g., button clicks, text input) within single components, thereby complementing the higher-level perspective of the cognitive walkthrough. Issues identified during cognitive walkthrough and action analysis were additionally validated against Nielsen's heuristics.

Refinement and Prioritization of the Feature Backlog The three inspection methods resulted in a set of identified issues, which were then translated into a requirements backlog. The requirements backlog was under constant refinement, as evaluation, validation, requirement refinement, and design implementation formed a continuous process. Beyond the identified *usability issues*, two other types of issues were distinguished, enabling the backlog to be structured into three categories. *Defects* are errors or technical flaws in the app. Due to their negative effect on functional quality and user perception, they were resolved with high priority. In addition to usability considerations, several non-functional requirements were identified as essential for completing and deploying the application. These requirements were discussed with the CD4AI product owner and designated as *priority features*. Although not always time-critical, priority features were required to be implemented by the conclusion of the final UCD iteration. To further structure and prioritize the backlog, usability issues were assigned a priority score, calculated as

$$Prio_i = \frac{Value_i}{Effort_i}, \quad Value_i, Effort_i \in \{1, 2, 3, 4, 5\}$$

where i indexes the individual backlog items, $Value_i$ is a 5-point estimation of the requirement's value-add for the user, and $Effort_i$ is a 5-point estimation of the implementation effort. Requirements were then implemented in the order of priority.

3.3. UCD-2: In-Person User Testing

The first user testing round involved a relatively small number of participants, with a focus on gathering qualitative and in-depth insights.

Participants. The study involved eight researchers who voluntarily participated and were affiliated with the Chair for Software Engineering for Business Information Systems at the Technical University of Munich. Participation was uncompensated. Although all participants were from the same chair as the CreateData4AI research project, none reported prior contributions to the project or previous use or testing of the CD4AI web application. Participants' self-reported familiarity with the project varied: two indicated *advanced* knowledge (i.e., familiarity with the project's scientific concepts), four reported *medium* knowledge (i.e., a basic understanding of its goals), and two reported no prior knowledge. Participants were aged between 25 and 34 years, held a Graduate degree, and had a background in either Computer Science / Software Engineering or Data Science / Natural Language Processing, with one to nine years of professional experience. Throughout this thesis, individual participants are identified by a code indicating the UCD iteration (e.g., 2) and the participant's ID within that iteration. Accordingly, participants from this iteration are referred to as P-2.1 through P-2.8.

Test Session. Each participant took part in a one-on-one, in-person test session, with the author of this thesis serving as the instructor. All sessions took place sequentially on the same day, were conducted either in English or German (depending on the participant's preferred language), and lasted between 30 and 45 minutes. Regardless of the participant's previous knowledge about the project, the same pre-defined introduction script (consisting of an overview of the app, its concepts, and the user's task to be fulfilled) was first read aloud to the test user. Once having opened the CD4AI landing page on their own laptop, each participant was then asked to start performing their task, while speaking out loud thoughts, comments, and questions that might arise. The participant's screen was, at the same time, duplicated to a conference screen in the room so the instructor could clearly follow each step. The sessions were neither video- nor audio-taped. The instructor noted participants' comments and peculiarities in their usage on a digital note sheet, structured by the originating UI component and step in the workflow. Besides his mostly passive, observing role, the instructor supported the test users by answering clarifying questions, guiding them through the overall workflow, and asking specific questions (e.g., *Why did you not click button X?*). After completing the test session, each participant was asked to fill out a questionnaire. This was done immediately by six participants and two days later by the remaining two.

Task. The test users were provided with a raw, unlabeled dataset of news articles from the New York Times. The dataset contained 600 samples, evenly split across three ground-truth classes (*business*-related, *politics*-related, and *sports*-related news). The New York Times corpus was chosen because it is widely recognized as a real-world, high-quality dataset with clearly distinguishable categories in a domain that even non-expert users readily understand, while

still being sufficiently challenging to meaningfully test CD4AI's core functionality. Users were then given the goal of using CD4AI to create a labeled dataset from the provided unlabeled data, with the premise that each article must be labeled as either *Business*, *Politics*, or *Sports*. After the first two sessions, all remaining users where instructed to only use CD4AI for two classes (instead of all three). The reason for this adjustment was to keep the sessions shorter in time, as users often encountered the same process for the second or third time without gaining any new insights or comments.

Evaluation Metrics. After participation in the test session, participants were asked to complete a follow-up survey. The first part contained the SUS survey (see Section 2.2.3). The wording of the ten standard SUS questions was minimally adapted for CD4AI. References to "the system" were replaced with "CD4AI", and the first question was prefaced with "Given the need to label data" to account for users who might not otherwise require the system. Users then rate their level of agreement with each item on a five-point Likert scale, where *Strongly Disagree* translates to a numeric rating of 1 and *Strongly Agree* translates to 5. Based on the survey results, the SUS score was computed for each user. Table 3.1 presents the adapted SUS survey used in the user testing sessions. Following the SUS section, the survey included three optional free-text questions designed to elicit additional qualitative feedback that participants might provide retrospectively. These responses offered deeper insights into user perceptions and potential usability issues that could not be fully captured by the think-aloud comments and the standardized survey metrics alone: 1) *What parts of CD4AI (e.g., its idea, concept, app) are still unclear to you, or are you still not understanding?* 2) *What did you enjoy most about the app?* 3) *What frustrated you about the app? Do you have any comments, suggestions, or recommendations for the future?*

Table 3.1.: SUS survey with adapted wording for CD4AI, along the best possible score (BPS) for each item.

ID	Question	BPS
Q1	Given the need to label data, I think that I would like to use CD4AI frequently.	5
Q2	I found CD4AI unnecessarily complex.	1
Q3	I thought CD4AI was easy to use.	5
Q4	I think that I would need the support of a technical person to be able to use CD4AI.	1
Q5	I found the various functions in CD4AI were well integrated.	5
Q6	I thought there was too much inconsistency in CD4AI.	1
Q7	I would imagine that most people would learn to use CD4AI very quickly.	5
Q8	I found CD4AI very cumbersome to use.	1
Q9	I felt very confident using CD4AI.	5
Q10	I needed to learn a lot of things before I could get going with CD4AI.	1

3.4. UCD-3: Pilot Remote User Testing

The goal of this third iteration was to validate users' ability to navigate the app independently and complete a specific task without external assistance.

Participants. Participants were recruited via the study recruitment platform Prolific[1]. Prolific allows setting participation filters, which enable only individuals who meet a defined set of criteria to participate in the study. For this study, only so-called *Qualified AI Taskers* were allowed to participate in the study, which Prolific defines as "participants who have verified experience and/or have passed a targeted skill assessment in areas that are essential for AI training and evaluation, such as reasoning, fact-checking, and image and video annotation" [40]. Besides that, the recruitment filters included an undergraduate or community/technical college degree as the minimum level of educational background, a minimum approval rate[2] of 95, a number of total previous submissions of 100-10000 and English as the primary, first, and fluent language. The mean hourly compensation rate for study participation was £25.32.

Table 3.2.: Demographic overview of participants in pilot remote user testing.

ID	Age	Sex	Professional Background	Highest Degree
P-3.1	39	M	Technical field (Data Science)	Graduate Degree
P-3.2	55	M	Technical field (Data Science)	Undergraduate Degree
P-3.3	38	M	Technical field (general)	Graduate Degree
P-3.4	43	M	Non-technical field	Graduate Degree
P-3.5	41	M	Non-technical field	Graduate Degree

Test Sessions. The participants were instructed remotely via an instruction sheet containing a short description of CD4AI's goals and the task to be performed. In contrast to the in-person sessions, no in-depth information about CD4AI's concepts (e.g., the idea of keywords and context windows) was given in advance. The goal was to solely rely on the help modals and onboarding wizard that were implemented after the in-person sessions.

Task. The same task used in the in-person sessions was also given to participants in the remote sessions. The New York Times news article dataset was downsized from 600 to 400 samples, spread across only two classes (*Business* and *Sports)*. 200 news articles belonging to the third class (*Politics)* were eliminated in order to keep the task shorter in time and focus on CD4AI's essential features. For trust purposes, it was decided that participants should not be forced to download the dataset, as was the case in the first round. Instead, a temporary *test mode* was implemented and deployed, meaning that the file upload feature is disabled and the prepared test file is automatically attached to any newly created project. Participants

[1] `https://www.prolific.com/`

[2] Approval rate refers to the proportion of studies in which a person's successful participation was confirmed by the study administrator.

were made aware of this fact through the instruction sheet and the UI to avoid confusion. Additionally, login and registration were temporarily modified to ensure that no personal data (i.e., users' real names and email addresses provided by the OAuth service used within CD4AI) is collected, stored, or processed, in order to comply with Prolific's platform policies.

Evaluation Metrics. Users were asked to complete a follow-up survey again. By placing the Prolific confirmation code at the end of the survey, users were required to complete the entire survey in order to receive compensation. The survey was not made available in advance but was integrated into the app and became accessible only once the participant had started their first extrapolation. The survey consisted of three sections, starting with the ten SUS questions. The second section consisted of ten Likert-scale questions focused on specific features. The free-text questions in the third section were slightly altered compared to UCD-2. The full content of the second and third sections is presented in Table 3.5 (see column *UCD-3*). Besides the questionnaire, three quantitative key metrics are collected for each user, as presented in Table 3.3.

Table 3.3.: Demographic overview of participants in final remote user testing.

Metric	Description
Task Completion Rate	Portion of users who were able to complete the test task.
Task Completion Time	Time required by users to complete the test task.
Help Modal Views	Number of each help modal was viewed by the users

3.5. UCD-4: Final Remote User Testing

This final round was designed as the most extensive evaluation of CD4AI among all iterations, focusing on both the effectiveness and efficiency of the workflow, as well as participants' satisfaction, preferences, and remaining challenges.

Participants. Participants were recruited via Prolific using the same filters as in the third iteration. While both technical and non-technical backgrounds were eligible, the first seven participants predominantly had non-technical profiles. To better represent technical users and CD4AI's real-world personas, the *Industry* filter was refined from *All* to *Information Services, Data Processing, Software, and Engineering*, recruiting participants P-4.8 to P-4.15. The mean hourly compensation rate for study participation was £25.32. One participant (not included in Table 3.4) was excluded due to suspicious and potentially unreliable survey responses. Although the attention-check question (between Q12 and Q13) was passed, the submission time was implausibly short (under two minutes, while no other participant took less than five), the answers contained contradictions (e.g., reporting confusion about archetypes but stating archetypes as the aspect they enjoyed most), and the SUS responses appeared suspiciously uniform, with the fourth option selected for all questions.

Table 3.4.: Demographic overview of participants in the large-scale usability study.

ID	Age	Sex	Professional Background	Highest Degree
P-4.1	37	M	Non-technical field	Undergraduate Degree
P-4.2	27	M	Technical field (general)	Undergraduate Degree
P-4.3	61	F	Technical field (general)	Doctorate Degree
P-4.4	36	F	Non-technical field	Undergraduate Degree
P-4.5	29	M	Non-technical field	Undergraduate Degree
P-4.6	40	F	Non-technical field	Undergraduate Degree
P-4.7	61	M	Non-technical field	Undergraduate Degree
P-4.8	36	M	Technical field (general)	Undergraduate Degree
P-4.9	42	M	Technical field (Data Science)	Doctorate Degree
P-4.10	40	M	Technical field (Computer Science)	Undergraduate Degree
P-4.11	32	M	Technical field (Data Science)	Technical/community college
P-4.12	46	F	Technical field (Computer Science)	Undergraduate Degree
P-4.13	56	M	Technical field (Computer Science)	Undergraduate Degree
P-4.14	39	M	Technical field (Data Science)	Graduate Degree
P-4.15	26	M	Technical field (general)	Undergraduate Degree

Test Sessions. Due to the improvements made to the in-app support in the third iteration (see Section 4.4), the instruction sheet's content has been reduced to only essential organizational information. The responsibility for teaching the user how to work with CD4AI has been shifted solely to the web application itself. Besides this adaptation, the sessions have been conducted in a manner analogous to the third iteration.

Task. The same task as in the third iteration was used again.

Evaluation Metrics. The user survey from the third iteration has been slightly adapted. Five questions aiming at the satisfaction with the in-app demonstration videos (introduced in the third iteration; see Section 4.4) and overall system satisfaction have been added, as shown in Table 3.5 (see column *UCD-4)*. All remaining questions and quantitative metrics were used again.

Table 3.5.: Survey questions used in the third (UCD-3) and fourth (UCD-4) UCD iteration. The "ID" column shows sequential question numbers (Q1–Q10 are the SUS questions, see Table 3.1). The "M" column indicates whether the question was mandatory.

ID	Survey Question	M	UCD-3	UCD-4
Q11	Creating the project for my labeling task was easy and straightforward.	x	x	x
Q12	Inside the project, I found it clear how to create classes.	x	x	x
–	To show that you are paying attention, please choose (1) “Strongly Disagree”.	x	x	x
Q13	The onboarding wizard provided enough and clear guidance for me to understand what to do.	x	x	x
Q14	If applicable: After watching the demo videos, I fully understood how to use CD4AI.			x
Q15	If applicable: I think the onboarding wizard and the demo videos were combined well.			x
Q16	Starting a keyword extraction was straightforward.	x	x	x
Q17	Selecting keywords was intuitive.	x	x	x
Q18	Selecting context windows was intuitive.	x	x	x
Q19	I understood what archetypes are and what to use them for.	x	x	x
Q20	Creating a labeled dataset felt intuitive.	x	x	x
Q21	While completing tasks, I felt that I knew what I was doing.	x	x	x
Q22	The overall look and feel of the UI was pleasant.	x	x	x
Q23	I imagine that using CD4AI to label data is more enjoyable than manual labeling.	x		x
Q24	Considering my goal to label a dataset, the amount of mental effort required to complete the tasks was reasonable.	x		x
Q25	I would recommend CD4AI to colleagues who need to label data.	x		x
Q26	Which difficulties did you face when using CD4AI?	x	x	x
Q27	Which aspect(s) frustrated you about CD4AI?	x	x	x
Q28	Which aspect(s) did you enjoy about CD4AI?	x	x	x
Q29	Which ideas or advice for future improvement of CD4AI do you have?		x	x

4. Results

Continuous testing and redesign guided the maturation of the CD4AI web application. Insights from each evaluation round directly informed the next, gradually aligning the system's functionality and design with user needs and expectations.

4.1. From Prototype to Deployable System

Several priority features were identified as essential prerequisites for transforming CD4AI from a prototype into a deployable system. Most of these features were implemented before or alongside the first inspection phase (UCD-1), with the exception of the user documentation (F-0.6), which was made public after UCD-4.

F-0.1 – Authentication Providers. In the initial implementation, authentication in CD4AI was limited to OAuth with GitLab as the sole provider. This created an unnecessary entry barrier, particularly for non-technical users who may not possess a GitLab account, thereby reducing the general accessibility of the system. To address this issue, feature F-0.1 introduced multi-provider support within the authentication mechanism, enabling the integration and maintenance of additional OAuth providers with minimal effort. As part of this extension, GitHub and Google were added as alternative sign-in options.

F-0.2 – Public Deployment. As part of F-0.2, CD4AI was deployed on a virtual machine (VM) hosted at the *Leibniz Rechenzentrum*. The application is containerized, with each component running in a dedicated Docker container: the Next.js web client, the FastAPI backend, the MongoDB database, and the Nginx web server. Deployment was based on an existing Docker configuration from a previous setup, which only required adaptation to the new environment. The system is served via HTTPS on port 443 and was initially accessible exclusively through a virtual private network connection to the *Münchner Wissenschaftsnetz*. Public access was enabled at a later stage to support the first external user testing.

F-0.3 – TLS Certificate. The CD4AI application is served under the domain `https://createdata4ai.com`. To establish secure HTTPS connections, the web server must present a valid certificate issued by a trusted certification authority. For this purpose, certificates from *Let's Encrypt* were used, which have a maximum validity of 90 days. After the initial deployment and before the first round of user testing, a new certificate was obtained.

F-0.4 – Archetype-based Classification Algorithm. As outlined in Section 2.1.1, archetypes form the basis for a new classification algorithm developed alongside this study and integrated into CD4AI as feature F-0.4. The implementation is provided via an external GitHub repository and requires two main adaptations of the web application.

First, a new task type was introduced chronologically between the context window task and the extrapolation task. In this task, user-selected context windows are clustered and distilled into archetypes by an on-device Large Language Model (LLM). These archetypes are then presented to the user for validation (see Figure 4.1). Based on discussions with the product owner, archetypes are expected to be concise (approximately one sentence) and few in number; consequently, the selection interface was implemented as a simple list. By default, all archetypes are marked as relevant, as most are assumed to align with the intended class if the preceding keyword and context window tasks were performed carefully. The task then is to manually filter out archetypes that do not fit the class.

Second, the extrapolation task was redesigned. Unlike the previous class-level approach, the integrated classification library operates at the project level, classifying an entire dataset across all classes simultaneously, which reflects the standard practice for classification problems. Accordingly, the former *Generate Result* functionality, which was based on aggregating class-level results, was replaced with project-level extrapolation. Users may still specify which archetype sets should be included in this process (see Figure 4.2).

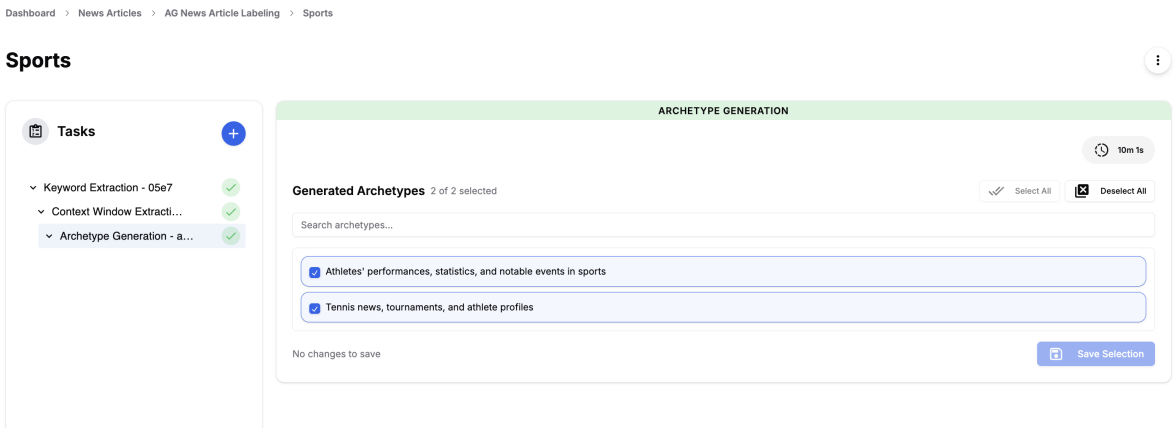

Figure 4.1.: On a dedicated archetype task panel (F-0.4), users can review, select, and filter generated archetypes before inclusion in downstream classification tasks.

F-0.5 – GPU Acceleration & Mutex Polling. The classification algorithm introduced in F-0.4 performs on-device inference of large machine learning models. Processing archetype and extrapolation tasks on a CPU is feasible only for small experimental workloads and becomes impractical for larger, real-world volumes. To efficiently handle these computational demands, the VM's GPU was made available to the API container, with the archetype and extrapolation tasks automatically utilizing CUDA for acceleration. Given the GPU's 16 GB VRAM limit, parallel execution of multiple large tasks can trigger out-of-memory errors. To mitigate this, a

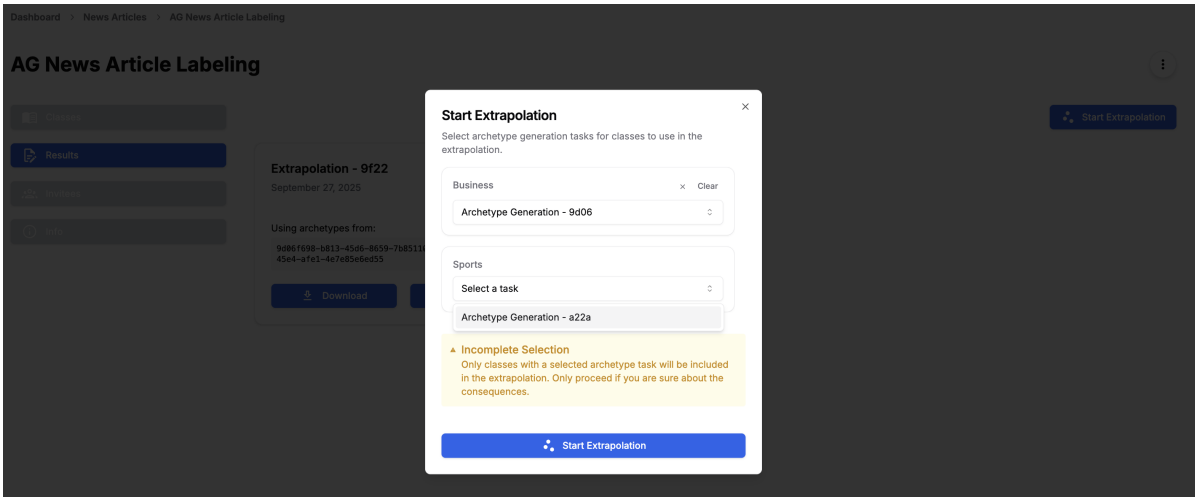

Figure 4.2.: Users can choose which archetype task results to include when performing project-level extrapolation (F-0.4).

mutex-based task queue was implemented, allowing only one GPU task to execute at a time. Concurrent tasks stall and poll a virtual mutex lock every 20 seconds until available. Task execution order is determined by which task acquires the mutex first, rather than a classical FIFO scheme.

F-0.6 – User Documentation. A user documentation was created to explain the key concepts and functionalities of CD4AI. The first chapter introduces the CD4AI pipeline and overall task flow, providing detailed explanations of each task type. The second chapter focuses on the collaboration model, covering workspace and project invitations as well as the use of selection tasks. The documentation is hosted externally as a *Notion Site*, which allows flexible, low-effort maintenance without modifying the source code or redeploying the application. A dedicated app route embeds the Notion Site into a native component within CD4AI, while it also remains publicly accessible at `https://createdata4ai.notion.site/documentation`. A persistent link in the application header directs users to the documentation page.

F-0.7 – Context of Use and User Requirements. Besides the implemented features, the web app was lacking an extensive developer guide and documentation. As part of the UCD process, CD4AI's Context of Use and non-functional requirements were documented. They are intended as a documentation of the app's baseline product philosophy and may serve as a basis for future refinement of the app. Usability stands out as the key requirement for CD4AI. In the context of CD4AI, users should face minimal effort in using the app and getting started with completing their tasks to foster tool adoption. CD4AI introduces a new concept of how the annotation process works, thereby competing with established workflows that users are accustomed to. Additionally, traceability ensures that all actions and tasks, as well as their progress within the annotation process, can be accurately tracked. In CD4AI, completing

this is essential for maintaining transparency in the workflows, understanding how data was processed, and enabling reproducibility. Accuracy is another important requirement. The web app's outputs must be accurate and align with user expectations. Correctness also refers to maintaining logical integrity in workflows, permissions, and data handling. This requirement is crucial for establishing trust in human-in-the-loop systems, especially when automation supplements or replaces manual tasks. Last, modularity and scalability are important. Due to the inherent GPU-accelerated NLP engine, the API service may encounter specific performance requirements with respect to the underlying infrastructure in practice.

Table 4.1.: Condensed Context of Use for CD4AI following Maguire [22].

Category	Summary Description
System Report	CD4AI is a web-based application for semi-automated annotation of text datasets. The primary application areas are data science and AI development, spanning both academia and industry.
User	Primary users comprise (a) *data scientists (DSs)* with expertise in building AI applications, and (b) *domain experts (DEs)*, with expertise in linguistic or subject-specific topics. While the first group is rather comfortable with `.csv`-based workflows and the concepts of Natural Language Processing, DEs may have limited experience with such technology and tools. Both groups are comfortable in understanding English. DSs are mainly motivated by saving time and cost in data preparation and by building innovative AI applications, whereas DEs are driven by contributing domain knowledge to solutions relevant to their own work. Demographics are expected to vary: DSs are predominantly male (around 80% according to Duranton, Erlebach, Brégé, et al. [41]), while DEs are assumed to be more gender-balanced. Overall, users range from early-career researchers to experienced professionals.
Tasks	DSs aim to produce high-quality, labeled datasets to train and evaluate AI models. To achieve this, they explore and select text corpora, prepare datasets, define annotation schemas, and iterate on model development, collaborating with DEs as needed. DEs aim to ensure that domain knowledge is accurately reflected in the data; they occasionally support annotation or validation tasks, while primarily focusing on their core professional activities.
Environment	Users typically work in small- and medium-sized enterprises or research organizations, often with limited resources for large-scale data annotation. Work is performed on standard office or remote setups (laptop or desktop with internet access). Text corpora, in German or English, are managed as `.csv` files. Tasks are carried out in collaborative organizational or research contexts, with few physical constraints and moderate communication requirements.

4.2. UCD-1: Inspection Phase

The inspection phase was conducted through several iterations of heuristic evaluation, cognitive walkthrough, and action analysis, until convergence was achieved. This approach enabled the identification of numerous usability issues, which were addressed through a total of 13 new key features (listed in Table 4.2) and nine fixes of defects and inconsistencies (listed in Table 4.3).

Table 4.2.: Key features resulting from heuristic evaluation (HE), cognitive walkthrough (CW), and action analysis (AA).

ID	Title	HE	CW	AA	Heuristic(s)	Figure
F-1.1	Reduced Friction to the Main Feature		x	x	Nielsen #7	
F-1.2	Linkable Tasks	x			Nielsen #7	4.9
F-1.3	Project-level Task Dashboard	x	x		Nielsen #1, #6	4.3
F-1.4	Email Notification System	x	x		Nielsen #1	4.4
F-1.5	Task Status Banners	x			Nielsen #1, #5	4.5
F-1.6	Auto-Confirm Seed Keywords			x		4.6
F-1.7	Copy, Paste, and Reuse Keywords	x		x	Nielsen #6, #7	
F-1.8	Auto-Confirm Files and Tasks			x	–	4.7
F-1.9	Selection Tasks Consistency	x			Nielsen #3	4.8
F-1.10	Tab Logic on Task Panel		x		–	4.9
F-1.11	Re-Design Keyword Task Panel	x		x	Nielsen #4, #8	4.9
F-1.12	Auto-Merge Selection Tasks	x	x		Nielsen #6, #7	4.10
F-1.13	Reset Controls for Context Window Tasks	x			Nielsen #3, #9	

F-1.1 – Reduced Friction to the Main Feature. During cognitive walkthrough (and a subsequent action analysis-based dive-in into the process), it has been found that the startup process (from initial user registration until the *main feature* of the app) takes longer than it has to be for users to reach the main feature (i.e., starting task-based work on the dataset). Users were forced by the app logic to thoroughly set up their project by defining a workspace, project, and classes with relevant metadata (minimum character length for the names, mandatory entity description). These boundaries conflict with the philosophy that users should be flexible and free to organize their work in a way that suits them best (Nielsen #7: *Flexibility and Efficiency of Use*). To accelerate the startup process and improve user freedom and flexibility, the following adaptations have been made: First, the concept of *standalone projects* has been introduced, i.e., projects with no associated workspace. Users without a need to organize complementary projects within a workspace can now create projects immediately, and optionally (re-)assign a

workspace to the project later. Secondly, all UI field validation of non-system-critical data has been removed (i.e., the minimum character length for workspace/project/class names has been reduced from three to one, and workspace/project/class descriptions are now optional instead of required). Thirdly, it is assumed that in most cases, multiple users create multiple classes at once as part of their initial project setup (most commonly during project setup). Therefore, the UI dialog for creating a class has been expanded to include a batch-create option, eliminating the need to manually open the dialog repeatedly during class creation.

F-1.2 – Linkable Tasks. A cognitive walkthrough of the class-specific task overview revealed a critical issue in task traceability, again conflicting with Nielsen's seventh heuristic. Previously, open tasks were managed solely through application state, rather than being reflected in the URL. As a result, tasks could not be directly accessed or shared via a link, and users were forced to manually search for the task again after refreshing the page in the browser, resulting in a disrupted workflow. With F-1.2, tasks were made linkable by embedding the task ID into the application URL, as shown in Figure 4.9. This enables direct access to specific tasks and allows CD4AI to automatically restore the corresponding task view upon reload. The improvement enhances both shareability and efficiency, especially in collaborative or repetitive task workflows.

F-1.3 – Project-Level Task Dashboard. Efficient project administration in CD4AI requires visibility into the progress of all tasks at a glance, which is considered a high priority for users. However, a key usability issue was identified during heuristic analysis, which conflicted with Nielsen's first heuristic (*Visibility of System Status*). Tracking task progress was evaluated to be cumbersome; navigation deep into individual class-specific task pages was necessary to view statuses. Even then, viewing task progress beyond the class level, such as across multiple task families, was not supported. To address this, a project-level task dashboard was implemented as part of F-1.3 (see Figure 4.3). It provides an overview of all tasks across the project, significantly improving traceability and reducing navigation overhead. Furthermore, leveraging the task linkout mechanism introduced in F-1.2, clicking a task entry on the dashboard now opens the detailed task page directly.

F-1.4 – Email Notification System. While the project-level task overview dashboard improves project-level task traceability, this solution still requires active manual checking of task progress during app usage. As part of F-1.4, an email notification system has been implemented, shifting from an information-pull to an information-push model. This provides users with a customizable option to receive timely updates on task progress (i.e., task completion or failure) via email notifications, thereby eliminating the need for constant manual checking (see Figure 4.4).

F-1.5 – Task Status Banners. In CD4AI, task responsibility is clearly delineated: at any given time, an active task is either being processed remotely by CD4AI or is awaiting input from the user (and/or collaborators). In this context, another issue related to Nielsen's first heuristic

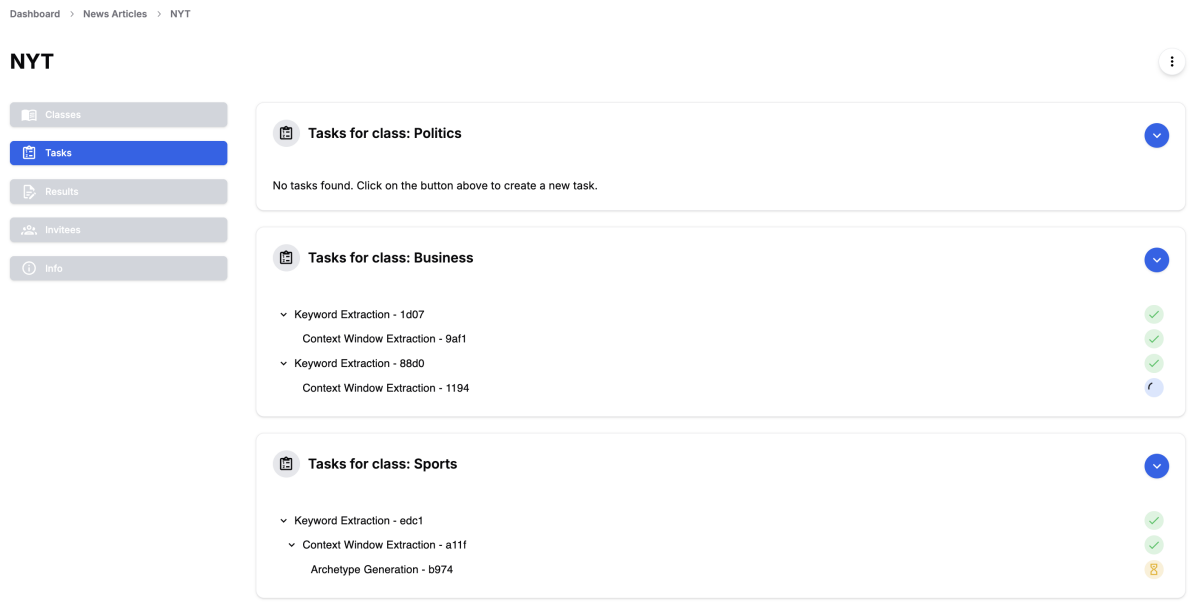

Figure 4.3.: The project-level task dashboard (F-1.3) provides an overview of all tasks, with direct links to detailed pages via the task linkout (F-1.2).

was identified. During remote processing phases, the task interface remained visually active, implicitly encouraging user interaction despite no action being required. Although a small status indicator was present, it lacked the clarity necessary to convey that the user had no current responsibility at that moment. To address this, F-1.5 introduced a solution in which all task interaction components are hidden during remote processing. These are replaced with a clear processing placeholder (see Figure 4.5), making system status more transparent and preventing unwanted user engagement.

F-1.6 – Auto-Confirm Seed Keywords. As outlined in Section 2.1.2, the keyword extraction task can be initiated through two distinct methods: either via the batch-creation action dialog, which allows the user to start tasks for multiple classes simultaneously (see Figure 2.5), or through a single class-level action dialog. Both approaches require the user to define an initial set of seed keywords. However, during action analysis, usability issues with the existing keyword definition process were identified. Under the current design, keywords are defined by typing a word into a text field, pressing *Enter*, and continuing with the next keyword. While this mechanism is explained directly inside the interface, it was found to be unintuitive in practice. Three possible misuse scenarios were identified: *a)* The user enters a keyword but is unsure how to add more, unaware that they need to press *Enter* to confirm the word. *b)* The user enters a keyword but does not press *Enter*, clicks *Submit*, and is confused by the resulting validation error (*at least one keyword is required*), which contradicts their perception of just having defined a keyword. *c)* The user defines multiple keywords but forgets to confirm the last one. Upon submission, the unconfirmed keyword is silently ignored. To address these

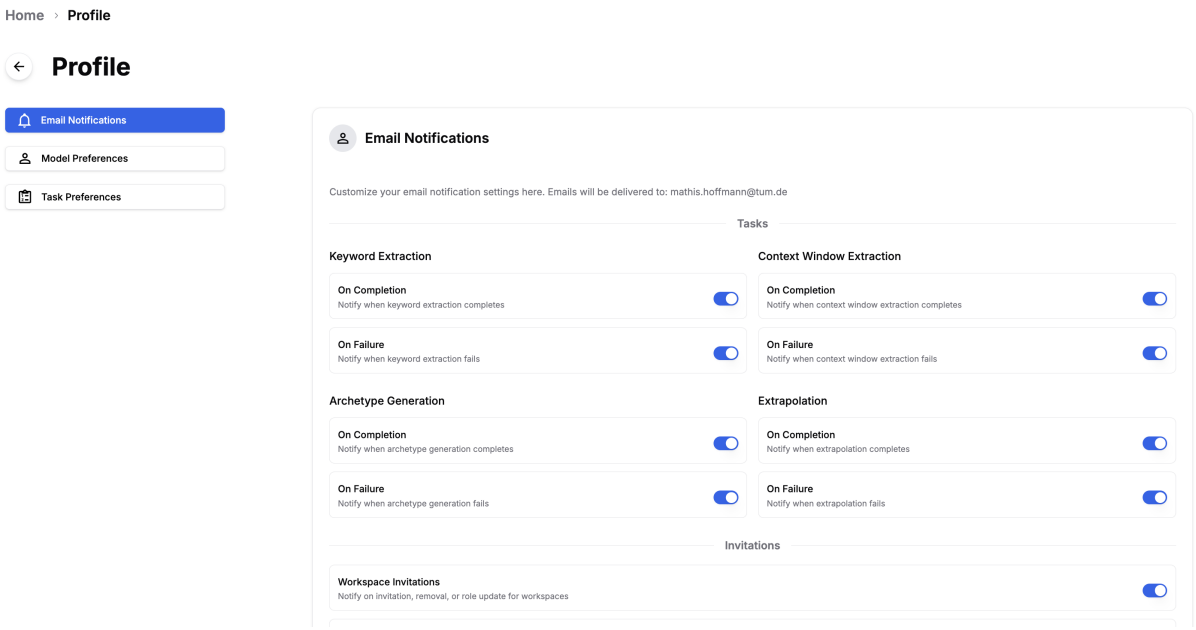

Figure 4.4.: The profile settings interface allows users to select which task updates to receive via the email notification system (F-1.4).

issues, an auto-confirmation mechanism was implemented as part of F-1.6. This mechanism automatically validates and includes the final keyword upon form submission. Consequently, the interaction model has shifted from an *Enter-to-confirm* paradigm to an *Enter-to-add-more* paradigm, thereby enhancing both the intuitiveness and reliability of the user interaction process.

F-1.7 – Copy, Paste, and Reuse Keywords. It is assumed that users may wish to reuse seed keywords from previous extraction tasks, particularly when most original keywords remain relevant and only a few require modification. Previously, CD4AI provided no mechanism to copy or export keywords from existing tasks, nor to paste multiple keywords into the seed definition field, which forced users to manually re-enter all keywords, reducing efficiency. Within the scope of F-1.7, in conjunction with the enhancements introduced in F-1.6, a two-fold solution was implemented to increase flexibility. First, a *Copy* button was added to the keyword input list, enabling users to copy all keywords as a comma-separated string. Second, the keyword input field now supports batch entry of multiple keywords, separated by commas, facilitating both rapid reuse between tasks and the import/export of keyword sets across systems.

F-1.8 – Batch Keyword Extraction: Auto-Confirm Files and Tasks. Batch creation of keyword extraction tasks in CD4AI is managed through a two-step dialog. In the first step, users select one or more project files and their corresponding text columns. In the second step, users define seed keywords and assign each task to a class. Although the interface conceptually

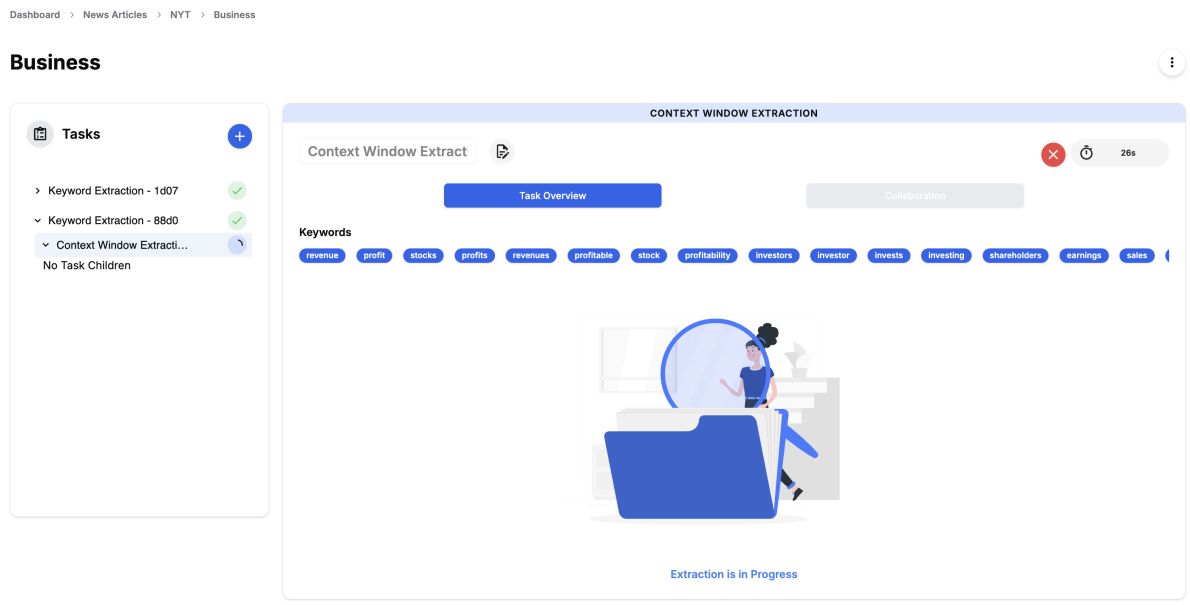

Figure 4.5.: A large task status banner hides the task panel during task processing, indicating that no action is possible (F-1.5).

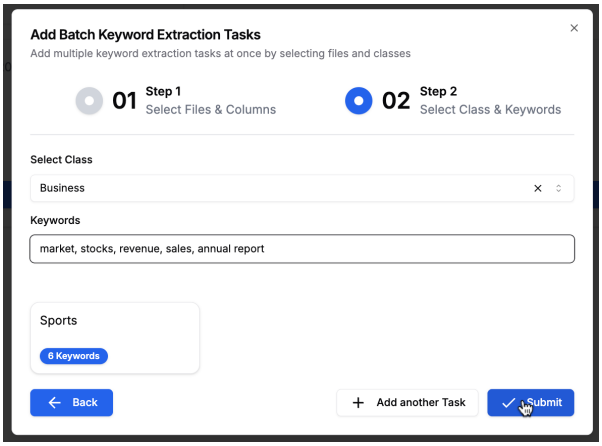

Figure 4.6.: Seed keywords can be batch-defined using commas and explicit confirmation is not necessary anymore (F-1.6). Tasks that still reside in the form fields are auto-confirmed when submitting the task batch (F-1.8).

allows the addition of multiple items (i.e., multiple files in step one, multiple tasks in step two), the implementation requires each item to be explicitly confirmed via an *Add* button. Action analysis revealed that this interaction pattern is unintuitive. Users selecting only a single file may naturally attempt to skip the *Add File* button, triggering a validation error (*At least one file must be selected*). A similar issue occurs in the second step: after defining the final keyword extraction task, users might click *Submit* without first confirming the task

via *Add Task*, causing the last-defined task to be silently discarded. To address this, an auto-confirmation mechanism is implemented with F-1.8. In step one, any valid file-column combination remaining in the input fields is automatically confirmed when proceeding to step two. In step two, any valid keyword extraction task remaining in the input fields is confirmed upon submission of the dialog.

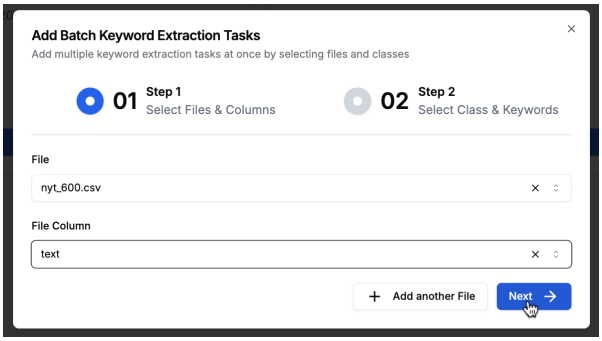

Figure 4.7.: When starting a keyword task, selected files do not have to be explicitly confirmed anymore. After selection, the user can directly proceed to the next step (F-1.8).

F-1.9 – Selection Tasks Consistency. A significant technical inconsistency was identified in the handling of selection tasks, which enable team collaboration at the individual task level. In the original design, admins could create only one selection task per main task, and collaborators' selections could be merged into the main task result only once all invitees had submitted their assigned portion. This design posed several limitations. First, admins were restricted in the number of selection tasks they could create and could not re-invite the same or different participants for subsequent iterations. Second, merging collaborators' selections was a one-time operation; a single subsequent manual update to the task results would irretrievably overwrite the collaboration results. To address these issues, selection tasks were redesigned to allow multiple instances per main task. The former one-time *merge* operation was replaced with an on-demand *retrieve* function, which copies the results of any selection task into the main task as needed.

F-1.10 – Tab Logic on Task Panel. Another issue tightly related to F-1.9 was the lack of visual separation between the task action buttons and the option to create a selection task. As a solution, the task panel was redesigned using a tab-like logic. The first tab (default) contains the as-is task interaction logic for single users. The second tab provides an overview of all existing selection tasks, offers the option to add a new one, and allows for the instant retrieval of results from any completed selection task.

F-1.11 – Re-Design Keyword Task Panel. The keyword task panel (see Figure 2.6) is a core component of CD4AI. Evaluation revealed several usability issues: the sorting control was found to be unintuitive and inefficient in size, with the selected option displayed next to

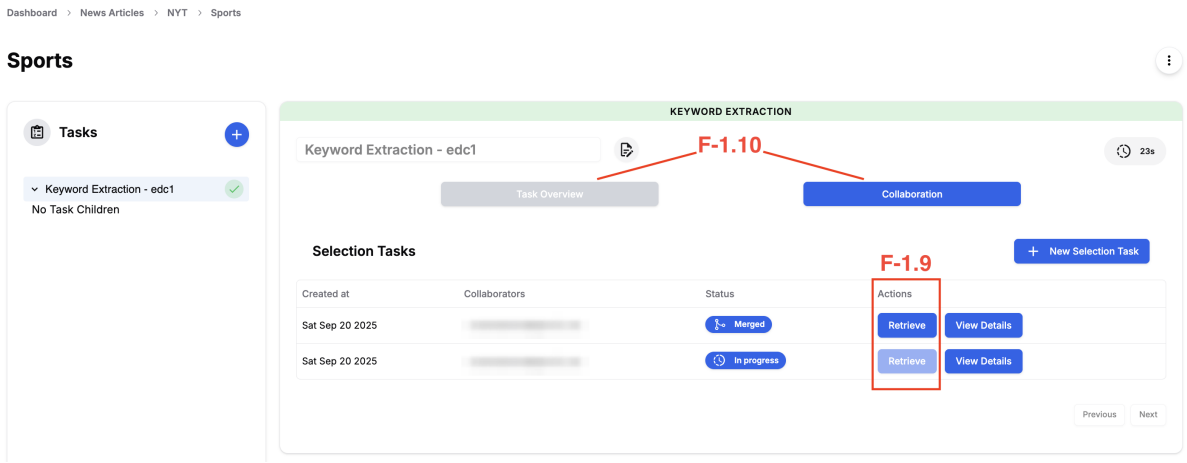

Figure 4.8.: A dedicated collaboration tab (F-1.10) allows to manage more than one selection task and retrieve results more than once (F-1.9).

the sort action; keyword selection actions were inconsistently styled and placed at both the top (*Reset All* styled as an icon) and bottom (*Select All* styled using plain text); plain text, texture-less buttons lacked visual salience; and the search bar label (*Enter a keyword*) risked being misinterpreted as part of keyword definition. To resolve these problems, the sorting function was redesigned as a standard dropdown, all action controls were consolidated at the top, and the search bar was relabeled *Search a keyword*. *Select All* and *Unselect All* were replaced with clear double-arrow icons centrally positioned in the selection panel, and a new option *Select Initial Keywords Only* (single arrow) was added to better reflect common usage patterns. These changes aim to enhance the clarity, consistency, and efficiency of the task panel.

F-1.12 – Auto-Merge Selection Tasks. Nielsen's seventh heuristic emphasizes that expert users may desire shortcuts to streamline their workflow. In CD4AI, results from collaborative selection tasks previously had to be manually merged once all subtask results were submitted. Only then could they serve as input for a downstream child task. While this safeguard ensures quality control, experienced users working in established teams may prefer to bypass manual review and merge. To support this, F-1.12 introduces an *Auto-Merge* option. When enabled during selection task creation, subtask results are merged automatically and directly used to launch the respective child task (e.g., automatically starting a context window task after a keyword selection). As part of this feature, the selection task creation dialog was also redesigned into two tabs: a basic configuration tab and an *Advanced Settings* tab. The latter consolidates options for enabling Auto-Merge and choosing the Split Option, thereby improving clarity while keeping advanced controls accessible but unobtrusive.

F-1.13 – Reset Controls for Context Window Tasks. Previously, context window selection tasks could not be reset or restarted. While users could toggle individual items between *yes*

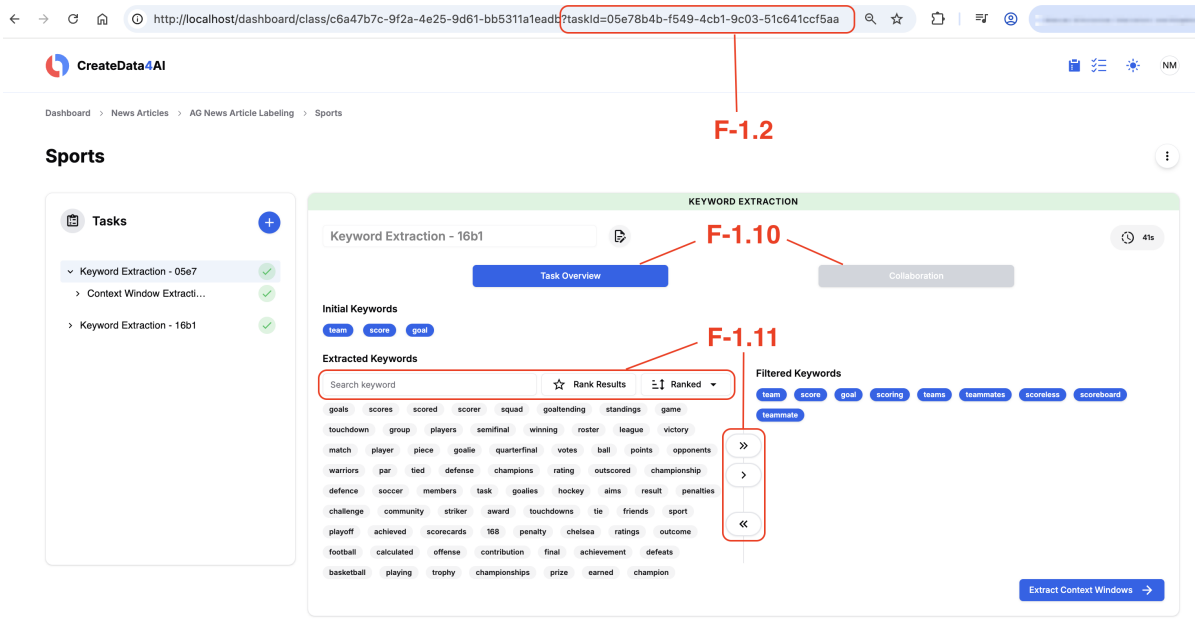

Figure 4.9.: The redesigned keyword task panel integrates multiple usability improvements, including tab-based task logic (F-1.10), linkable tasks for improved traceability (F-1.2), and an enhanced keyword task interface (F-1.11).

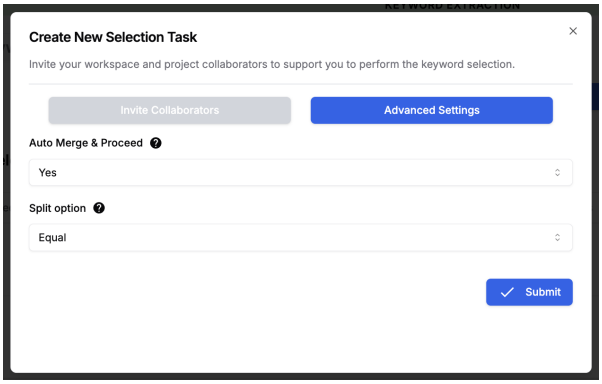

Figure 4.10.: Users can optionally enable *Auto-Merge* when creating a selection task (F-1.12).

and *no*, there was no global reset or action-level undo, violating Nielsen's heuristics on *User Control and Freedom* and *Help Users Recognize, Diagnose, and Recover from Errors*. In contrast, keyword selection tasks already support both functions, creating inconsistency in interaction design. F-1.13 introduces full reset and undo capabilities for context window tasks. The reset function allows users to start over, displaying a confirmation dialog that warns of potential data loss. The new *undo* function tracks the last five actions for step-by-step reversal. Together, these enhancements improve interaction consistency, restore user control, and align CW task behavior with established usability principles without altering core task logic.

Defects and Inconsistencies. In addition to the usability adaptations F-1.1 to F-1.13, a number of system defects, technical errors, and visual inconsistencies were identified and subsequently resolved. These issues ranged from minor visual misalignments to functional shortcomings that could affect task execution and overall workflow reliability. Addressing them was prioritized to maintain a high level of external system quality and to ensure a consistent, dependable user experience for all users. Table 4.3 summarizes the most relevant issues addressed.

Table 4.3.: Overview of resolved defects and inconsistencies in the inspection phase.

Issue	Problem	Resolution
F-1.14	Search bars existed in some views (projects, classes) but were non-functional; absent in others (workspaces, results).	Implemented fully functional, consistent search bars across workspaces, projects, classes, and results.
F-1.15	The task cancellation function was not implemented; although a confirmation message was displayed, no actual cancellation occurred on the backend.	Implemented full backend integration for task cancellation, ensuring that user-initiated cancellations are now properly executed and reflected in the system state.
F-1.16	The API rejected Windows-originating `.csv` files during project creation.	Ensured operating system-independent acceptance of `.csv` files.
F-1.17	The UI allowed to select files with arbitrary types. File rejection only happened inside the API.	Restricted frontend input to `.csv` only.
F-1.18	Mandatory form fields were not visually distinguished from optional ones.	Introduced explicit markers for required input fields.
F-1.19	Auto-generated task names falsely used raw Enum keys from the source code (e.g., `TaskType.KEYWORD_EXTRACTION`).	Converted enum values into readable task labels (e.g., *Keyword Extraction*).
F-1.20	The creator of a selection task was unable to reassign their own subtask, only those assigned to others.	Enabled reassignment of all subtasks, including those initially assigned to the task creator.
F-1.21	Numerous inconsistencies across UI elements (naming, color and design of several buttons and pages).	Standardized styling and alignment across all views.
F-1.22	Unstably implemented on-demand logic for task loading caused tasks to randomly vanish from the task tree.	Replaced on-demand loading with static logic, guaranteeing consistent task visibility.

4.3. UCD-2: In-Person User Testing

UCD-2 was conducted as a one-on-one, in-person user test session with eight participants. The session combined the *think-aloud* protocol with a post-test questionnaire to capture both behavioral and subjective feedback.

4.3.1. Quantitative Feedback

Table 4.4 shows each participant's response to each SUS question, and the resulting SUS score. Overall, a SUS score of 75.6 has been achieved, indicating above-average usability of the app. The range of SUS scores spans 62.5 points, and the standard deviation is 18.5.

Table 4.4.: SUS results from the in-person user testing.

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS Score
P-2.1	4	2	5	2	5	1	5	1	4	3	85.0
P-2.2	5	1	5	1	5	1	5	1	4	2	95.0
P-2.3	4	2	4	2	4	1	4	2	4	2	77.5
P-2.4	3	1	5	2	5	1	5	1	4	2	87.5
P-2.5	4	2	4	2	4	3	2	3	4	3	62.5
P-2.6	4	3	4	1	4	2	5	2	4	1	80.0
P-2.7	3	4	2	5	3	3	4	5	2	4	32.5
P-2.8	4	2	4	1	4	1	4	1	4	1	85.0
Best Possible (BP)	5	1	5	1	5	1	5	1	5	1	100
Mean Diff. to BP	1.1	1.1	0.9	1	0.8	0.6	0.8	1	1.3	1.3	24.4
Overall Mean SUS Score: 75.6											

4.3.2. Qualitative Feedback

Think-Aloud Issues (TAIs). During the think-aloud sessions, a total of 19 issues have been identified, as shown in Table 4.5.

Frustrations. In addition to the TAIs, two points of frustration were mentioned in the survey questions. While P-2.8 kept it rather general ("Sometimes the navigation was a bit confusing"), P-2.1 concretely pointed to the placement of the extrapolation feature: "extrapolation button was hidden in results. I don't expect it there, since it starts a next step". P-2.2 was frustrated about the fact that a manual page refresh was necessary after creating certain objects, like a workspace. It later turned out that this was a bug introduced in the implementation of F-1.17. Five participants did not report any frustrations.

Positive Feedback. Five participants provided feedback on which parts of CD4AI they enjoyed most. All of them praised a pleasant and intuitive user interface. P-2.8 additionally praised CD4AI's core feature to "automatically assign classes."

Table 4.5.: Think-aloud issues (TAIs) identified during in-person user testing.

ID	Description	#
TAI 1	No processing time shown for extrapolation tasks, reducing traceability and consistency.	1
TAI 2	Desire for an *Undo* button to revert keyword selections.	1
TAI 3	One email per finished task caused overload (approx. 10 per project).	2
TAI 4	Missing status banner for the *Queued* status, and dissatisfaction with the *Queued* status icon.	2
TAI 5	A user explicitly requested a guided tour explaining the overall workflow before the first task.	1
TAI 6	Confusion about how to proceed when all generated archetypes looked good and no manual action was needed.	2
TAI 7	When saving task selections, the button to start the next task appears in the same place, causing confusion.	1
TAI 8	In light mode, the tab buttons used across the entire app have a light gray background color when unselected, making them appear like disabled functions.	2
TAI 9	Unclear meaning and metric behind keyword *Rank* button.	2
TAI 10	Users did not understand the dropdown to choose an embedding model, as only one option was available per language and no inline information was provided.	2
TAI 11	The navigation from the task back to the project was confusing; one user misinterpreted the *Cancel* button.	3
TAI 12	The tab logic on the task panel was mistaken for action buttons, leading to uncertainty about task completion flow.	3
TAI 13	Selecting files and columns for keyword extraction was unintuitive.	2
TAI 14	Some users did not use the batch-create option because they did not understand the button label. Others overlooked the option to add more than one task inside the batch-create dialog.	4
TAI 15	Users did not find the function to start the extrapolation task. When starting the task, one user mentioned not understanding what they were doing there.	5
TAI 16	One user reported a missing status toast after project creation.	1
TAI 17	Users were frustrated that new workspaces or projects were not shown until manual refresh.	2
TAI 18	One user's name was displayed as *None* after GitHub login.	1
TAI 19	Users did not notice the option to add more classes in one go and created classes one by one instead.	4

Future Recommendations. Three participants explicitly provided suggestions for future improvements to the app. All of them suggested in-app onboarding features, such as a guided user tour, simple documentation, and more information buttons containing explanations of concepts. Besides that, P-2.2 suggested an authentication mechanism using email and password, instead of relying on OAuth with external providers only. P-2.6 concretely proposed the idea of improving certain button colors ("gray color made it seem like they were disabled"), remove the need to explicitly to select an embedding model as long as CD4AI supports only one model per language, and more consistency across task types (pointing to the fact that, unlike other tasks, no task timer was shown for extrapolation tasks).

4.3.3. Feedback Implementation

To address especially the emerged issues from the think-aloud sessions, a new backlog of features and fixes has been crafted.

F-2.1 – App Navigation and Routing Improvements. TAI 11 has been addressed by adding additional navigation buttons. To facilitate easy navigation, a dedicated *back* button has been introduced with a fixed placement in the top left corner, as indicated in Figure 4.11. The button enables users to route back from class to project, from project to workspace, and from workspace to dashboard. Furthermore, TAI 15 and TAI 6 were addressed by adding a quick navigation button to the archetype task panel, providing direct access to the labeled dataset overview page. Additionally, users can now record brief notes for archetype tasks to facilitate the selection of input archetype results during dataset generation. These notes are displayed inline within the archetype dropdown menu to improve clarity and task traceability.

F-2.2 – Time-based Task Auto-Save. Several problems had emerged with CD4AI's original task save mechanism. In its initial form, the context window task used an event-based auto-save: after each user decision (accept/reject), the client triggered an API call. This introduced a short processing delay of approximately 1000ms, during which subsequent user inputs could not be registered. Any selections made in this interval were lost and re-presented to the user, disrupting the workflow. To address this issue, two potential approaches were considered: (a) disabling the input buttons during the saving period, or (b) switching to a time-based auto-save strategy. The latter was implemented for F-2.3, with the saving interval set to a rough estimation of 1000ms. Instead of saving immediately after each interaction, CD4AI now batches decisions and writes them to the database at fixed intervals, thereby reducing the frequency of blocking states. The subsequent user testing was intended to determine whether this approach was more practical than disabling the input buttons during each save. In addition, the locational placement of the save button caused confusion to some users, as reflected in TAI 7. The save button was therefore moved to the top right corner of the task panel. Presumably, the auto-save mechanism and the more subtle placement of the button will also help eliminate TAI 6.

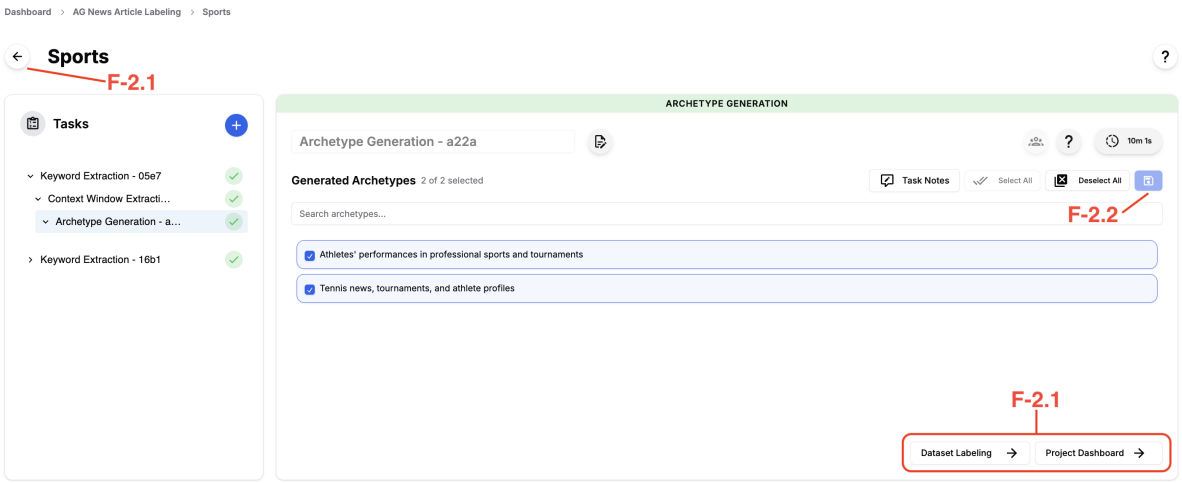

Figure 4.11.: A back navigation button (top left), newly added navigation icons at the bottom of the archetype task panel (F-2.1), and the repositioned save button (F-2.2) intend to provide enhanced workflow clarity and smoother task interactions.

F-2.3 – Redesigned Tab Buttons and Task Panel. Users were confused about the background color of tab buttons, which are primarily used for navigation on the CD4AI dashboard (TAI 8). Therefore, the default background color for all tab buttons was changed from light gray to white, and the text color was adjusted from white to black. The new design aims to appear more neutral and to avoid the impression of a disabled state. In addition, confusion also arose from the use of these tab buttons within the task panel (introduced in F-1.10). Although this was partly related to the color scheme, some users also misinterpreted the tab descriptions, leading to uncertainty about the completion flow (TAI 12). To maintain a clear and minimalist interface, the tab logic was removed with F-2.3, and the elements previously located under the *Collaboration* tab were moved into a modal window that opens when the *Collaboration* button is clicked. This adjustment provides a more guided and focused user experience by encouraging users to first complete the task content before engaging with collaboration options.

F-2.4 – Keyword Interaction and Ranking Enhancements. Several refinements were introduced to improve the usability of keyword-related tasks. User confusion about the keyword ranking feature (TAI 9) was addressed by integrating ranking options into the existing sort dropdown menu, replacing the standalone *Rank* button. The default ranking was renamed from *Ranked* to *Default Ranking*, and the alternative ranking option was named *CD4AI Ranking*, with hover messages provided to enhance transparency. Additionally, to support common user workflows observed during testing, an *undo* button was added in the top-right corner of the keyword task panel (TAI 2). This feature works similarly to the *undo* feature for context windows (F-1.13) and tracks the five most recent actions, allowing users to reverse up to five selections. A search bar for selected keywords was also introduced, enabling users to effi-

ciently locate and filter both selected and unselected keywords. This addition accommodates the frequent strategy of selecting all keywords first and then manually deselecting undesired items (see Figure 4.12).

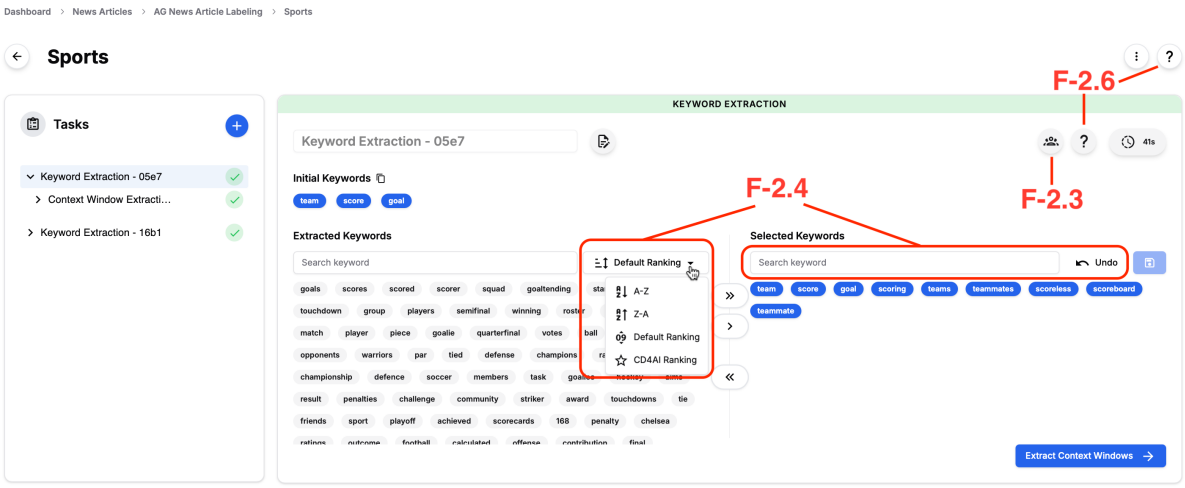

Figure 4.12.: The redesigned keyword task panel incorporates the removal of tab buttons (F-2.3), integrated keyword ranking within the sort dropdown (F-2.4), and help icons for quick access to contextual guidance (F-2.6).

F-2.5 Terminology Adjustment: *Create Labeled Dataset.* Several issues related to extrapolation tasks were identified (TAI 15). To improve clarity, the technically abstract term *extrapolation* was replaced with terminology that more directly conveys the user's goal of *generating a labeled dataset*. Accordingly, the *Results* tab, which previously contained all extrapolation tasks, was renamed *Labeled Datasets*, and the extrapolation start dialog was renamed *Generate Labeled Dataset*. This approach will also guide future revisions of user documentation and help materials.

F-2.6 – Onboarding Wizard. Think-aloud sessions highlighted areas where users required additional guidance. In response, an introductory help system, or onboarding wizard, was implemented to support independent use of CD4AI. Two approaches were considered. Guided user tours, such as React Joyride[1], provide step-by-step walkthroughs, whereas static help modals present targeted information. As user challenges were primarily conceptual rather than navigational, the latter approach was chosen. Help modals provide concise explanations of CD4AI's concepts while minimizing cognitive load. Each modal communicates only the information necessary to complete a task and uses procedural elements to clarify sequence and progression. A hidden tracker records how often a modal has been viewed. Unseen modals open automatically when a user first visits the corresponding page, but they can

[1] `https://react-joyride.com/`

always be reopened via a consistently placed help icon, typically located in the top right corner of the page or component. Because users demonstrated little confusion regarding CD4AI's organizational structure (workspaces, projects, classes), the introductory modals were kept minimal (see Figure 4.13).

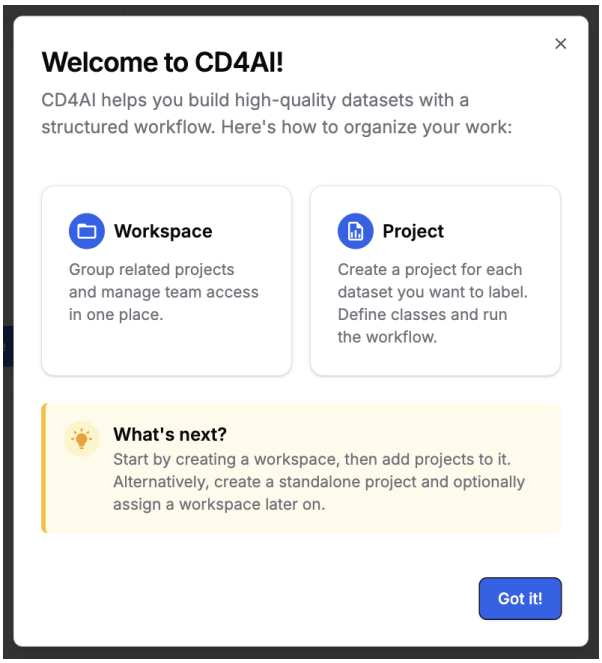

Figure 4.13.: The welcome modal opens on first login and introduces the user to workspaces and projects (F-2.6).

The primary complexity was found to lie in the workflow model. Test users expressed a desire for a high-level overview of all necessary steps to create a dataset, without being overwhelmed by excessive technical detail. To balance these needs, the project dashboard integrates a sequential overview of task types (Figure 4.14). Dedicated modals for keyword, context window, archetype, and extrapolation tasks provide explanations of task-specific interactions. Their procedural design ensures that users understand what has happened, their current position in the workflow, and the next step to take. Additional modals support collaboration features, comprising workspace and project invitations, task-level collaboration guidance, the keyword selection interface for task invitees, and the context window selection interface for task invitees.

Other. In addition to the major feature implementations described above, several minor user interface refinements were introduced to address individual TAIs. **TAI 1** was resolved by aligning archetype task cards with the visual design of other task types. A task timer was added to ensure consistency and improve traceability of processing durations. **TAI 3** was addressed by disabling email notifications for completed tasks by default. Users can enable these notifications manually through the application settings if needed. **TAI 4** was mitigated by

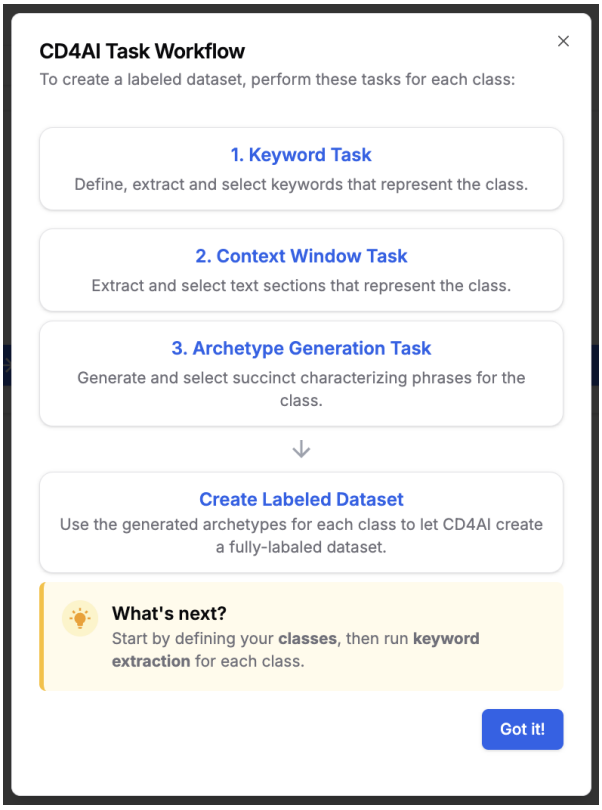

Figure 4.14.: The task help modal provides users with sequential guidance through all steps of the CD4AI pipeline (F-2.6).

replacing the *queued* status icon with a more intuitive alternative and adding a corresponding status banner. The new banner follows the same visual logic as the *in progress* and *failed* statuses to ensure interface consistency. **TAI 10** was addressed by disabling manual input for the embedding model selection field. CD4AI now automatically selects and displays the appropriate model based on the chosen data language. To maintain transparency and enable future extensibility, the field remains visible but inactive. **TAI 13** was addressed by introducing an informational icon (*?*) adjacent to the file and column selection fields. Additionally, the selection interface was redesigned for consistency, utilizing dropdown menus to enhance clarity and usability. **TAI 16** was resolved by standardizing toast notifications across the application and adding the previously missing confirmation message, utilizing dropdown menus to enhance clarity and usability. All emojis were removed from toast messages to enhance linguistic professionalism and reduce visual clutter. **TAI 17** was resolved by ensuring that newly created entities (e.g., workspaces, projects) appear immediately within the interface, eliminating the need for manual refreshes. **TAI 18** was linked to GitHub's privacy mechanism, which occasionally obscures user names in OAuth. A fallback mechanism was implemented to handle cases where no real name is provided by the authentication service, defaulting the display name to *User*. **TAI 19** was addressed by improving the discoverability of the *add*

multiple classes function. A "+" symbol was added to the corresponding button, enhancing visual salience with minimal implementation effort. **TAI 14** is the only issue that was not addressed. Possible solutions, such as adding a confirmation prompt to encourage users to create additional tasks, were considered. However, these interventions risked introducing unnecessary interruptions for experienced users and were judged to have limited benefit relative to the implementation effort. Consequently, this TAI has been deferred.

4.4. UCD-3: Pilot Remote User Testing

This section provides an overview of the third UCD iteration, whose goal was to validate users' ability to navigate the app independently and complete a specific task without external assistance.

4.4.1. Quantitative Feedback

System Usability Scale (SUS). The user testing yielded a SUS of 63.0, as shown in Table 4.6, indicating below-average usability. The score dropped by 12.6 points compared to the in-person user testing. Only two out of five responses yielded a SUS over the baseline threshold of 68. Notably, the wide distribution of SUS scores spans a difference of 65 points from the lowest (27.5) to the highest (92.5). The standard deviation is 21.5.

Table 4.6.: SUS results from the pilot remote user testing.

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS Score
P-3.1	4	4	4	2	3	2	5	2	4	3	67.5
P-3.2	4	2	4	1	3	1	4	2	2	2	72.5
P-3.3	5	1	5	1	5	1	5	1	5	4	92.5
P-3.4	2	5	2	4	2	2	2	5	2	3	27.5
P-3.5	2	2	3	1	4	1	2	3	2	4	55.0
Best Possible (BP)	5	1	5	1	5	1	5	1	5	1	100
Mean Diff. to BP	1.6	1.8	1.5	0.8	1.6	0.4	1.4	1.6	2	2.2	37.0
Overall Mean SUS Score: 63.0											

Survey Results. Table 4.7 shows the mean Likert ratings for each survey question in the second testing round. As all questions were phrased positively, higher scores reflect more positive evaluations. Clear differences between the two participant groups can be observed: on average, technical participants rated their experience more positively (mean 4.0) than non-technical participants (mean 3.0), with an overall mean of 3.5 across all participants. A two-sample t-test assuming equal variances[2] indicated that this difference is statistically significant, with $t(18) = -3.42$, $p = .003$ (one-tailed). However, given the small sample size

[2]Mean survey rating: $Variance_{non-technical} \approx .53$, $Variance_{technical} \approx .48$

($n = 5$), this result should be interpreted with caution and considered only as an indication of a potential trend. The main challenges for non-technical users remained in conceptual aspects of the workflow, particularly understanding context windows (Q18), archetypes (Q19), and how these tasks contribute to creating a labeled dataset (Q20, Q21). The onboarding wizard (Q13) was widely perceived as insufficient for effective onboarding. By contrast, more practical elements, such as creating projects and classes (Q11, Q12), and performing the keyword task (Q16, Q17) received more positive ratings. A large discrepancy of 2.2 points can be observed between the non-technical and technical groups in the rating of the UI's look and feel (Q22), with the latter group rating this aspect notably better.

Table 4.7.: Mean item ratings for the pilot remote user testing survey. Note: Q14, Q15, Q23, Q24, and Q25 were not part of this iteration.

	Q11	**Q12**	**Q13**	**Q16**	**Q17**	**Q18**	**Q19**	**Q20**	**Q21**	**Q22**
All Users	**4.0**	**4.0**	**2.8**	**4.0**	**4.4**	**3.8**	**3.0**	**3.2**	**3.0**	**3.8**
Non-Technical Users	3.0	3.0	3.0	3.5	4.0	4.0	2.5	2.0	2.0	2.5
Technical Users	4.7	4.7	2.7	4.3	4.7	3.7	3.3	4.0	3.7	4.7

Task Completion Rate (TCR). Four out of five participants were able to complete the task, resulting in a TCR of 80%. Even after requesting support out via the Prolific contact form, P-3.1 initially had severe issues in understanding the use and goals of CD4AI. After 80 minutes into the task, he was thus asked to proceed directly to the survey, without creating a labeled dataset after generating archetypes. P-3.1 will thus be excluded from the following quantitative metrics.

Task Completion Time (TCT). The mean TCT was 29 minutes and 27 seconds (excluding P-3.1). Notably, the median time is only 21 minutes, as three participants completed their task within 19 to 22 minutes, while the fourth participant took over 56 minutes to complete it.

Help Modal Views. Table 4.8 shows the average number of times each help modal has been viewed (excluding P-3.1). The task and introduction help modals were accessed most frequently, suggesting that users primarily needed guidance on the overall workflow and task execution. In contrast, the classes and context window help modals were used very little, indicating that these concepts were understood more easily. On average, technical participants opened the help modals more frequently.

4.4.2. Qualitative Feedback

Largest Difficulties. Four out of five participants stated that their largest difficulty using CD4AI was a lack of clear instructions or overall conceptual guidance, leaving them uncertain

Table 4.8.: Mean number of views per help modal during the pilot remote user testing. KW = keyword task, CW = context window task, AT = archetype task, EXT = extrapolation

	Intro	Classes	Task Flow	KW	CW	AT	EXT	Total
All Users	**1.8**	**1.3**	**2.0**	**1.0**	**1.0**	**1.3**	**1.0**	**9.3**
Non-Tech. Users	1.0	1.0	1.0	1.0	1.0	1.0	1.0	7.0
Technical Users	2.7	1.3	3.0	1.0	1.0	1.0	1.3	11.5

about how to proceed in certain scenarios, what the system was doing, and what the ultimate goal of their actions was. P-3.2 stated:

> "The tool seemed very easy to use. The actual instructions were very minimal though. The wizard helped me go through all the steps pretty easily, but I wasn't entirely sure what the end goal was going to be so the steps all felt a bit separate to me."

P-3.5 gives a similar statement and mentions a missing overall explanation of the system:

> "There wasn't a really clear overall explanation of what and how the system worked and what one was trying to achieve. With no experience prior to this, it is hard to really tell if what I was doing was right."

This statement indicates that the onboarding wizard's illustration of the general task flow (see Figure 4.14) did not fully fulfill its purpose. Only P-3.3 stated not to have perceived any difficulties with the system, responding: "Nothing, the instructions given were clear and easy to follow."

Frustrations. The reported difficulties were reflected in the responses of three of the five participants regarding their main frustrations with CD4AI. Two distinct types of usage confusion emerged: P-3.1 and P-3.4 were frustrated by not knowing which steps to perform and in what order, whereas P-3.2, while finding the system intuitive, was frustrated by a lack of explanation regarding the purpose of each step and the overall goals, which made the process feel procedural rather than meaningful. On the other hand, P-3.5 stated a frustration with CD4AI's task processing time: "There was a reasonable wait for the system to perform certain functions. Since there was no indication of any sort of progress it made it a little frustrating just having to wait." The response suggests that setting clear expectations about the task processing time may reduce or even resolve that frustration. Only P-3.3 stated not to have perceived any frustration.

Positive Feedback. Participants also provided insight into which aspects of CD4AI they enjoyed most. P-3.1 appreciated the system's step-wise workflow, describing it as "cool and fluid" once familiar with the steps. Similarly, P-3.2 highlighted the tool's intuitiveness, noting

that it guided him through the process effectively and that the contextual help allowed him to continue smoothly even when he was uncertain about the next step:

> "The tool was very intuitive. Even when it wasn't clear what I 'had' to do next, the tool lead me through it nice and easily and clicking the '?' [*the participant refers to the button that opens the help modal*] always got me onto the next step easily if I didn't immediately know what I wanted to do next."

Other participants emphasized the simplicity and accessibility of the interface: P-3.3 praised the "simple User Interface", while P-3.4 enjoyed "teaching myself how to use it". P-3.5 reflected on the system's potential utility, stating that "I certainly like the idea and could see it be very useful."

Suggestions for Improvement. Four out of five participants suggested that the app needs more extensive onboarding methods for novice users, providing clear examples and step-by-step guidance to help new users build confidence and reduce uncertainty in their initial interactions. Ways to achieve this could be to give a "full example first" (P-3.4) or a "brief introductory document that highlights all the steps that you will be going through" (P-3.2). Two of those (P-3.1 and P-3.4) even suggested a video-based introductory demonstration. P-3.2's and P-3.5's feedback focused more on the conceptual understanding of the tasks, noting that grasping the overall objectives and workflow would help them perform more confidently and efficiently. P-3.5 summarized it as this: "[...] I think I would need to have a deeper understanding of what I was trying to achieve and how the system worked." P-3.3 did not see any potential for improvement and concluded: "For me, I think it is perfect".

Table 4.9.: Summary of user feedback on CD4AI from the pilot remote user testing.

Dimension	**Theme**	**# Mentions**[3]
Difficulties	Conceptual Understanding of the Workflow	**4** (2 + 2)
	–	**1** (0 + 1)
Frustrations	Conceptual Understanding of the Workflow	**3** (1 + 2)
	Task Processing Time	**1** (1 + 0)
	–	**1** (0 + 1)
Positive Feedback	Structure of the Workflow	**2** (0 + 2)
	Joy in Learning and Exploring the App	**2** (1 + 1)
	User Interface	**1** (0 + 1)
	Conceptual Usefulness and Potential	**1** (1 + 0)
Recommendations	Improvements to the Onboarding Experience	**4** (2 + 2)
	–	**1** (0 + 1)

[3]Interpretation note: 3 (1 + 2) means 3 **overall** mentions, originating from 1 **non-technical** and 2 **technical** participants.

4.4.3. Feedback Implementation

The second round of user testing revealed issues related to onboarding, instruction, and support. Based on this feedback, three key features have been identified.

F-3.1 – Demonstration Videos. As mentioned earlier, four out of five participants suggested a more extensive onboarding process for novice users, while two specifically suggested an introduction based on demonstration videos. Additionally, the verbal instruction style in the first round of user testing may have positively influenced users' understanding of the workflow and, consequently, their overall perception of the system, compared to the statically designed onboarding wizard in the second round. Therefore, the onboarding wizard was extended with optional (i.e., skippable) demonstration videos. Just as the onboarding wizard, the videos were logically placed and structured along the CD4AI workflow.

Table 4.10.: Overview of demonstration videos added to the onboarding wizard.

Demo Content	URL	Duration
Setup: Introduction to workspaces, projects, and classes in CD4AI	`https://app.supademo.com/demo/cmfbew4cb6x2t39ozfw4to1tl?utm_source=link`	56s
Keyword (KW) Extraction: Defining seed KWs and starting a KW extraction	`https://app.supademo.com/demo/cmfbgoqkq6ype39ozw83dcdz3?utm_source=link`	1min 18s
Keyword (KW) Tasks: The idea of KWs and how to perform KW selections	`https://app.supademo.com/demo/cmf6zqnbf4kgm39ozzj7m6w1q?utm_source=link`	44s
Context Window (CW) Tasks: The idea of CWs and how to perform CW selections	`https://app.supademo.com/demo/cmf70b6gr00olt80iijjsd53a?utm_source=link`	1min 22s
Archetype (AT) Tasks: The idea of AT and how to perform a AT selection	`https://app.supademo.com/demo/cmf708kyl4kx839ozmuxtibbt?utm_source=link`	40s
Extrapolation: How to use archetypes to create a labeled dataset	`https://app.supademo.com/demo/cmf70da9z4l5l39ozypqadu03?utm_source=link`	60s

The demonstration videos were created using Supademo[4] and are designed as interactive, step-by-step flows with an AI-generated reader voice. They are between 40 and 82 seconds long. In total, it takes exactly six minutes to watch all demo videos. The demo videos are watchable via a newly integrated button inside the individual onboarding wizard modals. First-time users who close an onboarding wizard modal without clicking on the *Watch Demo* button receive an additional reminder, explicitly asking them whether they wish to watch the demo video.

[4] `https://supademo.com/`

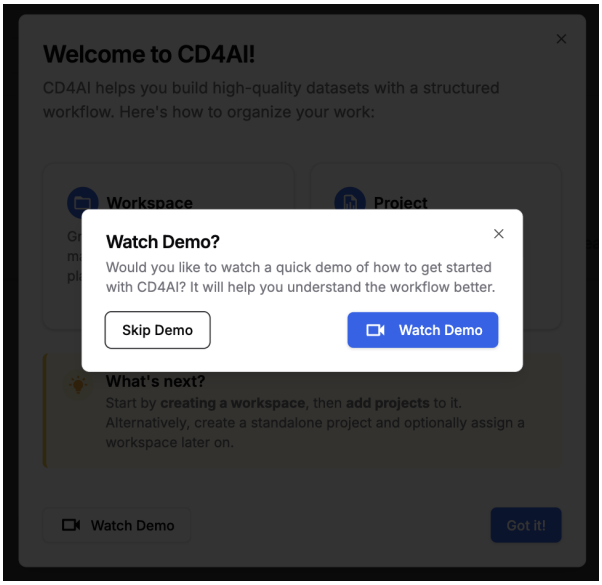

Figure 4.15.: An explicit reminder opens for first-time users who close the onborading wizard without having watched the demonstration video (F-3.1).

F-3.2 – Refinement of the Onboarding Wizard. The onboarding wizard appeared insufficient to fully educate novice users on the mechanisms of CD4AI. Some users criticized a lack of step-by-step instructions for the overall workflow. While the demo videos are designed to address this issue, other participants have suggested providing a more in-depth explanation of the conceptual framework of CD4AI. For that purpose, the help modal containing the task waterfall model (see Figure 4.14), which was initially designed as a simple chronological overview of the steps to be performed, has been refined with more conceptual information to make users understand each task is needed and what the conceptual goals are.

F-3.3 – Redesigned Landing Page. To better set user expectations and provide an initial overview of CD4AI, the landing page was redesigned to present the system's goals and workflow in a concise, visually structured manner. The updated design aims to reduce initial confusion and support a more goal-oriented user experience. The redesigned landing page is structured into four sequential sections, each conveying key aspects of CD4AI. The first section aims to convey a concise understanding of CD4AI's core functionality with a single slogan: *Better Data, Faster. Your AI-Powered Labeling Assistant.* The subsequent sections appear one by one as the user scrolls further down. The second one intends to provide more context of how CD4AI works by highlighting the key advantages of using the system. The third section provides a conceptual overview of the system's underlying mechanisms, specifically the keyword, context window, and archetype tasks. Finally, the fourth section illustrates a sample output, showing how a labeled dataset produced by CD4AI might appear. This layout aims to communicate the system's purpose and workflow at a glance, enabling users to form accurate expectations before engaging with the tool.

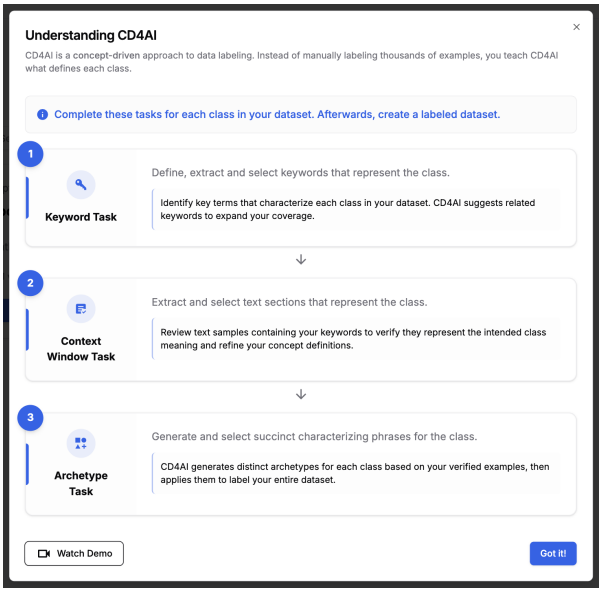

Figure 4.16.: The modal illustrating CD4AI's task workflow was enriched with conceptual guidance to help users understand the purpose and sequence of each task (F-3.2).

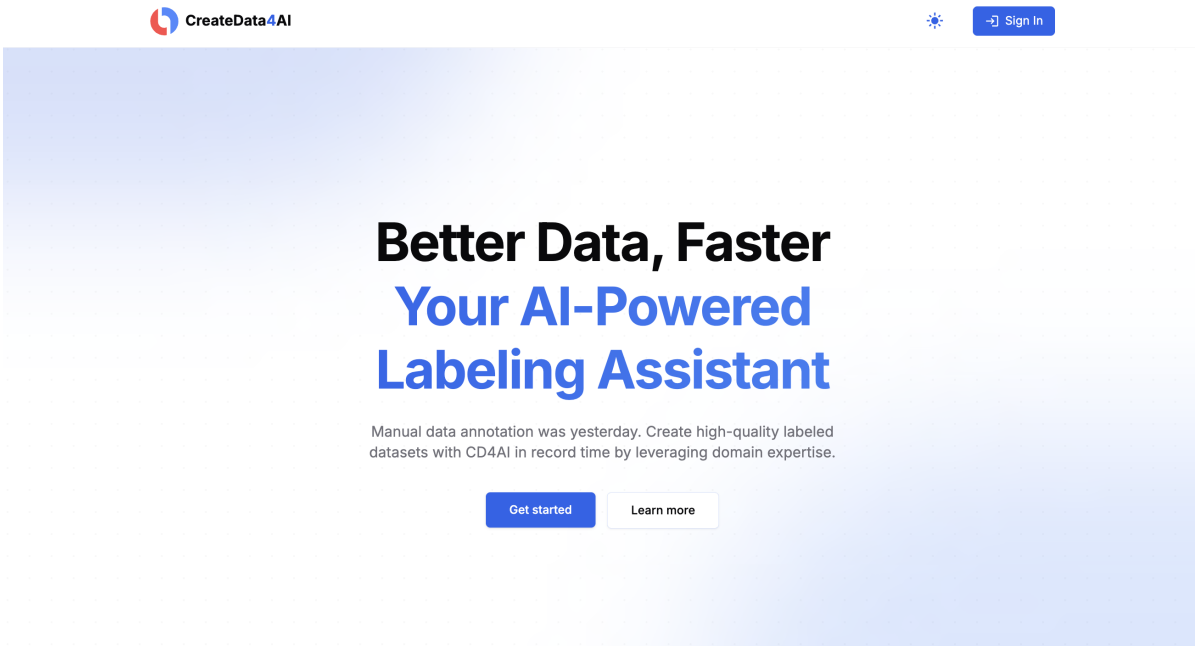

Figure 4.17.: The redesigned CD4AI landing page places more emphasis of CD4AI's core goals. Scrollable content reveals insights into the CD4AI pipeline (F-3.3).

4.5. UCD-4: Final Remote User Testing

This section presents the fourth and final UCD iteration, which aimed to validate whether users could work with CD4AI independently, benchmark the system's usability, and uncover the remaining strengths, weaknesses, and pain points of the application.

4.5.1. Quantitative Feedback

System Usability Scale (SUS). The final user testing yielded a SUS of 72.8, constituting a substantial gain of 9.8 points compared to the last round, and indicating above-average usability. The percentage of participants rating the system higher than the threshold of 68 increased from 40% in the previous round to 67% in this round. Compared to the last round, the range between the smallest and the largest SUS score slightly dropped by five points to 60, and the standard deviation decreased from 21.5 to 19.3.

Table 4.11.: SUS results from the final remote round user testing.

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS Score
P-4.1	4	2	4	1	4	2	5	2	3	3	75.0
P-4.2	5	1	5	1	5	1	5	1	5	1	100.0
P-4.3	3	4	2	1	3	3	3	4	2	4	42.5
P-4.4	4	2	4	1	5	2	3	2	4	1	80.0
P-4.5	3	3	3	3	3	3	2	3	2	4	42.5
P-4.6	3	4	3	4	3	2	3	4	2	4	40.0
P-4.7	3	3	3	2	4	2	3	2	2	4	55.0
P-4.8	4	1	5	1	5	1	4	1	4	1	92.5
P-4.9	5	3	3	2	4	3	4	3	4	3	65.0
P-4.10	3	2	4	1	4	2	4	1	3	2	75.0
P-4.11	4	1	5	2	4	1	5	2	4	3	82.5
P-4.12	5	1	5	1	5	1	5	1	4	1	97.5
P-4.13	4	2	4	1	4	2	4	2	4	2	77.5
P-4.14	5	1	5	1	5	1	4	1	5	3	92.5
P-4.15	4	2	3	1	3	1	4	1	3	2	75.0
Best Possible (BP)	5	1	5	1	5	1	5	1	5	1	100
Mean Diff. to BP	1.1	1.1	1.1	0.5	0.9	0.8	1.1	1.0	1.6	1.5	27.2
Overall Mean SUS Score: 72.8											

A notable difference in the SUS scores among participants with a technical and those with a non-technical educational background can be observed, as shown in Table 4.12. While the responses from non-technical participants yielded a mean SUS score of 58.5, the mean score for technical test users was 21.5 points higher at 80. An independent-samples t-test (assuming equal variances[5]) confirmed that this difference was statistically significant, with $t(13) \approx -2.22$, $p \approx .022$ (one-tailed). With Cohen's $d \approx 1.22$, the observed difference

[5]SUS: $Variance_{non-technical} \approx 336, Variance_{technical} \approx 301$

corresponds to a large effect size. However, given the small sample sizes ($n = 5$ for non-technical and $n = 10$ for technical participants), this result should be interpreted as indicative rather than conclusive.

Table 4.12.: Mean difference to the best possible value for each SUS question.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS Score
All Users	**1.1**	**1.1**	**1.1**	**0.5**	**0.9**	**0.8**	**1.1**	**1.0**	**1.6**	**1.5**	**72.8**
Non-Technical Users	1.6	1.8	1.6	1.2	1.2	1.2	1.8	1.6	2.4	2.2	58.5
Tech. Users	0.8	0.8	0.9	0.2	0.8	0.6	0.8	0.7	1.2	1.2	80.0

User Survey. Table 4.13 shows the mean Likert rating for each survey question. Unlike the SUS, all questions were phrased positively, meaning that a higher score means a more positive rating of the system in the respective category. Again, clear differences between the two participant groups are observable. On average, technical participants rated their experience more positively (mean 4.2) than non-technical participants (mean 3.5), with an overall average of 3.9 across all participants. A two-sample t-test assuming unequal variances[6]) confirmed that this difference is statistically significant, with $t(23) = -4.27$, $p < .001$ (one-tailed).

The primary pain points for first-time users appear to be the conceptual understanding of context windows (Q18), archetypes (Q19), and the theoretical concepts underlying how these tasks are used to create a labeled dataset (Q20, Q21). Although the onboarding wizard was reconfirmed to be not expressive enough as the sole source of onboarding material (Q13), the demonstration videos (Q14, Q15) were appreciated by users. Besides the demo videos, the intuitiveness of workspaces, projects, classes (Q11, Q12), and keywords (Q16, Q17), as well as the look and feel of CD4AI's user interface (Q22) were rated positively. 13 out of 15 (87%) participants (strongly) agreed that using CD4AI for data labeling is more enjoyable than manual labeling (Q23).

Table 4.13.: Mean item ratings for the final remote user testing survey.

Q–	–11	–12	–13	–14	–15	–16	–17	–18	–19	–20	–21	–22	–23	–24	–25
All	**4.1**	**4.2**	**3.2**	**4.3**	**4.0**	**4.1**	**4.2**	**3.7**	**3.6**	**3.5**	**3.6**	**4.1**	**4.5**	**4.1**	**3.9**
Non-T.	3.6	3.8	2.8	3.6	3.8	3.6	4.0	2.8	2.8	3.0	3.0	3.8	4.4	3.8	3.8
T.	4.4	4.4	3.4	4.6	4.1	4.3	4.3	4.1	4.0	3.8	3.9	4.3	4.5	4.2	4.0

Task Completion Rate (TCR). All users except one were able to complete their tasks without any external support and without requesting help via the Prolific contact form, resulting in a TCR of 93%. Only P-4.2 was asked to abort the task and proceed directly to the survey after

[6]Mean Survey Rating: $Variance_{non-technical} \approx .25, Variance_{technical} \approx .09$

archetype generation (i.e., without creating a labeled dataset, which would have been the final step in the test task). The reason for this was that CD4AI's GPU was still busy processing a large number of context windows from the previous study participant, causing P-4.2's archetype generation to be stuck in the queue and blocking the participant from completing the test task. For the sake of comparability, P-4.3 has been excluded from all of the following quantitative metrics.

Task Completion Time (TCT). Overall mean TCT was 32 minutes and 23 seconds. This approximately matches the expected time of 30 minutes. A notable difference is observed between non-technical participants, whose mean TCT was 43 minutes and 13 seconds, and technical participants, whose mean TCT was 26 minutes and 2 seconds. An independent-samples t-test (assuming unequal variances[7]) confirmed that this difference was statistically significant, with $t(7) = -2.90$, $p = .011$ (one-tailed), and a large effect size of Cohen's $d \approx 1.76$. The median values are 31 minutes and 16 seconds overall, 45 minutes and 46 seconds for non-technical participants, and 24 minutes and 54 seconds for technical participants.

Help Modal Views. Table 4.14 shows the average number of times each help model has been viewed. Similar to the last round of user testing, the task help modal has been viewed the most, followed by the introduction help model, indicating that these are the parts where users needed the most help. This is not surprising, as these cover the overall usage instructions for CD4AI. The classes help modal has been viewed the least, as no user decided to click on the help modal again after the initial auto-open. This might indicate that the concept of classes was easily understood by the users, possibly due to the relatively straightforward, example-based demonstration of how to create classes in the introductory demo video. Interestingly, unlike in the last user testing round, technical participants viewed the help modals less frequently on average than non-technical participants, as shown in Table 4.14. The impact of educational background on the total number of help view modals is not statistically significant, as indicated by an independent-samples t-test (assuming unequal variances[8]), with $t(7) = -0.8$, $p = .23$ (one-tailed).

Table 4.14.: Mean number of views per help modal during the final remote user testing. KW = keyword task, CW = context window task, AT = archetype task, EXT = extrapolation

	Intro	**Classes**	**Task Flow**	**KW**	**CW**	**ATG**	**EXT**	**Total**
All Users	**1.7**	**1.0**	**2.3**	**1.1**	**1.5**	**1.6**	**1.2**	**10.4**
Non-Tech. Users	1.6	1.0	3.2	1.0	1.8	2.4	1.0	12.0
Technical Users	1.8	1.0	1.8	1.1	1.3	1.2	1.3	9.6

[7] mean TCT: $Variance_{non-technical} \approx 128, Variance_{technical} \approx 73$

[8] Number of help views: $Variance_{non-technical} = 41, Variance_{technical} \approx 8$

4.5.2. Qualitative Feedback

Largest Difficulties. Three core difficulties when using CD4AI emerged from the survey results. The difficulty mentioned most was the **initial conceptual understanding** of the app and getting started with the task flow. However, of the six participants who mentioned this difficulty, three qualified their statement and concluded that the demonstration videos helped significantly in getting started. P-4.1, for example, stated: "[*The largest difficulty for me was*] first understanding what was a workspace, a project and a class, but once I watched the demo video it all made sense to me". Still, the three others did not mention anything comparable, and stated "a lot of buttons and different areas" (P-4.6) and a lack of explanation "in layman's terms" (P-4.7) as the reason for their frustration. P-4.9 mentioned "the archetype part" as a difficulty. Due to the lack of further detail in the participant's comment, only speculation about the exact difficulties regarding archetypes is possible. The comment presumably refers to insufficient explanation of the concept of archetypes.

Another recurring source of difficulty was the **navigation between tasks** and the clarity of subsequent steps. Four participants reported uncertainty about how to advance within the workflow, particularly regarding the need to click on individual tasks in order to open the corresponding windows. P-4.3 and P-4.11, for example, noted moments of hesitation about what to do next, although P-4.11 emphasized that these issues were resolved relatively quickly once the interaction pattern became clearer.

A further set of difficulties emerged during the **assessment of extracted context windows**. Participants P-4.8 and P-4.13 noted that some of the provided excerpts were too short (sometimes consisting of only three to five words), making it difficult to reliably judge their relevance. Both emphasized that having the option to view more surrounding text would have improved their ability to make accurate decisions. In contrast, P-4.12 did not report conceptual difficulties with this task but highlighted a technical issue: the interface occasionally required multiple clicks when accepting or rejecting items, which slowed down the workflow.

P-4.5 reported an **inability to initiate keyword tasks from the project dashboard**, requiring navigation to the task page instead. Unfortunately, no further details regarding reproducibility and error details were provided ("[*starting the keyword extraction*] from the dashboard just didn't work"). While other participants did not encounter this issue, subsequent manual testing identified an uncovered edge case in the batch-creation workflow implemented in features F-1.6 and F-1.8 as the possible root cause. These features had enhanced the keyword task form to eliminate the need for explicit confirmation of form inputs. However, the implementation did not account for scenarios where users define a keyword task (by selecting a class and entering comma-separated keywords) and attempt to create another task without first pressing *Enter* to confirm the keywords. The issue is illustrated in Figure 4.18. It is likely that P-4.5 was unclear about the necessity of explicit keyword confirmation in this case, as the system fails to auto-confirm the input, resulting in validation errors that prevent task creation.

P-4.14 stated that they had not experienced any difficulties.

Frustrations. Some of the aforementioned difficulties in use were reiterated in the participants' responses regarding their largest frustrations with CD4AI. Two participants emphasized challenges in grasping the system's **conceptual underpinnings**, with P-4.9 specifically pointing to "the archetype part", without going into further detail.

Frustration was also voiced in relation to **task navigation**, as P-4.15 noted that moving between the individual tasks and labeled dataset felt unnecessarily fragmented ("I [...] had to go to a separate tab to find the labeled dataset. I feel like these could all be on the same page for easier functionality").

In addition, frustration about the **transparency of CD4AI's task processing** was mentioned twice, with P-4.1 reporting frustration about the task queue and the waiting times during archetype generation. It is unclear whether the frustration is related to the length of the archetype processing itself (which took roughly two to three minutes during the test sessions), or to a lack of quantified insight into the task progress, similarly to P-3.5's comment in the last round of user testing (see Section 4.4.2). Besides that, P-4.5 expressed uncertainty about whether a task that is processed by CD4AI would continue to run in the background after "closing the window".[9]

Another comment addressed the **demonstration videos**: P-4.3 was left unsure whether a video had finished, and mentioned: "On more than one occasion I sat waiting [*for the video to continue*]".

Two other statements were not directly tied to CD4AI's functionality itself, but rather concerned the test setup, and will therefore be dismissed and not considered further. P-4.13 expressed disappointment about not being able to review the final dataset. Presumably, this is because, in test mode, the invitation to the user survey appeared immediately after starting the process to create a labeled dataset, and thus the user did not have the opportunity to preview the dataset before completing the survey. P-4.14 suggested that it would be helpful to preview data points when selecting a dataset column. While this comment may reflect the participant's unfamiliarity with the test dataset, and although a dataset preview is available in the project information section, a column-level preview during selection could still be valuable for real-world users who are familiar with their dataset.

Finally, five participants (i.e., one third) explicitly stated that nothing about CD4AI frustrated them.

Positive Feedback. Participants highlighted several aspects of CD4AI that they particularly enjoyed. Seven test users praised the **interface design**, describing it as clean, straightforward, and easy to navigate. P-4.11, for instance, emphasized the clarity of the tiered workspace–project–task structure, noting that the transitions between levels "made a lot of sense". Similarly, P-4.10 stated that the interface was "well designed and simple to navigate/use once I understood how the system functioned". Individual elements also received positive attention, such as the keyword selection interface mentioned by P-4.5.

The **demonstration videos** stood out as another key strength. Six participants described

[9] It is unclear whether the participant refers to closing the browser window, or to navigating to another page inside the web app.

them as clear, concise, and helpful for onboarding. P-4.2 and P-4.1 both emphasized their value for users less familiar with data labeling, with P-4.2 noting they were particularly well-suited for beginners. P-4.4 added that the videos provided "good instructions without being overwhelming", while P-4.8 positively mentioned the "ability to move forward or backward between different [*video*] sections instead of having to re-watch the whole video". P-4.11 even remarked that the chosen voiceover contributed positively, as it remained pleasant to listen to throughout.

Besides that, **ease of use** was a recurring theme across responses, being mentioned by four participants. P-4.13 appreciated the "step by step approach" and called it "systematic and logical". P-4.15 particularly emphasized a "relatively small learning curve", suggesting that once familiar with the workflow, the application supported efficient progress.

Beyond these general impressions, certain features of the system itself were highlighted. Two participants expressed appreciation for the **keyword expansion** feature.

Performance was also positively noted by two test users, with P-4.6 mentioning that the system "ran quickly once you'd inputted all the correct information".

Finally, P-4.9 mentioned "the labeling" as something he enjoyed. However, since no further detail was provided, it remains unclear whether this referred to the labeling process itself, the interface supporting it, or the outcome of the task. This comment will therefore be dismissed and not considered further.

Future Recommendations. Four participants suggested a clearer **onboarding experience** and more effective guidance throughout the workflow. P-4.3 proposed the idea of a "pictorial overview of the complete process to be followed, like a flowchart", while P-4.6 recommended numbering the tasks to make their sequence more explicit. In a similar vein, P-4.7 highlighted the importance of a conceptual explanation of what the system is doing. However, considering that the task help modal (see Figure 4.16) automatically opens for all new users and already provides a flow-oriented overview, there is a chance that participants may have dismissed the modal without engaging with its content, or that they did not find it relevant enough for further consideration. The latter might potentially be due to content complexity and cognitive overload. Additionally, while appreciating the demonstration videos, P-4.10 proposed the idea of a guided practice project for first-time users before they engage in the actual workflow.

Three comments addressed the **context window assessment** itself. P-4.5 and P-4.15 expressed uncertainty about how the system interprets a negative decision when rejecting context windows, suggesting either a clearer explanation or an additional "not sure" option. To resolve the issue of context windows being too short, P-4.8 and P-4.14 suggested an option to view a longer version of the context window to be assessed, either by increasing the minimum length for context windows, or by adding a button to expand the context window on demand, or even to view the entire source text

Two further recommendations concerned the **task navigation**. P-4.4 noted that the process of moving between tasks could be made more obvious, suggesting that once a task is completed, the interface should make it clearer how to proceed to the next one. P-4.15 suggested to "make it easier" to initiate key steps such as extracting keywords and context

windows, generating archetypes, and creating labeled datasets.[10] Although no further detail was provided by P-4.15, the participant's comments on the previous survey questions suggest that his recommendation refers to the navigation between single task types.

Two participants suggested nice-to-have **advanced dataset features**. P-4.14 proposed the option to automatically split the dataset into training and test subsets, and to analyze CD4AI's labeling accuracy compared to some ground-truth labels. P-4.13 likewise expressed interest in handling cases where articles might belong to multiple classes[11] and suggested providing a mechanism to adjust classifications.

Beyond these recurring themes, there are two other rather isolated suggestions. P-4.11 was disappointed that the auto-assign feature was disabled in the test and wanted to try it out. This issue is related to the test setup and not the app itself. P-4.12 proposed reconsidering the product name, remarking that "Seedy For AI might not give the best impression". The participant likely interpreted the name *CD4AI* as "Seedy for AI" based on its pronunciation in the demo videos. Due to these circumstances, these comments will be dismissed and not considered any further.

Three participants did not give any future recommendations.

4.5.3. Feedback Implementation

Based on user feedback, three additional features were implemented to address the most commonly reported usability issues and streamline the workflow.

F-4.1 – Context Window Auto-Save. P-4.12 reported that context window selection occasionally felt slow, sometimes requiring multiple clicks on the *accept* button. Manual inspection revealed two causes. First, the time-based auto-save (introduced in F-2.3) could interfere with user input when a decision coincided with the save cycle, producing short-lived race conditions. The time-based approach, originally implemented to avoid disabling buttons during saves, proved less responsive and impractical in real-world use. In F-4.1, the logic was redesigned to a more robust event-based model, where each selection is saved immediately, with input buttons temporarily disabled during the saving process. This eliminates overlapping actions and ensures every user decision is captured without repeated input. The second cause was that keyboard-based inputs were occasionally not registered, despite the corresponding animation being shown. This was resolved by directly binding arrow-key presses to the selection logic, ensuring reliable keystroke recognition and preventing duplicate context windows after a selection.

[10] The reference of P-4.15's comment is an interpretation, as the original response contained incorrect terminology, such as "generate the context keywords" and "start the artifact".

[11] It is assumed that P-4.13's original comment "multiple datasets" is a typo and that the participant was referring to *classes* instead of *datasets*, because only a single dataset was part of the test session.

[12] Interpretation note: 3 (1 + 2) means 3 **overall** mentions, originating from 1 **non-technical** and 2 **technical** participants.

Table 4.15.: Summary of user feedback on CD4AI from the final remote user testing.

Dimension	Theme	# Mentions[12]
Difficulties	Getting Started	**3** (2 + 1)
	Task Navigation	**4** (1 + 3)
	Context Window Assessment	**3** (0 + 3)
	Batch-Creating KW Tasks Not Working	**1** (0 + 1)
	–	**1** (0 + 1)
Frustrations	Conceptual understanding	**2** (1 + 1)
	Task Navigation	**1** (0 + 1)
	Task Processing	**1** (1 + 0)
	Demo Videos	**1** (0 + 1)
	–	**5** (1 + 4)
Positive Feedback	Layout and UI	**7** (2 + 5)
	Demonstration Videos	**6** (3 + 3)
	Ease of Use	**4** (0 + 4)
	Keyword Expansion	**2** (0 + 2)
	Performance	**2** (1 + 1)
Recommendations	Onboarding Experience	**4** (2 + 2)
	Context Window Assessment	**3** (1 + 2)
	Task Navigation	**2** (1 + 1)
	Advanced Dataset Features	**2** (0 + 2)
	–	**3** (1 + 2)

F-4.2 – Auto-Confirm Keywords when Adding Another Task As mentioned earlier, one participant reported that attempting to start a keyword task from the project dashboard was not working. This was likely due to the user being confused about the necessity of explicit manual confirmation of keywords via the *Enter* button before adding another keyword task. As part of F-4.2, an auto-confirm mechanism was implemented, which removed the need to explicitly press *Enter* after inputting a set of comma-separated keywords before adding another task. This mechanism works analogously to F-1.6 and F-1.8.

F-4.3 – Auto-Open New Tasks. A recurring usability issue was that users often failed to realize they needed to open newly created tasks to continue the workflow. To address this, an auto-open mechanism now redirects users directly to the new task (for example, from a keyword task to a context window task or from a context window task to an archetype task). This streamlines navigation and provides better guidance for novice users, reducing their reliance on the onboarding wizard and demo videos. For most task types, the mechanism simply replaces the manual click to open the next task. When multiple keyword tasks are created simultaneously from the project page, users are instead redirected to the project's task dashboard. To accommodate experienced users who may prefer to stay on the parent page, the feature can be disabled in the profile settings (see Figure 4.19). It is enabled by default for first-time users.

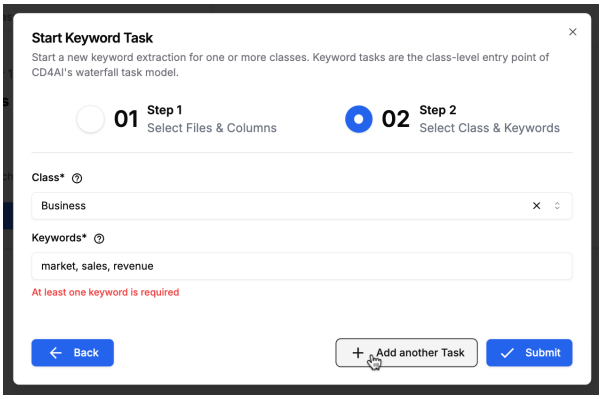

Figure 4.18.: Adding another task does not auto-confirm input keywords, preventing P-4.5 from batch-creating keyword tasks. The issue is resolved via F-4.2.

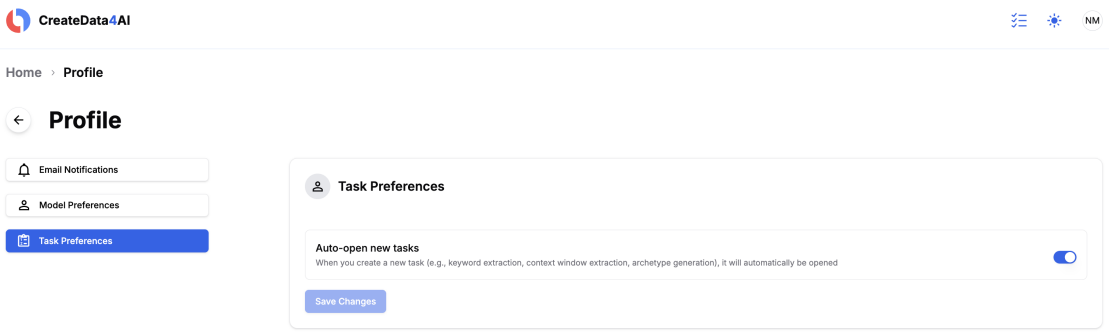

Figure 4.19.: The profile settings interface allows users to disable the auto-open mechanism for newly created tasks (F-4.3).

5. Discussion

Over the course of four UCD iterations, CD4AI advanced from a lab-based prototype to a mature web application. The results of the evaluation and refinement revealed not only measurable improvements in usability and workflow efficiency, but also notable differences in perceived usability across technical and non-technical study participants.

5.1. Contributions to the Research Questions

The iterative process enabled a detailed exploration of each research question, translating insights into practical design and implementation decisions.

RQ1: *What efforts can be taken to reduce manual effort in the CD4AI web application?*
The methods applied during the inspection phase identified several features that are expected to significantly reduce manual effort in the app, operating on two levels. On the one hand, features at the *workflow level* impact how users will utilize CD4AI throughout the overall process. On the other hand, features at the *action level*facilitate fine-grained interactions with the app more efficiently and smoothly.

At the **workflow level**, workspaces are not mandatory anymore to set up a project (F-1.1), which is believed to reduce app startup time, as users are able to reach the *main feature*, i.e., the actual project tasks, more quickly. Additionally, the email notification system (F-1.4) enhances task status tracking efficiency, as the need to actively track tasks is replaced by the option to passively wait for task updates, which can be highly beneficial for long-running tasks such as archetype generation and extrapolation. In-app task statuses can be monitored more efficiently on the project-level task dashboard (F-1.3), which replaces the need to manually dive into each class page and browse through the class's task tree. Lastly, expert users and established teams of collaborators can streamline their workflow by automating tasks such as auto-merging and selection, and auto-starting the next chronological task (F-1.12).

At the **action level**, several mandatory attribute fields were made optional, which enables users to set up their project more quickly and be more autonomous in the administration of their projects. F-1.6, F-1.7, and F-1.8 enabled users to go through the process of starting a keyword task more quickly and more smoothly by making several click sequences obsolete, which were believed to feel unnatural to users. Besides that, tasks can now be directly opened via a URL (F-1.2), eliminating the need to manually browse through a class's entire task list to (re-)open a task, when, for example, refreshing the browser page or searching for a certain task from the task dashboard. The Auto-Save logic from CW tasks was also rolled out to

keyword and archetype tasks (F-2.3), making manually saving the selection results obsolete. Last, auto-opening tasks (F-4.3) remove the need to manually click on newly created tasks.

However, these adaptations do not fundamentally alter the structure of the CD4AI pipeline itself. As the pipeline is laid out as an HITL framework, a certain portion of manual effort is naturally inherent in it, and the degree of automation is limited. Features to optionally auto-run the entire pipeline (i.e., auto-accept all keywords and all context windows) are technically feasible but were not implemented due to the assumed low practical relevance of such a feature. Given the lack of user feedback, manual effort does not appear to be a major usability issue, indicating a high degree of maturity in this area for CD4AI.

RQ2: *How can the CD4AI web application be iteratively optimized for usability?*
Due to its iterative nature, the UCD is considered an appropriate methodology for addressing the second research question in retrospection. UCD-1 allowed for low-cost (in terms of time and effort for validation) incubation of the app, benchmarking the usability against heuristics and best practices, and discovering long-term issues that might go unnoticed in testing sessions with first-time users. In this phase, new key features and more issues were discovered and implemented than in any other iteration. Following the convergence of inspection-based findings, UCD-2 enabled the initial validation of the design against real-world user needs and provided in-depth insights into the thoughts and opinions of first-time users. Building on these insights, it was possible to improve features and design, and tailor it more toward the perspective of novice users. One key adaptation was the integration of an in-app onboarding wizard. In UCD-3, users' ability to independently onboard and use the app was validated. No sheer usability issues emerged, but a lack of initial understanding of how to use the app was observed. Based on these findings, in-app demonstration videos were added. UCD-4 proved users' ability to independently onboard and use CD4AI productively. Besides, final usability was assessed and benchmarked.

The overall level of usability is indicated by the SUS, which combines all aspects of usability in the survey questions. As shown in Table 5.1, SUS scores initially reached 75.6 in UCD-2, a level generally considered above average, but then dropped to 63.0 in UCD-3. This decrease can likely be attributed to difficulties with onboarding and the conceptual clarity of CD4AI's functions. In UCD-4, the SUS increased again by nearly ten points, reaching 72.8. While this remained slightly below the UCD-2 level, the difference between user groups revealed a more nuanced picture. Technical participants consistently rated the system higher across iterations, culminating in a final SUS of 80.0 in UCD-4, which is considered almost excellent. In contrast, non-technical participants rated the system at 58.5, indicating persistent usability challenges.

Despite the overall evaluation, several indicators exist for the subcategories of usability, including effectiveness, efficiency, and satisfaction. Given the high task completion rate of 93% in UCD-4, **effectiveness** can be seen as high. The integration of in-app user guidance was the most essential feature to achieve this, bringing the app from a state where independent usage without external guidance was not possible (UCD-2) to a state where 14 out of 15 participants completed their task independently. The integration of demonstration videos in the final iteration represented the most impactful improvement. Unlike the onboarding

Table 5.1.: SUS scores across testing iterations.

	UCD-2 (In-Person)	UCD-3 (Pilot Remote)	UCD-4 (Final Remote)
All Users	**75.6**	**63.0**	**72.8**
Non-Technical Users	N/A	41.3	58.5
Technical Users	75.6	77.5	80.0

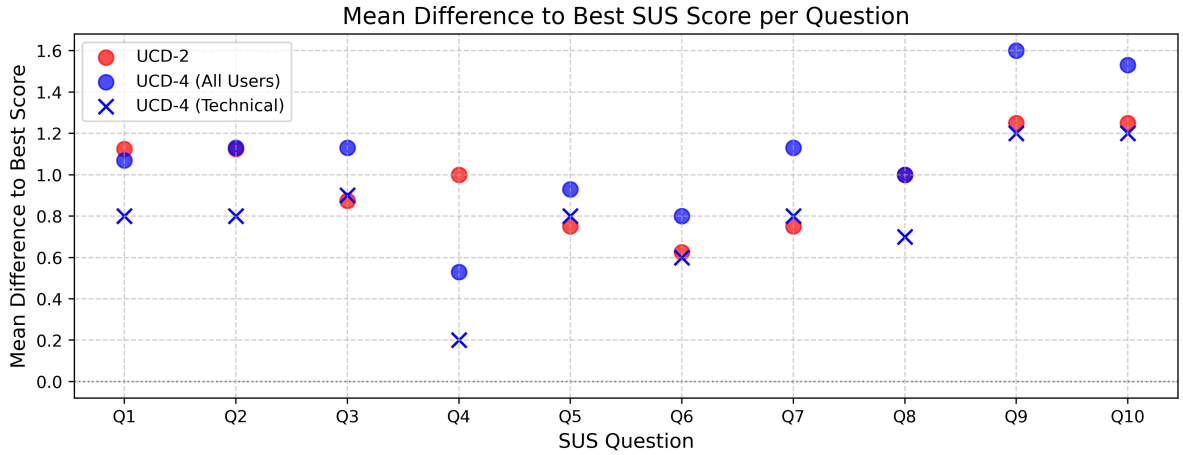

Figure 5.1.: Mean deviation from the maximum possible SUS score for each question across UCD-2, UCD-4, and UCD-4 (technical participants only). Lower values indicate more positive ratings. Note: The UCD-2 and UCD-4 dots are overlapping for Q2, Q8.

wizard, which was perceived as insufficient for a proper introduction experience, the videos provided an intuitive and accessible explanation of CD4AI's core concepts. Their introduction significantly reduced learnability-related problems and demonstrated the importance of aligning onboarding materials with users' actual learning preferences.

High **efficiency** is achieved by minimizing the resources required to produce satisfactory results [17]. In the context of CD4AI, time and mental effort are the primary resources that influence efficiency. The time required to complete the task appears reasonable. The mean task completion time (TCT) of 32 minutes and 23 seconds in UCD-4 represents an increase of approximately two minutes compared to UCD-3. This difference was expected, as most participants likely spent at least six minutes, which corresponds to the total duration of all demonstration videos, watching the provided tutorials. The expert TCT (that is, the time the author of this thesis required to complete the same task, including reading the onboarding wizard and watching the videos) was 17 minutes and 38 seconds. When subtracting the duration of the demonstration videos, the resulting TCTs are 26 minutes and 23 seconds for UCD-4, compared to an expert TCT of 11 minutes and 28 seconds. Some users expressed mild frustration with processing times, although it remained unclear whether this was due to

the actual duration or the lack of visible progress indicators. In terms of cognitive demand, the ratings for mental effort (Q24) and task enjoyment compared to manual labeling (Q23) suggest that participants generally perceived the workflow as efficient and manageable.

Regarding **satisfaction**, positive ratings for Q22 (pleasantness of the UI's look and feel), Q23 (enjoyability of CD4AI compared to manual labeling), and Q25 (recommending CD4AI to others) indicate a generally positive feeling towards CD4AI. 13 out of 15 (87%) participants agreed that using CD4AI is more enjoyable than manual labeling, and eleven (73%) would recommend CD4AI to friends who need to label data. Free-text responses further emphasized the system's attractive design and overall usability. This suggests that, while onboarding challenges remain, users found CD4AI to be a well-designed and engaging tool once familiar with its concepts.

However, all three categories are presumably strongly affected by the **learnability** of the system. Judging from the results, learnability concerns both purely instructional aspects (i.e., teach users how to use the app) and aspects related to the conceptual understanding of the functional fundament (i.e., teaching users the concepts of the CD4AI framework). Learnability from an instructional perspective initially received controversial user ratings. On the one hand, several users (across all UCD iterations) praised the intuitiveness and well-structured workflow. On the other hand, more extensive methods of onboarding and introduction into the task flow were suggested. Overall, as reflected in higher survey scores, positive feedback on the demonstration videos, and a reduced frequency of "onboarding" being named as the greatest difficulty (four of five cases in UCD-3 compared to three of fifteen in UCD-4). These results suggest a high maturity of CD4AI in terms of instructing users on how to use the app. Learnability from a perspective of users' conceptual understanding seems to be improvable, however. While questions Q13 (helpfulness of the onboarding wizard), Q19 (understanding of archetypes), Q20 (intuitiveness of creating a labeled dataset), and Q21 (confidence while completing tasks) all improved between UCD-3 and UCD-4, they still received the lowest ratings overall. This comes with several caveats, however. First, the first aspect ("how to use the app") is seen as more important for first-time users, as this perspective actually allows them to perform tasks independently. Developing a conceptual understanding is possible over time, but only with a clear understanding of how to use the app. Second, the documentation (F-0.6), which is intended to teach users the theoretical concepts of the CD4AI pipeline more thoroughly, was not part of the user testing and might thus eliminate this issue in the future. Third, the usability definition explicitly mentions that usability is specific to the target user in a specified Context of Use. It must be considered that the actual real-world Context of Use differs from the Context of Use in the test sessions. Real-world users will have a more natural inherent goal of what they are trying to achieve, and thus have more extensive knowledge about the concept of domain-specific data labeling.

In summary, the iterative optimization through UCD significantly enhanced CD4AI's usability, particularly for technical users. The primary remaining challenge is lowering the entry barrier for non-technical experts, where onboarding continues to act as the primary bottleneck.

RQ3: *What technical adaptations are required to transform the CD4AI application from a research prototype into an externally usable system?*
Throughout this study, several key adaptations advanced the CD4AI app from a research prototype to a deployable system. Public deployment and infrastructure configuration now allow external accessibility, as confirmed through public user testing. The integration of the archetype-based document classifier represents a major milestone, incorporating the latest and most promising research outcomes from the CD4AI project directly into the operational system. Supporting features such as the onboarding wizard, demonstration videos, and user documentation enable productive use by external users. Future extension and maintenance are facilitated by the retrospective identification of non-functional requirements, documentation of the Context of Use, and formalization of the data model. Current scalability is constrained by a mutex lock that queues tasks, allowing only one GPU-intensive operation (such as archetype generation or extrapolation) to run at a time. Future work should explore options to auto-scale the lock and optimize GPU utilization.

5.2. Reflection

Throughout this thesis, several key observations, challenges, and insights have emerged. At the same time, the study design entails certain limitations.

5.2.1. Challenges and Learnings

Adapting the UCD process to an already existing application introduced methodological and conceptual challenges. Traditionally, UCD is intended to progress from low-fidelity to high-fidelity prototypes, allowing user needs to shape the product architecture from the outset. In this case, however, CD4AI already existed as a working prototype with no available documentation, requiring the definition of requirements, Context of Use, and personas to be done retrospectively. The structure of the underlying NLP pipeline largely predetermined the overall workflow and interaction design, leaving limited room for fundamental reconfiguration. Consequently, the UCD activities in this thesis primarily focused on improving usability and interaction efficiency rather than exploring alternative conceptual architectures. Nonetheless, user feedback throughout the iterative process indicated general satisfaction with the workflow structure, suggesting that the existing pipeline design aligns well with users' expectations and mental models.

During the inspection phase, it became evident that heuristic evaluation, while valuable for early-stage feedback, provides broad guidance and allows for flexible interpretation. In-person testing revealed the irreplaceable value of field observation. Participants interacted with the system in ways that were not anticipated during the design phase, such as interpreting the task timer as a countdown or mistaking the cancel button for a close button. These observations highlighted the limitations of desk-based evaluations and underscored the necessity of testing with real users. Remote user testing presented a different type of challenge: balancing the level of preliminary instruction with the assessment of true in-app usability. Extensive instructions

reduced confusion but risked creating artificial conditions, while minimal instructions more accurately reflected real-world use but left (especially non-technical) participants uncertain. This tension proved to be a recurring consideration in the setup of remote test sessions.

Moreover, UCD-3 highlighted the fundamental role of learnability. While effectiveness, efficiency, and satisfaction are central components of usability, meaningful evaluation of these dimensions was not possible without sufficient learnability. This realization led to an additional iteration, emphasizing that learnability forms the baseline upon which all other usability factors depend.

The significant difference in perceived usability observed between technical and non-technical users has an impact on the hypothesized Context of Use, showing a clearer picture of the personas. Leimeister [42] introduced the concept of Collaboration Engineering, which serves as a systematic framework for successful collaboration processes in organizations, differentiating between three types of roles. In recurring high-value processes, the *Collaboration Engineer* is responsible for designing and documenting the overall process, which is then transferred to the *Facilitator*. The Facilitator is responsible for planning, structuring, and implementing the collaborative team process. The actual process is then conducted by a group of *Practitioners*, i.e., a group of task experts in the respective field. Transferring this framework to CD4AI, technical data experts will presumably take on the role of the Facilitator, and are thus the driving forces behind the labeling process with CD4AI, and can be seen as the main users of the app. In practice, this could be reflected in their administration of the projects and task flow. On the other hand, domain experts, often presumed to have a non-technical background, could then be seen as Practitioners who have only selective touchpoints with CD4AI, such as performing keyword and context window tasks. When participating in the workflow selectively, this approach might enable a more gradual understanding of the system and counteract a feeling of overwhelming when confronted with the entire task flow at once, as indicated by non-technical participants' ratings of Q9 and Q10 in the final SUS survey. Effective collaboration and intra-team communication can foster an understanding of more complex conceptual concepts, such as context windows and archetypes (Q18, Q19). Still, the poor rating of these questions should be seen as a signal to further improve the in-app explanation of these concepts.

An additional observation concerns the absence of explicit positive user feedback on most newly implemented features. Given that minor defects in earlier versions, such as issues with the auto-save mechanism, immediately triggered user responses, the lack of comments on recent implementations can be interpreted as an indicator of satisfactory functionality and seamless integration. While the overall positive feedback on keyword-related features suggests successful design improvements, evaluating the impact of individual features would require targeted survey items or dedicated usability probes. However, such a fine-grained evaluation would have expanded the study's scope considerably in terms of effort, time, and cost. Therefore, within the given methodological constraints, the absence of negative feedback may be reasonably regarded as a proxy for feature acceptance and stability.

Another key observation is that neither of the iterations involving remote user testing (UCD-3 and UCD-4) revealed as many fine-grained usability issues and new features as the

in-person session in UCD-2. This may be attributed to two considerations. First, the format of think-aloud sessions enables a continuous feedback loop, where even minor issues may be found with minimal effort (participants verbalize their thoughts in real time, allowing immediate detection of difficulties or confusion). The remote sessions, in contrast, rely on users recalling issues retrospectively, while the written feedback format causes a higher barrier than the verbal form. On the other hand, however, Nielsen's formula to estimate the portion of detected usability issues [23, 24] (see Section 2.2.3) suggests that with eight participants in UCD-2, already approximately 95% of usability issues were detected. Although this figure is only an estimate and should not be regarded as conclusive evidence, it suggests that the majority of usability problems had already been identified after UCD-2 under the given circumstances. The subsequent iterations (UCD-3 and UCD-4) involved five and 15 participants, respectively. While this falls slightly short of Nielsen's benchmark of 20 users for quantitative studies, the sample size remains appropriate given the prior qualitative iteration and the diminishing returns of additional participants.

Direct comparison of SUS scores between CD4AI and the tools presented in Section 2.2.4 is challenging due to domain-specific differences. Overall, the achieved SUS score of 72.8 (80.0 for technical users) aligns with the range typically observed in user-centered design evaluations. The closest qualitative similarities are observed in the evaluation of the STAT tool [34]. As in this thesis, the authors of the STAT study reported challenges related to user guidance and task clarity when extending their evaluation to external or remote participants. Despite improvements to STAT's in-app guidance and task complexity following a preliminary external pilot test with a low task success rate, the subsequent SUS score still indicated below-average usability. Compared to earlier in-person sessions, the researchers observed a significant drop from 81.1 to 55.7, similar to the decrease from 75.6 to 63.0 experienced in UCD-3 of this thesis. Further enhancements to user guidance led to an improvement of approximately 18 SUS points in the final iteration, resulting in a final score of 73.8, which is comparable to CD4AI's 72.8. Unlike CD4AI, the authors did not report any usability differences between participant groups. The SUS standard deviation for STAT ranges between 9.8 and 13.8 (with one exception of 20.1 in the iteration where the SUS dropped), while the standard deviation for CD4AI's SUS ranges between 18.5 and 21.5. Despite recruiting so-called *master workers*, that is, participants with a record of consistently high accuracy and reliability across previous MTurk tasks, the authors excluded six of the 20 participants from the final survey after reviewing their annotation results and response quality. In comparison, only one of 21 recruits was disqualified across UCD-3 and UCD-4 in this thesis, suggesting that recruiting *Qualified AI taskers* via Prolific (see Section 3.4) was a sound choice.

The findings from this study suggest several design implications that may extend to human-in-the-loop (HITL) systems in general. HITL workflows inherently alternate between phases of user action and automated processing, which creates a need for clear communication about responsibility. Users must always be aware of whether they are expected to act or wait for the system. In CD4AI, this was addressed through status banners (F-1.5) that explicitly signaled system activity and user responsibility. Moreover, in parallelized HITL environments, designers should support efficient multitasking to minimize idle time. During the in-person

user testing (UCD-2), the implemented task dashboard (F-1.3) proved effective in this regard, allowing users to monitor progress and continue work across CD4AI tasks simultaneously as processing occurred in the background. Another recurring challenge concerns the limited transparency of AI-driven workflows. Unlike purely manual systems, users may perceive system processing in HITL applications as a black box, which can impede their conceptual understanding of the pipeline. It is essential to help users develop an intuitive understanding of how their input is related to the processed output. Furthermore, CD4AI was intentionally designed to support exploratory and iterative experimentation with different keywords and context windows during the development of context rules (i.e., archetypes). Fang, Alqazlan, Du Liu, et al. [32] observed a comparable usage pattern in their evaluation of a HITL topic modeling tool, indicating that such design principles may be relevant beyond this specific application and could potentially extend to other HITL systems. Although not explicitly examined, no signs of user distrust toward the CD4AI system were observed during this study. This aligns with previous findings [31, 15], which suggests that trust issues tend to be less pronounced in HITL systems than in more autonomous AI applications.

5.2.2. Limitations

When conducting the UCD process, trade-offs had to be made between effort and depth of qualitative insights, causing several study limitations

Interdisciplinary team collaboration is a central principle of the UCD framework [5]. Especially the design validation using inspection methods is generally considered most effective when performed by multiple evaluators with substantial HCI expertise [19]. The UCD process in this study, including all employed inspection methods, was, however, solely conducted by the author of this thesis. Involving trained multi-disciplinary teams of experienced HCI researchers, software developers, UI designers, and domain experts in future design iterations could yield additional usability improvements.

User testing focused on time-bound sessions with novice users. The test task in the user testing sessions was designed to reflect CD4AI's core user journey. Collaboration features, such as inviting team members, creating selection tasks, or merging results, could not be tested. Extended and repeated use of CD4AI in realistic, multi-task environments may reveal additional findings, both positive (e.g., recognition of long-term value) and negative (e.g., usability issues arising from frequent use).

Within the test sessions, the primary focus of the evaluation was on the ergonomics and usability of the CD4AI interface, rather than its actual usefulness in authentic scenarios. However, the Technology Acceptance Model, a widely used framework for explaining user adoption of information systems, suggests that a user's actual use of a system is not only determined by the perceived *ease of use*, but also by the perceived *usefulness*. The latter refers to the extent to which an individual believes that using the system improves their job performance [1]. In the context of CD4AI, the usefulness is likely influenced by factors such as the quality of the generated datasets and the tool's integration into existing workflows and organizational structures. Testing CD4AI in an authentic real-world case study may reveal additional system requirements, as observed in the user-centered study of the MedRec tool

[35], for example (see Section 2.2.4). This also includes the testing of CD4AI's underlying NLP engine in terms of performance and accuracy.

Furthermore, while the SUS is a generally popular tool for measuring usability and has several advantages, such as its versatility, the ease of interpreting its score, and comparability against benchmarks, its generality comes with limitations. Alternative evaluation tools can provide more detailed insights into specific areas of interest. The NASA Task Load Index (TLX), for example, is designed to measure perceived workload, including mental demand, effort, frustration, temporal demand, and performance [43]. The User Experience Questionnaire (UEQ) captures both pragmatic and emotional aspects of user experience, such as efficiency, perspicuity, dependability, stimulation, and novelty [44].

The quantitative iterations (UCD-3 and UCD-4) involved a total of 20 participants, comprising five in UCD-3 and 15 in UCD-4. This sample provided a solid foundation for analysis, although a slightly larger pool of 20 to 30 participants per iteration could have improved generalizability [19, 25]. In particular, the relatively small subgroup sizes for technical and non-technical users limit the statistical significance of observed differences between these groups. Therefore, as mentioned before, these differences should be interpreted as indicative rather than conclusive. Furthermore, because participants in the external testing were compensated, payment-related bias may have influenced their responses. The number of participants in the qualitative iteration (UCD-2) can be considered sufficient [24], but their professional proximity to the research project may have introduced unconscious bias, as well. These circumstances could have influenced their willingness to critically evaluate the system, despite instructions to provide honest feedback.

The Context of Use framework proposed by Maguire [22] provides a holistic and in-depth structure for capturing various aspects of user interaction. In this thesis, the Context of Use was defined retrospectively and under limited resource conditions. While this approach provided a solid understanding of the main user groups and their goals, direct contextual inquiry could have provided more detailed insights. Iterating on the Context of Use is a natural part of User-Centered Design (see Section 2.2.3); future work could therefore employ additional elicitation techniques such as interviews, field observations, or participatory workshops to refine the understanding of personas, their goals, and their needs.

Finally, the results of this thesis leave two issues open with respect to RQ2 and RQ3. The final SUS score for non-technical participants (58.5) indicates that usability has not yet reached a satisfactory level. Depending on the future direction of the CD4AI project, such as whether non-technical participants are considered primary users, additional efforts should focus on improving usability for this group. Second, the scalability of the CD4AI web app is not yet fully guaranteed. The current task mutex lock for GPU tasks allows only one GPU task to run at a time. Future work should explore options for a dynamic lock design, enabling the CD4AI NLP engine to execute as many parallel GPU tasks as possible.

5.3. Outlook and Future Recommendations

The findings and limitations of this thesis point to a range of potential directions for both functional enhancements and methodological exploration.

5.3.1. Functional Outlook

Based on insights from user testing, several functional extensions could further improve CD4AI's usability and better align the system with user needs. Table 5.2 summarizes a number of concrete feature ideas raised during the evaluations.

Table 5.2.: Feature suggestions for future development of CD4AI.

Feature	Description	Justification
Context Window Expansion	Option to extract more surrounding text for individual context windows	Users in UCD-4 noted that some windows were too short to classify with confidence
Context Window Assessability	Make it clearer how to handle context windows that cannot be judged, e.g., signaling that dismissing (swiping left) is the correct action	Improves usability by making the workflow for unjudgable windows more intuitive
Label-Only Mode	Apply previously generated archetypes to new, conceptually similar datasets without regenerating them	Supports scalable annotation and saves users from repeating the full archetype generation workflow, particularly for expanded datasets
Column Preview	Preview of column content when selecting a `.csv` column for a keyword task	Supports smoother workflows and reduces the need to recall the file structure externally
In-App Notifications	Notifications about task status directly in the application	May be more effective than email, particularly for short-running tasks
Email + Password Sign-In	Ability to create a native CD4AI account without OAuth	Could reduce entry barriers by allowing anonymity

Beyond these specific additions, several broader directions emerged. First, further improvement of the onboarding experience remains important, particularly for non-technical users. While demonstration videos significantly improved learnability in UCD-4, onboarding could be made more effective by incorporating interactive elements. One promising option is a guided sample project that walks first-time users through the task flow, as suggested by participant P-4.10. Moreover, the CD4AI documentation, introduced only after UCD-4, may provide further support in explaining concepts in accessible terms.

Second, multiple participants expressed interest in more advanced data-related functionality. Suggestions included a built-in data explorer to better understand datasets, as well as an

evaluation mode that allows users to manually classify a subset of data and benchmark CD4AI's labeling accuracy against it. While such features could help foster trust and strengthen CD4AI's role in the data-labeling pipeline, they would also expand the system's functional scope into areas already covered by standard spreadsheet-based productivity tools. Whether CD4AI should evolve toward a more holistic data-processing environment or remain focused on its core labeling workflow, therefore, warrants careful consideration.

Third, possibilities for enabling users to gain deeper insights into task processing may be explored. As discussed earlier, user concerns regarding waiting times for remotely processed tasks could potentially be mitigated by setting clearer expectations. This may be achieved through more transparent progress feedback, such as upfront or real-time estimations of processing time, or visual indicators like progress bars and percentage trackers.

Last, the scalability and flexibility of the technical architecture require attention. Currently, the task queuing system is optimized to accommodate the constraints of the underlying GPU hardware, allowing only one archetype or extrapolation task to run at a time. Future work could develop a more adaptive mechanism that dynamically manages GPU resources. Likewise, integration of the best-performing classification algorithm from ongoing CD4AI research should be pursued, with attention to possible trade-offs. Finally, current support is limited to two languages and one embedding model per language. Expanding to multiple embedding models and additional languages could increase applicability; however, such extensions would need to strike a balance between performance, complexity, and maintainability.

5.3.2. Methodological Outlook

Future usability evaluations should place stronger emphasis on collaboration features. In particular, testing should assess effectiveness, efficiency, and satisfaction when working in teams, covering the full process from inviting members to a workspace to splitting tasks and merging results. Testing could be conducted with reference to the key concepts of the Collaboration Engineering (see Section 5.2.1). This framework postulates that the facilitation effort required during collaboration should be kept as low as possible, while maintaining consistent effectiveness of the group process. In the context of CD4AI, this implies that users should be able to coordinate and execute collaborative labeling tasks without extensive guidance or external moderation. Future studies could therefore investigate how well CD4AI enables self-sustaining collaboration, for example, by analyzing whether recurring collaboration patterns emerge naturally through the system's interface and workflow design, or whether end users feel the need to further apply domain-specific collaboration engineering. For this purpose, it may also be useful to compute a separate SUS score specifically for collaboration features, allowing direct comparison with the single-user journey. Such studies, however, will require greater planning and coordination efforts due to the simultaneous participation of multiple users.

Beyond collaboration, further evaluation should be extended in both scope and duration. With a final SUS of 72.8 and a task success rate of 93%, CD4AI can be considered mature for short-term usability and novice user enablement. Yet, as noted in Section 5.2.2, longer-term usage may reveal more nuanced insights into usability and automation potential. Such long-

running user studies may also help validate or further conceptualize the possible features outlined in the previous section. These assessments could be supported through in-app surveys and periodic feedback. Complementary quantitative monitoring, such as tracking task completion times, user retention, and error rates, would enable a more comprehensive understanding of usage patterns and functional quality. In addition, continuous model evaluation (particularly in domain-specific scenarios) remains an important methodological task, although this would require new features for ground-truth collection or benchmarking against alternative labeling models. Expanding the scope and duration of user testing does not necessarily require increasing the number of participants or scaling the application's infrastructure. Fang, Alqazlan, Du Liu, et al. [32], for instance, conducted two case studies by inviting real-world users to complete authentic tasks within their application, which yielded additional insights into usage patterns and usability challenges. Similarly, future evaluations of CD4AI could involve a small number of genuine users (or even just a single user) whose authentic goals align with the system's intended purpose, thereby providing richer, contextually grounded feedback than that obtained from recruited study participants.

Trust is a particularly relevant aspect in human–AI collaboration, especially when systems such as CD4AI are deployed in high-stakes domains like healthcare or when their outputs are used to build downstream AI models. In such contexts, users must not only find the system usable but also reliable and transparent. Future research could therefore examine how users develop trust in CD4AI and which design features may strengthen it. Notably, the HITL design of CD4AI may already provide a foundation for trust by keeping humans involved in key decision stages. Pailian and Li [15] argued that HITL systems are particularly suitable for low-trust environments, as the human role can mitigate uncertainty and increase perceived reliability. This hypothesis is additionally supported by Smith, Kumar, Boyd-Graber, et al. [31], who did not observe any indications of distrust among users during user-centered evaluation of a HITL topic modeling tool.

Furthermore, as of 2024, over 60% of website traffic worldwide originates from mobile devices [45]. Many CD4AI interface components, including the context window panel with a mobile-friendly swipe function, are already optimized for mobile use. Future research could investigate whether and how users utilize CD4AI on their mobile devices. Future design efforts might focus on enabling mature mobile usability.

Finally, future research should broaden its focus from usability to overall user experience (UX) of CD4AI, guided by the Technology Acceptance Model [1]. This concerns the entire journey, from before to after the system's usage. Before use, studies could examine the alignment of the hypothesized context with real-world practices, user awareness, and expectations. During use, research may focus on entry barriers, the system's support for importing real-world raw data, and the extent to which users experiment across different task types, including keyword, context window, and archetype tasks. After task completion, investigations could assess user satisfaction with labeling accuracy, identify common usage patterns, and determine the system's potential to support downstream activities. Such a phased approach would provide comprehensive insights into CD4AI's usability and real-world applicability.

6. Conclusion

This thesis makes significant contributions to the CD4AI research project by advancing the usability, maturity, and practical applicability of the corresponding web application. The evaluation and validation against both renowned usability heuristics and real-world user requirements, conducted across four iterative cycles, resulted in 37 targeted system adaptations. Notable enhancements included a transparent task tracking system, several improvements to the keyword panel, and enhancements to in-app navigation.

Above-average usability was demonstrated, as reflected in a final SUS score of 72.8, with technical users rating the system even higher (SUS score of 80.0). A vast majority of test users agreed that using CD4AI is more satisfactory than manual labeling (87%), and would recommend CD4AI to friends who need to label data (73%). A high task completion rate of 93% in the final iteration attests to the proper maturity, enabling users to independently navigate and complete tasks. In this context, learnability, as supported by the onboarding wizard and demonstration videos, emerged as one of the key factors contributing to user satisfaction. Low usability ratings among non-technical participants (SUS score of 58.5) can be partly attributed to the artificial test setup; however, this issue warrants further exploration in the future. The system was successfully advanced from a research prototype to a mature, externally deployable application. Public deployment and infrastructure configuration now allow external accessibility. The integration of the archetype-based document classifier represents a major milestone, incorporating the latest and most promising research outcomes from the CD4AI project directly into the system, thereby enhancing its practicality and usability. Additionally, retrospective identification of non-functional requirements, formalization of the data model, and documentation of the Context of Use provide a foundation for future extension and maintenance. Infrastructural aspects currently limit the scalability to a larger user base.

Furthermore, functional and methodological recommendations were derived from the results, guiding future development and supporting broader adoption of CD4AI in real-world environments. While CD4AI has reached a high level of usability and readiness for deployment, some functional refinements remain, such as optimizing task queuing, GPU utilization, context window management, and onboarding. Future research can build on the methodological insights by conducting longer-term, contextually grounded user studies that assess collaboration features, mobile usage, trust development, and overall user experience across the task lifecycle. Such studies would provide richer, real-world insights into system performance and user behavior, supporting evidence-based design decisions and sustainable adoption. More broadly, the findings of this thesis can inform future research on effective and user-centered design practices for human-in-the-loop systems.

A. Raw Results

A.1. Quantitative Results

Table A.1.: Survey responses and task completion times (TCT) in UCD-3.

	Q11	Q12	Q13	Q16	Q17	Q18	Q19	Q20	Q21	Q22	TCT
P-3.1	5	5	1	4	4	3	3	3	4	4	–
P-3.2	4	4	2	4	5	3	2	4	2	5	21 min 54 s
P-3.3	5	5	5	5	5	5	5	5	5	5	20 min 7 s
P-3.4	1	2	2	2	4	4	2	1	2	2	56 min 1 s
P-3.5	5	4	4	5	4	4	3	3	2	3	19 min 45 s

Table A.2.: Survey responses and task completion times (TCT) in UCD-4.

Q–	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	TCT
P-4.1	4	5	3	5	4	5	4	5	3	4	4	4	5	5	5	51 min 3 s
P-4.2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	–
P-4.3	4	3	2	4	3	3	4	2	3	3	2	4	4	2	3	24 min 25 s
P-4.4	5	4	4	4	5	4	4	2	4	5	4	3	5	4	4	45 min 46 s
P-4.5	3	3	4	3	4	3	5	2	3	2	3	5	5	4	4	31 min 32 s
P-4.6	2	3	1	4	3	3	4	2	2	2	2	4	4	3	3	56 min 13 s
P-4.7	4	4	2	2	3	3	3	3	2	2	2	3	3	3	3	31 min 28 s
P-4.8	5	5	4	5	5	5	5	5	5	5	5	4	4	4	4	29 min 2 s
P-4.9	3	4	4	4	4	4	4	4	4	4	3	4	4	4	4	34 min 25 s
P-4.10	5	4	2	4	2	4	3	5	4	4	3	4	5	5	4	17 min 54 s
P-4.11	5	4	3	5	5	4	4	4	5	4	4	5	5	4	3	31 min 4 s
P-4.12	5	5	4	5	5	5	5	5	4	4	4	4	5	5	5	24 min 54 s
P-4.13	4	4	4	4	4	4	4	4	3	2	4	4	4	4	4	41 min 45 s
P-4.14	5	5	4	5	5	5	5	5	4	5	5	5	5	5	4	17 min 6 s
P-4.15	3	5	2	5	3	4	4	2	3	2	4	4	4	4	4	16 min 41 s

A.2. Qualitative Results

Table A.3.: Free-text survey feedback in UCD-2.

Question	Responses
What parts of CD4AI (e.g., its idea, concept, app) are still unclear to you, or are you still not understanding?	While I generally understand most of it, I think the app definitely needs user explanations of the underlying idea and the necessary workflow steps I think the whole process could be explained within CD4AI to understand the necessary steps. I understand it but a novice maybe would not. I understood everything All is clear
What did you enjoy most about the app?	The ease of use and interactivity UI and overall process The UI and the buttons with the clear descriptions Pretty design Very intuitive and smooth interface, option to automatically assign classes
What frustrated you about the app?	extrapolation button was hidden in results. I don't expect it there, since it starts a next step Nothing, except maybe having to constantly refresh the page, but this is obviously just a temporary issue, so it doesn't count :) Nothing i can recall Nothing Sometimes the navigation was a bit confusing
Do you have any comments, suggestions or recommendations for the future?	Guided tour for the user, simple documentation, email login Nothing i can recall Something like a guide or more tips and guidance would be useful for the first time use. Some buttons gray color made it seem like they were disabled. Selecting a model is not necessary if by now there is only one model I would like to have consistency in showing how long tasks in each step etc. take (display it for all different tasks running) Add even more info buttons with descriptions

Table A.4.: Free-text survey feedback in UCD-3.

Question	Responses
Which difficulties did you face when using CD4AI?	I think getting the hang of it for the first time on what to do & how to proceed and all... The tool seemed very easy to use. The actual instructions were very minimal though. The wizard helped me go through all the steps pretty easily, but I wasn't entirely sure what the end goal was going to be so the steps all felt a bit separate to me. Nothing the instructions given was clear and easy to follow the instructions weren't clear enough, i had to guess a lot There wasnt a really clear overall explanation of what and how the system worked adn what one was trying to acheive. With no experience prior to this, it is hard to really tell if what I was doing was right.
Which aspect(s) frustrated you about CD4AI?	Figuring out what's the next step as I couldn't exactly find the help guide initially. The tool was very intuitive. The most frustrating thing was I was going through the several steps without clearly knowing what I was trying to achieve in each step. I do the general goal was to tag sports/business articles but the steps to get there felt like I was just being walked through do a pretty set number of individual steps. Nothing it just assumed i know how to do everything even though i had never used it before There was a reasonable wait for teh system to perform certain functions. Since there was no indication of any sort of progress it made it a little frustrating just having to wait.
Which aspect(s) did you enjoy about CD4AI?	I think once you get the hang of it, the step-wise flow looks cool and fluid. The tool was very intuitive. Even when I wasn't clear what I 'Had' to do next the tool lead me through it nice and easily and clicking the '?' always got me onto the next step easily if I didn't immediately know what I wanted to do next. For example I wasn't sure where the data was going to come from for these articles (I don't think the instructions mentioned it) but when I got to the file/data to load the select list only contained a single csv file (I think) so it was obvious at that point I needed to click on that. The simple User Interface teaching myself how to use it I certainly like the idea and could see it be very useful.
Which ideas or advice for future improvement of CD4AI do you have?	A initial short step-wise instructional video would help someone who is traversing the portal for the first time move through easily. The tool seemed great. For this particular task, I think a brief introductory document that highlights all the steps that you will be going through to get to the 'labelled dataset' would be useful. As I've already said - I was just going through each step individually one a t a time and not quite sure what would be coming up next. The '?' help was nice and clear what you needed to do next. When I failed to label the data using just 'Sports' I guessed then that I had to finish the 'Business' class to be able to complete that - And subsequently get sent to this survey. For me i think it is perfect give a full example first, even consider a little instruction video, and it would be so much easier If I were to be using this on a regular basis, I think I would need to have a deeper understanding of what I was trying to acheive and how the system worked (especially in relation to the results acheived). Probably just need practice and/or a further demonstration.

Table A.5.: Free-text survey feedback in UCD-4 (difficulties and frustrations).

Question	Responses
Which difficulties did you face when using CD4AI?	The initial introduction to the platform, but the demo videos helped immensely. Only thing was first understanding what was a workspace, a project and a class, but once I watched the demo video it all made sense to me It wasn't always clear what the next steps were. It was not clear that to go to the next task in the class you click on it, so I some time trying to find out how to create context windows because it wasn't clear you needed to click on Keyword Extraction to then find it. there were 2 ways to start the keyword task, but one of them from the dshboard just didnt work. I found it hard to understand how to get going with it, there are a lot of buttons and different areas and it isnt very clear which one you need to start with first. Understanding what I was doing in layman's terms When running through the "Extracted Context Windows", some of the extracts were hard to judge since some of the Extracts only consisted of 3-5 words, not giving you enough context to judge the Extract. The Archetype part I had some slight difficulty understanding the precise flow of the application as this was my first time using it. Viewing the guide videos a couple of times was helpful I did not really face any difficulties once I got going. I guess guidance on the "20-30 context windows" could have been a little more clear. I remember not knowing exactly what to do next after selecting my initial keywords, but I found out relatively quickly. No difficulties with the product itself. If anything when accepting the contents windows (Accepting or Rejecting) It was a little slow to move to the next one, you often had to click accept twice for it to continue to the next one. The short sentence fragment was sometimes too short - it would have been nice to be able to optionally view additional text around that sample to better determine context I did not face any real difficulties when using CD4AI. Everything worked great! I was confused that I had to click on the task itself to open up a new window to be able to generate the context windows.
Which aspect(s) frustrated you about CD4AI?	The queue and wait time for generating the archetypes. Nothing frustrated me about using CD4AI It wasn't always obvious when the videos had finished. On more thsn one occasion I sat waiting. Nothing frustrating as such, it was just all quite new and took a bit of time to get used to, but the videos were very helpful. It wasnt clear if closing the window would cancel the progress or let it run in the background The context windows didnt contain enough information to always know what the extracted piece related to so hard to confirm if it was relevant or not. Not really understanding what I was doing conceptually The "Extracted Context Windows" section was probably the only section that was close to frustrating for the same reason as the previous question of not having a large enough Extract to judge the Context, however the system seems to do ok with its labelling regardless. The Archetype part Nothing N/A. Nothing I didn't get to see and assess the final dataset - more a frustration with the study than with CD4AI Perhaps, when initially selecting the dataset text column, I would like to see I am selecting the right column, it could show examples of data points in the column. Another thing is, when pressing tick or cross to select context windows - perhaps that could be a little more responsive, but that is only a minor point. After finding the context window generation that was quite hidden. I then had to go to a separate tab to find the labeled dataset, I feel like these could all be on the same page for easier functionality.

Table A.6.: Free-text survey feedback in UCD-4 (enjoyed Features and recommendations).

Question	Responses
Which aspect(s) did you enjoy about CD4AI?	The straightforward layout and demo videos. I mostly enjoyed the Demo videos and I think these are great for people who aren't that familiar with data labelling I liked the way it expanded the list of keywords to include similar ones. The videos gave good instructions without being overwhelming. The keyword selection UI was nice The system ran quickly once you'd inputted all the correct information. The demo videos were clear The UI generally felt good, each to navigate and all the sections were clearly marked off or sectioned. The instructions and especially the videos were very helpful and having ability to move forward or backwards between different sections instead of having to rewatch the whole video was great. The labelling The interface was well designed and simple to navigate/use once I understood how the system functioned. The simple/clean UI is amazing, as is the tiered approach. Like Workspace/Projects/Tasks is laid out beautifully and the transition between each tier makes a lot of sense. Minorly, the voice chosen/used for the demo videos was enjoyable and never got grating. That it is easy and fast to use. The system adding similar keywords to my keyword list is a great feature. I liked the step by step approach - it was systematic and logical I like how simple it was use to use and how the ui is fast and was able to select keywords easily and how fast everything was. Really nice UI, relatively small learning curve
Which ideas or advice for future improvement of CD4AI do you have?	Possibly being able to look at the inbuilt dataset that I used at a glance before doing the classifying can make the process feel slightly less abstract A pictorial overview of the complete process to be followed, like a flowchart, and perhaps some idea of where you are in that process. Make navigating between tasks (i.e. once one task is done, click on that task to reveal the next one) more obvious. During the context window extraction its not clear if voting NO means that the Ai will just ignore this article or if it will use it as like a negative attribute, make this more clear or add a NOT SURE button if theres not enough context to tell Maybe number each task so you know what order to complete them in. Provide an conceptual overview of what the software is doing. The only real item I can think of is just changing the "Extracted Context Windows" section, perhaps set the AI a minimum word or character length for the Extracts. You could also add to this a button that lets you see a larger Extract or even the source page for the Extract to allow you to more confidently rate the Context None. If it is a user's first time using the application, the demo videos are helpful, but potentially make the user conduct a guided simple test project before they begin as this would be helpful in establishing how to use the application. I was a little disappointed that the auto assign button was disabled in testing, as that seems like a key feature here and I wanted to take it for a spin. I cannot think of anything as it worked fine for me and was easy to comprehend. Maybe change the name - Seedy For AI might not give the best impression of the product. would be interesting to see which articles fall into multiple datasets and be presented with a method of manually adjusting those as examples for the system to recategorise I would like a feature to (automatically?) split the dataset into test train, and then it should show the accuracy of the result data. Another feature, when selecting context windows, it shouldn't really be binary yes or no, perhaps an "unsure" button if you cannot tell if the text fits in that class, or the ability to "show more" or expand the current text, perhaps show the whole data point so I can make a better decision. Make it easier to generate the context keywords, start the artifact, and create labeled dataset

List of Figures

List of Tables

Bibliography

[1] F. D. Davis. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology". In: *MIS Quarterly* 13.3 (1989), p. 319. ISSN: 02767783. DOI: 10.2307/249008.

[2] W. Xu. "Toward human-centered AI". In: *Interactions* 26.4 (2019), pp. 42–46. ISSN: 1072-5520. DOI: 10.1145/3328485.

[3] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza. "Power to the People: The Role of Humans in Interactive Machine Learning". In: *AI Magazine* 35.4 (2014), pp. 105–120. ISSN: 0738-4602. DOI: 10.1609/aimag.v35i4.2513.

[4] J. Brooke. "SUS: A 'Quick and Dirty' Usability Scale". In: *Usability Evaluation In Industry*. Ed. by P. W. Jordan, B. Thomas, I. L. McClelland, and B. Weerdmeester. CRC Press, 1996. ISBN: 9780429157011.

[5] ISO. *Ergonomics of human-system interaction: Part 210: Human-centred design for interactive systems (ISO 9241-210:2019)*. 2019. URL: https://www.iso.org/standard/77520.html (visited on 10/16/2025).

[6] S. Meisenbacher, T. Schopf, W. Yan, P. Holl, and F. Matthes. "An Improved Method for Class-specific Keyword Extraction: A Case Study in the German Business Registry". In: *Proceedings of the 20th Conference on Natural Language Processing (KONVENS 2024)*. Ed. by Luz de Araujo, Pedro Henrique, A. Baumann, D. Gromann, B. Krenn, B. Roth, and M. Wiegand. Vienna, Austria: Association for Computational Linguistics, 2024, pp. 159–165. URL: https://aclanthology.org/2024.konvens-main.18/.

[7] A. Dix, J. Finlay, G. D. Abowd, and R. Beale. *Human-Computer Interaction*. 3rd ed. Harlow: Pearson Prentice-Hall, 2004. ISBN: 9780130461094.

[8] D. A. Norman. *The design of everyday things*. Revised and expanded ed. New York: Basic Books, 2013. ISBN: 9780465050659. URL: http://swb.eblib.com/patron/FullRecord.aspx?p=1167019.

[9] J. Nielsen. *10 Usability Heuristics*. 1994. URL: https://www.nngroup.com/articles/ten-usability-heuristics/ (visited on 10/17/2024).

[10] B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, and N. Elmqvist. *Designing the user interface: Strategies for effective human-computer interaction*. Sixth edition, global edition. Boston et al.: Pearson, 2018. ISBN: 9781292153926. URL: https://elibrary.pearson.de/book/99.150005/9781292153926.

[11] ISO. *Ergonomics of human-system interaction: Part 110: Interaction principles (ISO 9241-110:2020)*. 2020. URL: https://www.iso.org/standard/75258.html (visited on 10/16/2025).

[12] J. Nielsen. "Enhancing the explanatory power of usability heuristics". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Ed. by B. Adelson, S. Dumais, and J. Olson. New York, NY, USA: ACM, 1994, pp. 152–158. ISBN: 0897916506. DOI: 10.1145/191666.191729.

[13] E. Horvitz. "Principles of mixed-initiative user interfaces". In: *CHI 99: The CHI is the limit, human factors in computing systems; CHI 99 Conference proceedings; [Pittsburgh, PA, USA, May 15 - 20 1999*. Ed. by M. G. Williams. New York: ACM Press, 1999, pp. 159–166. ISBN: 0201485591. DOI: 10.1145/302979.303030.

[14] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, J. Teevan, R. Kikin-Gil, and E. Horvitz. "Guidelines for Human-AI Interaction". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Ed. by S. Brewster. ACM Digital Library. New York,NY,United States: Association for Computing Machinery, 2019, pp. 1–13. ISBN: 9781450359702. DOI: 10.1145/3290605.3300233.

[15] H. Pailian and L. Li. "Landscape of User-Centered Design Practices for Fostering Trustworthy Human-AI Interactions". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 66.1 (2022), pp. 1255–1259. ISSN: 1071-1813. DOI: 10.1177/1071181322661387.

[16] G. Margetis, S. Ntoa, M. Antona, and C. Stephanidis. "HUMAN–CENTERED DESIGN OF ARTIFICIAL INTELLIGENCE". In: *HANDBOOK OF HUMAN FACTORS AND ERGONOMICS*. Ed. by G. Salvendy and W. Karwowski. Wiley, 2021, pp. 1085–1106. ISBN: 9781119636083. DOI: 10.1002/9781119636113.ch42.

[17] ISO. *Ergonomics of human-system interaction: Part 11: Usability: Definitions and concepts (ISO 9241-11:2018)*. 2018. URL: https://www.iso.org/standard/63500.html (visited on 10/16/2025).

[18] J. Nielsen and T. K. Landauer. "A mathematical model of the finding of usability problems". In: *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*. Ed. by B. Arnold. ACM Conferences. New York, NY: ACM, 1993, pp. 206–213. ISBN: 0897915755. DOI: 10.1145/169059.169166.

[19] A. Holzinger. "Usability engineering methods for software developers". In: *Communications of the ACM* 48.1 (2005), pp. 71–74. ISSN: 0001-0782. DOI: 10.1145/1039539.1039541.

[20] D. Alonso-Ríos, A. Vázquez-García, E. Mosqueira-Rey, and V. Moret-Bonillo. "Usability: A Critical Analysis and a Taxonomy". In: *International Journal of Human-Computer Interaction* 26.1 (2009), pp. 53–74. ISSN: 1044-7318. DOI: 10.1080/10447310903025552.

[21] D. A. Norman, ed. *User centered system design: New perspectives on human-computer interaction*. 9. [print.] Hillsdale, NJ: Erlbaum, 1986. ISBN: 0898598729.

[22] M. Maguire. "Context of Use within usability activities". In: *International Journal of Human-Computer Studies* 55.4 (2001), pp. 453–483. ISSN: 10715819. DOI: 10.1006/ijhc.2001.0486.

[23] J. Nielsen. *Usability engineering*. Cambridge, Mass.: AP Professional, 1993. ISBN: 9780080520292. URL: https://learning.oreilly.com/library/view/-/9780125184069/?ar.

[24] J. Nielsen. *Why You Only Need to Test with 5 Users*. 2000. URL: https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/ (visited on 10/14/2025).

[25] J. Nielsen. *Quantitative Studies: How Many Users to Test?* 2006. URL: https://www.nngroup.com/articles/quantitative-studies-how-many-users/ (visited on 10/14/2025).

[26] A. Bangor, P. Kortum, and J. Miller. "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale". In: *Journal of Usability Studies* (2009). URL: https://dl.acm.org/doi/10.5555/2835587.2835589.

[27] A. Bangor, P. T. Kortum, and J. T. Miller. "An Empirical Evaluation of the System Usability Scale". In: *International Journal of Human-Computer Interaction* 24.6 (2008), pp. 574–594. ISSN: 1044-7318. DOI: 10.1080/10447310802205776.

[28] J. Sauro. *Measuring Usability with the System Usability Scale (SUS)*. 2011. URL: https://measuringu.com/sus/ (visited on 10/11/2025).

[29] M. Schreier, R. Brandt, H. Brown, T. Saensuksopa, C. Silva, and L. M. Vardoulakis. "User-Centered Delivery of AI-Powered Health Care Technologies in Clinical Settings: Mixed Methods Case Study". In: *JMIR human factors* 12 (2025), e76241. DOI: 10.2196/76241.

[30] A. C. Griffin, S. Khairat, S. C. Bailey, and A. E. Chung. "A chatbot for hypertension self-management support: user-centered design, development, and usability testing". In: *JAMIA open* 6.3 (2023), ooad073. DOI: 10.1093/jamiaopen/ooad073.

[31] A. Smith, V. Kumar, J. Boyd-Graber, K. Seppi, and L. Findlater. "Closing the Loop: User-Centered Design and Evaluation of a Human-in-the-Loop Topic Modeling System". In: *23rd International Conference on Intelligent User Interfaces*. Ed. by S. Berkovsky. ACM Conferences. New York, NY: ACM, 2018, pp. 293–304. ISBN: 9781450349451. DOI: 10.1145/3172944.3172965.

[32] Z. Fang, L. Alqazlan, Du Liu, Y. He, and R. Procter. "A User-Centered, Interactive, Human-in-the-Loop Topic Modelling System". In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Ed. by A. Vlachos and I. Augenstein. Dubrovnik, Croatia: Association for Computational Linguistics, 2023. URL: https://arxiv.org/pdf/2304.01774.

[33] C. Jimenez, P. Lozada, and P. Rosas. "Usability heuristics: A systematic review". In: *2016 IEEE 11th Colombian Computing Conference (CCC): Conference proceedings : 27-30 September, 2016, Popayán, Colombia*. Ed. by I. D. Jácome V. Piscataway, NJ: IEEE, 2016, pp. 1–8. ISBN: 978-1-5090-2966-2. DOI: 10.1109/ColumbianCC.2016.7750805.

[34] X. He, H. Zhang, and J. Bian. "User-centered design of a web-based crowdsourcing-integrated semantic text annotation tool for building a mental health knowledge base". In: *Journal of biomedical informatics* 110 (2020), p. 103571. DOI: 10.1016/j.jbi.2020.103571.

[35] S. Marien, D. Legrand, R. Ramdoyal, J. Nsenga, G. Ospina, V. Ramon, and A. Spinewine. "A User-Centered design and usability testing of a web-based medication reconciliation application integrated in an eHealth network". In: *International journal of medical informatics* 126 (2019), pp. 138–146. ISSN: 1872-8243. DOI: 10.1016/j.ijmedinf.2019.03.013. URL: https://pubmed.ncbi.nlm.nih.gov/31029255/.

[36] A. C. S. Da Sobrinho, G. A. d. O. Gomes, and C. R. Bueno Júnior. "Developing a Multiprofessional Mobile App to Enhance Health Habits in Older Adults: User-Centered Approach". In: *JMIR formative research* 8 (2024), e54214. DOI: 10.2196/54214.

[37] P. P. Adinda and A. Suzianti. "Redesign of user interface for e-government application using usability testing method". In: *Proceedings of the 4th International Conference on Communication and Information Processing*. Ed. by J. Ben-Othman. ACM Other conferences. New York, NY: ACM, 2018, pp. 145–149. ISBN: 9781450365345. DOI: 10.1145/3290420.3290433.

[38] M. Zeiler, N. Dietzel, F. Haug, J. Haug, K. Kammerer, R. Pryss, P. Heuschmann, E. Graessel, P. L. Kolominsky-Rabas, and H.-U. Prokosch. "A User-Centered Design Approach for a Screening App for People With Cognitive Impairment (digiDEM-SCREEN): Development and Usability Study". In: *JMIR human factors* 12 (2025), e65022. DOI: 10.2196/65022.

[39] L. Happe, M. Sgraja, A. Hein, and R. Diekmann. "Iterative Development and Applicability of a Tablet-Based e-Coach for Older Adults in Rehabilitation Units to Improve Nutrition and Physical Activity: Usability Study". In: *JMIR human factors* 9.1 (2022), e31823. DOI: 10.2196/31823.

[40] Prolific, ed. *Participants skilled at AI tasks*. 2025. URL: https://researcher-help.prolific.com/en/article/d1c536?_gl=1*8ru8ki*_gcl_au*MTgwMDU3NDU0OS4xNzU4MTgzOTYz (visited on 10/08/2025).

[41] S. Duranton, J. Erlebach, C. Brégé, J. Danziger, A. Gallego, and M. Pauly. *What's Keeping Women Out of Data Science?* 2020. URL: https://www.bcg.com/publications/2020/what-keeps-women-out-data-science (visited on 10/13/2025).

[42] J. M. Leimeister. *Collaboration Engineering: IT-gestützte Zusammenarbeitsprozesse systematisch entwickeln und durchführen*. Berlin and Heidelberg: Springer Gabler, 2014. ISBN: 978-3-642-20890-4. DOI: 10.1007/978-3-642-20891-1.

[43] S. G. Hart and L. E. Staveland. "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research". In: *Human mental workload*. Ed. by P. A. Hancock and N. Meshkati. Advances in psychology. Amsterdam: North-Holland, 1988, pp. 139–183. ISBN: 9780444703880. DOI: 10.1016/s0166-4115(08)62386-9.

[44] B. Laugwitz, T. Held, and M. Schrepp. "Construction and Evaluation of a User Experience Questionnaire". In: *HCI and usability for education and work: 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2008, Graz, Austria, November 20-21, 2008 ; proceedings*. Ed. by A. Holzinger. Vol. 5298. Lecture notes in computer science. Berlin and Heidelberg: Springer, 2008, pp. 63–76. ISBN: 978-3-540-89349-3. DOI: 10.1007/978-3-540-89350-9{\textunderscore}6.

[45] StatCounter, ed. *Percentage of mobile device website traffic worldwide from 1st quarter 2015 to 2nd quarter 2025*. 2025. URL: https://www.statista.com/statistics/277125/share-of-website-traffic-coming-from-mobile-devices/ (visited on 10/13/2025).