Electronic Communications of the EASST
Volume 85 Year 2025

deRSE25 - Selected Contributions of the 5th Conference for Research Software Engineering in Germany

Edited by: René Caspart, Florian Goth, Oliver Karras, Jan Linxweiler, Florian Thiery, Joachim Wuttke

Linguistic corpus research software at the Leibniz-Institute for the German Language (IDS)

Nils Diewald, Franck Bodmer, Peter M. Fischer, Elena Frick, Marc Kupietz, Mark-Christoph Müller, Helge Stallkamp, Uyen-Nhu Tran

DOI: 10.14279/eceasst.v85.2692

Electronic Communications of the EASST (https://eceasst.org).
Published by **Berlin Universities Publishing**
(https://www.berlin-universities-publishing.de/)

Linguistic Corpus Research Software at the Leibniz-Institute for the German Language (IDS)

Nils Diewald[1], Franck Bodmer[2], Peter M. Fischer[3], Elena Frick[4], Marc Kupietz[5], Mark-Christoph Müller[6], Helge Stallkamp[7], Uyen-Nhu Tran[8]

[1] diewald@ids-mannheim.de
[2] bodmer@ids-mannheim.de
[3] peter.fischer@ids-mannheim.de
[4] frick@ids-mannheim.de
[5] kupietz@ids-mannheim.de
[6] mark-christoph.mueller@ids-mannheim.de
[7] stallkamp@ids-mannheim.de
[8] tran@ids-mannheim.de
Leibniz-Institut für Deutsche Sprache
Mannheim, Germany

Abstract: Empirical linguistic research requires access to richly annotated and metadata-enhanced language corpora. This paper presents the ongoing development of corpus search and analysis platforms at the Leibniz-Institute for the German Language (IDS), which provide access to DeReKo, the world's largest collection of contemporary written German corpora, and the Archive for Spoken German (AGD) among others. We describe our platforms, especially focusing on improving, extending and evaluating their user interfaces. Challenges addressed include legal constraints, handling large and heterogeneous datasets, ensuring reproducibility, maintenance of legacy systems, and especially meeting accessibility and usability standards for a diverse scientific audience from the humanities.
This work contributes to the broader effort of advancing research infrastructure in linguistics and offers insights into sustainable and user-friendly corpus technology design.

Keywords: Corpus Linguistics, Language Resources, User Interface Design, Legacy Software

1 Introduction

Linguistic research on language phenomena requires access to language data, which may consist of collections of written texts, transcriptions of spoken language or audio/video recordings. To be suitable for empirical analysis, such data collections, called *language corpora*, must be accompanied by comprehensive metadata, such as information about the author, genre, or source of the text, or in the case of spoken language corpora, details about the communication situation and its participants. In addition, corpora should be sufficiently large and represent a broad range of linguistic strata to allow for reliable generalizations about language structure and usage. It is equally important that these collections are prepared in a way that supports efficient linguistic analysis. For example, recordings of spoken language need to be transcribed (converted into the written form) while both texts and transcriptions can undergo further processing like segmentation (e.g. in tokens, sentences or speaker contributions) and annotation with linguistic information (e.g. part-of-speech tags). Finally, specialized software is required to efficiently search and analyze large datasets and to visualize results in various statistical and graphical formats, enabling the systematic exploration of linguistic patterns.

This paper focuses on the development of search and analysis software for linguistic corpora at the Leibniz Institute for the German Language (IDS) in Mannheim, Germany. The IDS is the central scientific institution for the study and documentation of the contemporary usage and recent history of the German language. It provides and continuously updates two major collections of German-language corpora, which serve as essential empirical resources for linguistic research on the German language worldwide:

The German Reference Corpus (DeReKo) [KL14] — the world's largest collection of corpora of contemporary written German (approx. 61 billion tokens; release 2025-I), comprising a wide range of text media sources, text types, and genres. DeReKo follows a so-called "primordial sample" design, which means that the aim is not to create a corpus that is balanced or representative in any way, but rather to cover as many strata as possible and leave the creation of subcorpora (in the form of so-called *virtual corpora* [KBKW10]) for research questions to the user.

The Archive for Spoken German (AGD) [SS14] — the most extensive archive of audio and video corpora documenting spoken German in interaction and regional variation (approx. 5,000 hours). The most important corpus in the collection is FOLK, the Research and Teaching Corpus of Spoken German [RDS23], which, with around 390 hours and 3.7 million transcribed tokens (version 2.24), is the largest corpus of authentic spontaneous conversations from various private, institutional and public communication settings.

To make these datasets accessible to researchers worldwide, IDS develops and maintains corpus search and analysis platforms and tools that enable users to explore the data online. The following platforms, currently in active use, can be cited as examples:

COSMAS II[1] [Bod05] is a web application for querying IDS text corpora, including DeReKo, historical corpora, and other collections. The development started in the mid-1990s as the successor of COSMAS I [al-94]. It supports searches in multiple layers of annotations using its own corpus query language developed specifically for linguistic research. Key features of COSMAS II include virtual corpus building, display of corpus contents and search results based on metadata, intra-textual properties and frequency distribution, support for lemma based word form generation, user management of word lists, conducting collocation analysis, and the export of results and related metadata.

[1] https://www2.ids-mannheim.de/cosmas2

[2] https://korap.ids-mannheim.de

KorAP[2] [BFF+12] is an open-source[3], web-based corpus analysis platform that has been developed since 2011 as a successor to COSMAS II for the primary access to DeReKo, but its data-independent architecture also enables its use with other corpora at external institutes [KBD+24]. Thanks to its modular architecture, KorAP can be extended with additional components tailored to specific corpus analysis tasks, with Kalamar [DS24] (see Subsection 3.1 and Subsection 3.2) being its primary user interface. It supports the querying of large, multiply annotated corpora (across layers, sources, and structurally overlapping) and offers a variety of visualization options, including graphical views of dependency and syntactic annotations. KorAP is based on a fulltext search engine and supports multiple query languages commonly used in corpus linguistics.

DGD[4] [Sch14] is a web-based corpus platform for browsing and querying access to some of the transcribed corpora of spoken German from the AGD. The platform has been available online since early 2012 and is regularly updated (approx. two releases per year). It is built on a relational database and supports token-based concordance searches within multi-layer-annotated transcriptions of spoken language. In addition to various filtering options and output visualizations for search results, DGD also provides access to audio and video data that are synchronized with the corresponding transcripts.

ZuRecht[5] [FHW23] is a prototype implementation of a web-based graphical user interface for CQP-based queries on transcriptions of spoken language. The underlying search engine is implemented using the open source ZuMult[6] architecture [SFF+23] that applies MTAS[7] [BBK17], a Lucene-based[8] linguistic search engine. Currently, ZuRecht serves both as an extension of the DGD platform and as an experimental environment for developing new features tailored to the analysis of spoken language data, such as searches for time-aligned annotations, speaker turns, and repetitions.

grammis[9] [SL22] is a hypermedia-driven online platform offering comprehensive resources on German grammar. Initially launched in the mid-1990s and optimized for mobile use since 2018, it transcends static digitization by integrating multimedia (e.g., audiovisual examples of intonation, animated grammar explanations) and interactive tools for corpus-based analysis. The platform integrates empirical corpus linguistics with systematic grammatical descriptions by grounding its grammatical analyses in observable language data derived from large, annotated corpora of contemporary German. In addition, grammis includes dictionaries, terminological databases, and curated data collections, which serve both as analytical tools and as the empirical basis for scientific publications. With over 100,000 monthly page views, grammis ranks among the most widely used online resources for linguistic research on the German language.

[3] KorAP consists of multiple components, most of which are published under the BSD-2-Clause license, accessible via https://github.com/KorAP.

[4] https://dgd.ids-mannheim.de

[5] https://zumult.ids-mannheim.de

[6] https://github.com/zumult-org

[7] https://github.com/textexploration/mtas

[8] https://lucene.apache.org/

[9] https://grammis.ids-mannheim.de/

In this paper, we draw on several years of hands-on experience in developing and maintaining these corpus analysis platforms to present selected design strategies and implementation approaches that have proven effective in this context. We begin by outlining challenges associated with the nature of language corpora in Section 2. In Section 3 we will focus in particular on the developments that are of great importance to the end user. These are improvements regarding the user experience (in Subsection 3.1) and enhancements of functionalities in the user interface (in Subsection 3.2), which are made especially for the corpus systems for written language and are presented as examples. These improvements and enhancements were developed both against the background of user feedback and in order to achieve functional parity with the predecessor system. With regard to the systems for spoken language, an extensive evaluation was carried out, which we present in Subsection 3.3 — and which we intend to adapt for the systems for written language in the future in order to be able to coordinate further developments on this basis.

Developments that are not immediately noticeable to users are described in Section 4. In addition to the maintenance of existing functionalities, this primarily involves ensuring the availability of functionalities that are no longer maintained or that use dependencies of functions that are no longer maintained (maintenance of legacy software). Here, too, we present examples of how to deal with such software. Section 5 concludes the paper with a summary.

2 Conceptual Challenges and Related Work

Corpora, used in linguistic research, consist of language data essentially structured around three main components: a) the primary data, b) the metadata and c) annotations that add further information to the primary data.

In the case of non-native digital input, primary data is based on raw data. In the case of oral corpora, primary data is usually manually created transcripts that convert audio data into a textual form. In the case of written corpora, the primary data is often based on image data (see Figure 1; Step “Processing”). It is important to understand that the primary data of a corpus can have an interpretative distance to the original raw data, which is often resolved by adding the underlying media to the corpus in a non-interpreted form. In the case of oral language primary data, this means that ideally a direct association between individual transcript parts and their equivalents in audio and video files is kept (cf. [LZ10], p. 44), as is the case for DGD and ZuRecht.

Metadata describes the linguistic material and also has different properties for written and spoken corpora. For example, written texts usually contain a title, a publication date and author information, which is encoded for each individual document. In oral corpora, on the other hand, due to their more individual nature, a richer set of metadata categories is usually employed. It includes, for example, time and place of the underlying (audio or video) recording, the general communication situation (e.g. private conversation, TV show, teaching situation, etc.), and most notably speaker information. This, in turn, includes information like age and gender, but can also extend to biographical data like place of birth and language socialisation, language skills, or education (cf. [DR23] for the FOLK corpus).

Annotations describe parts of the primary data. In order to assign these to the content, the data is usually segmented. This normally takes the form of tokenization (i.e. splitting into graphemic words), but in the case of oral data, segmentation is also concerned with assigning

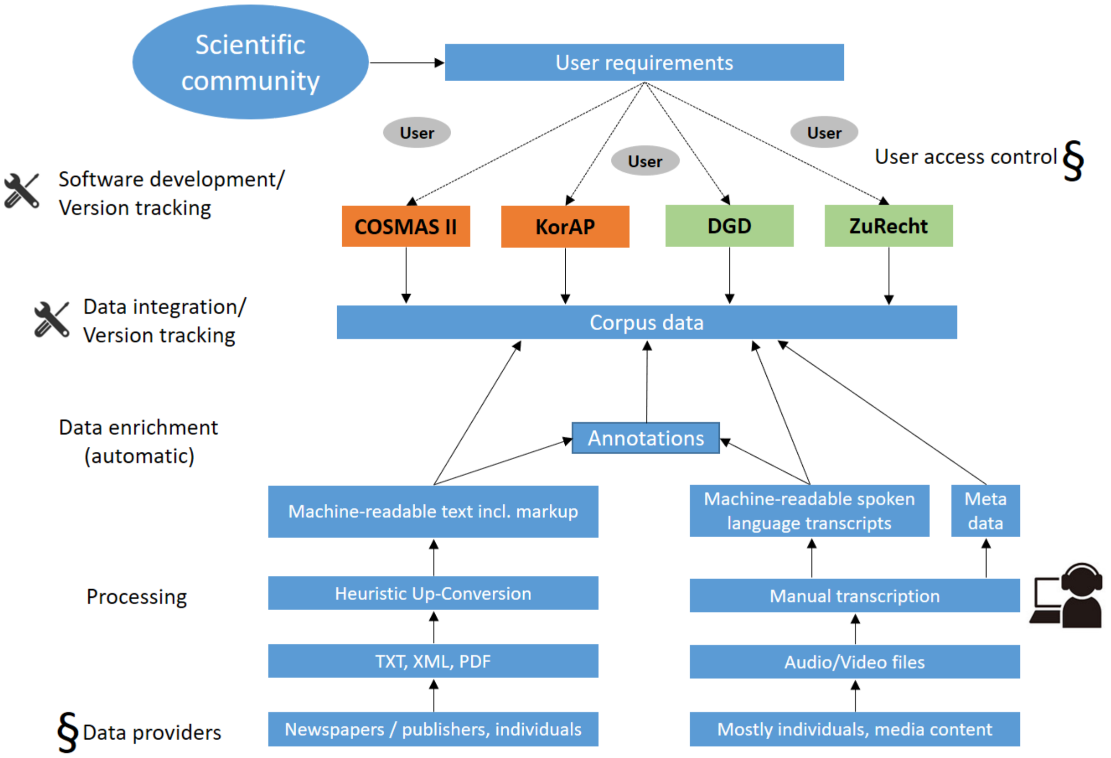

Scientific community
User requirements
User
User
User
User access control §
Software development/ Version tracking
COSMAS II
KorAP
DGD
ZuRecht
Data integration/ Version tracking
Corpus data
Data enrichment (automatic)
Annotations
Machine-readable text incl. markup
Machine-readable spoken language transcripts
Meta data
Processing
Heuristic Up-Conversion
Manual transcription
TXT, XML, PDF
Audio/Video files
§ Data providers
Newspapers / publishers, individuals
Mostly individuals, media content

Figure 1: Data preparation and search in corpora of written language (left) and corpora of spoken language (right). The diagram highlights, on an abstract level, those processing steps that are *different* for corpora of written and spoken language (lower part), and those that are *identical* (upper part). On a technical level, however, the systems for written (COSMAS II and KorAP) and spoken data (DGD and ZuRecht) are kept separate, mostly because of different user access limitations and different data storage and querying requirements of written vs. spoken language corpora.

sections of speech to different speakers, and with structuring the data on the basis of pauses and speaker changes in order to ease reading and navigation. Annotations can reflect structural information that is already present in the raw data but has been lost during the transfer to primary data. In the case of written corpora, for example, this could be chapter sections, headings or footnote markings. Linguistic annotations are added to these structural annotations and can contain morphological, syntactical, lexical, phonological and other information (see Figure 1; Step "Data enrichment").

Linguists refer to such curated and enriched corpus data for various purposes. For example, to search for examples for dictionary work, but also for quantitative studies of linguistic patterns in both written and spoken data. Depending on the perspective, studies can also be conducted on different dimensions of the data, for example for the investigation of diachronic language states, regional varieties, or different text and conversational genres. To address these different needs, corpus analysis systems (see Figure 1; Step "Software development") must offer complex search functionalities and filters for data exploration and produce result views that provide immediate access to the underlying corpus data, including its primary data, metadata and annotations. In addition, corpus analysis systems must provide a high degree of reproducibility across different development stages [DMK21].

Linguistics is in the special situation that its research data are typically affected by third parties' rights. These rights include intellectual property rights, mainly but not exclusively affected in the case of written corpora, and general personal rights such as privacy and data protection, mainly but not exclusively affected in the case of spoken corpora. In addition, the affected rights holders do not belong to the research community so that open data models are not as easily applicable as in other disciplines. For these reasons, licenses usually need to be obtained to use texts for linguistic research purposes, and informed consent needs to be obtained for spoken language recordings. Alternatively, for spoken data, if no such wide-ranging consent can be obtained from the affected rights holders (usually the recorded persons), measures have to be taken to technically control and limit user access to the data. Regarding texts, the Text and Data Mining (TDM) limitation, first introduced in 2017 in EU copyright laws, has improved the situation by allowing for non-commercial use of texts. However, under this limitation, corpora can only be shared with a specifically defined group of people and not with the entire research community [Kam18].

These conceptual challenges lead to technical challenges that the applications presented here address and can also be viewed in the context of related work. Especially at the beginning of the development of each platform, there were no systems available that fulfilled the specific requirements[10] targeted by the respective project. Even today, only a few systems fully meet the set of requirements that led to the development of KorAP and ZuRecht.

Since the corpora presented cannot be downloaded for legal reasons, web access is essential. However, some corpus systems for analyzing written language, such as AntConc [Ant22] or CorpusExplorer [Rüd20], and for spoken corpora, such as ELAN [Wil19] and EXAKT/EXMARaLDA [SW09], are primarily intended for use with project corpora on desktop systems, which is why they are not suitable for this application. The legal restrictions also mean that

[10] In addition to these conceptual challenges, the required functionality of KorAP was specified at the start of the project by conducting a survey followed by an evaluation of existing systems [BFF+12]. Likewise, systems were evaluated in advance for ZuRecht/ZuMult [BFS21] and a user study for the DGD was conducted [FFH+16].

the corpus search and analysis platforms used to provide DeReKo and AGD must have a user management and a finely differentiated rights management, in order to guarantee that the rights of authors and speakers are not infringed upon on the one hand, and on the other hand, to restrict the possibilities of usage as little as possible [MDKK24].

Furthermore, given the sheer volume of data, especially in written language, the system must be able to deal with that. Established systems, however, often focus on smaller sized corpora. The IMS Open Corpus Workbench, for example, has a technical size limitation of 2.1 billion tokens [EH15]. Similarly, the widely used ANNIS [ZJLC09] is primarily aimed at smaller, albeit highly annotated corpora, which does not meet the requirements for user created virtual corpora. Multiple annotations, on the other hand, are only supported to a limited extent by other systems. Many older systems only support individual annotation layers (e.g., only one POS annotation) such as MonoConc[11] — but not multiple, especially conflicting, annotation levels. Only recently have systems emerged with requirements similar to those formulated for KorAP, such as BlackLab[12] [DND17], which has a similar feature profile and approach as KorAP.

Comparable requirements apply to spoken language, which also necessitates special handling of overlapping speech, close linking of queries with metadata, and integration of audio and video material in the presentation of results. At the time of the development of the DGD, comparable platforms already existed online (e.g., Talkbank [Mac07]). However, these platforms were based on architectures that did not support deployment in other data repositories or for other data formats (cf. [MS09] for an overview of corpus analysis platforms for spoken language at the time). Today, many open-source platforms offer flexible, data-independent architectures, but only a few of them support the search for features specific to spoken language — such as pauses, speaker overlaps, time-aligned annotations and non-verbal events like laughter or frowning — as is possible with ZuRecht (cf. [FS25] for a current overview of search platforms for spoken language).

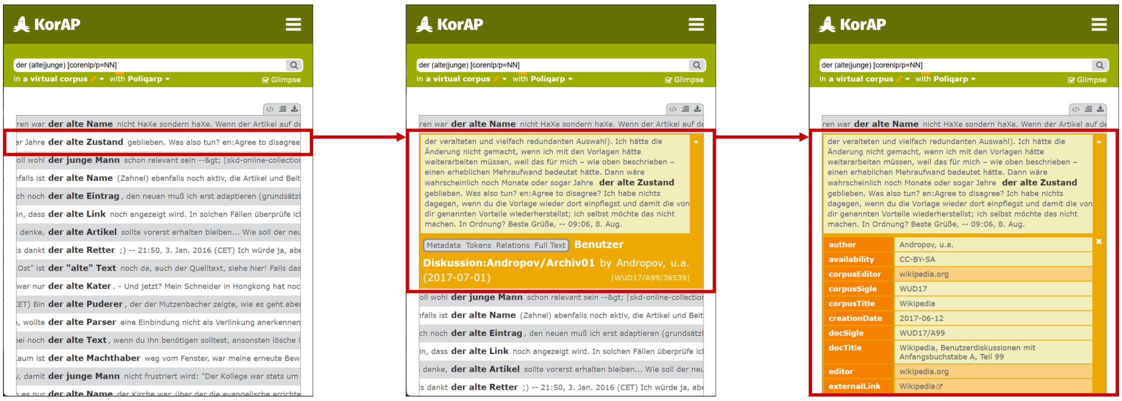
3 Developing User Interfaces

The majority of users of our corpus analysis platforms come from the humanities (access to the corpora mentioned in this paper is granted upon registration and is restricted to scientific, non-commercial purposes only), which is why in-depth technical understanding cannot necessarily be assumed. In addition, the different underlying research interests result in a very broad and diverse user base that requires correspondingly different functionalities (cf. [SZR08]). A multi-level access system was therefore established for the written corpora [KDM22], which enables users with technical skills to access the database very effectively and beyond the functionalities of the graphical front end. For the oral corpora, two separate software accesses to the data were developed, DGD and ZuRecht, in order to satisfy the different user groups. DGD is aimed at a general user audience with limited query options, while ZuRecht, with its support for complex CQP queries, is intended to appeal to expert users willing to use a more powerful, but more complicated query console.

Especially for less tech-savvy users (but with a strong scientific research background), the

[11] https://monoconc.com/

[12] http://inl.github.io/BlackLab/



der (alte|junge) [corenlp/p=NN]

ren war **der alte Name** nicht HaXe sondern haXe. Wenn der Artikel auf de
ar Jahre **der alte Zustand** geblieben. Was also tun? en:Agree to disagree
oll wohl **der junge Mann** schon relevant sein --&gt; [skd-online-collectio
nfalls ist **der alte Name** (Zahnel) ebenfalls noch aktiv, die Artikel und Beit
ch noch **der alte Eintrag** , den neuen muß ich erst adaptieren (grundsätzl
in, dass **der alte Link** noch angezeigt wird. In solchen Fällen überprüfe ic
denke, **der alte Artikel** sollte vorerst erhalten bleiben... Wie soll der neu
s dankt **der alte Retter** ;) -- 21:50, 3. Jan. 2016 (CET) Ich würde ja, abe
Ost" ist **der "alte" Text** noch da, auch der Quelltext, siehe hier! Falls das
war nur **der alte Kater** . - Und jetzt? Mein Schneider in Hongkong hat noc
ET) Bin **der alte Puderer** , der der Mutzenbacher zeigte, wie es geht abe
, wollte **der alte Parser** eine Einbindung nicht als Verlinkung anerkennen
ei noch **der alte Text** , wenn du ihn benötigen solltest, ansonsten lösche i
aum ist **der alte Machthaber** weg vom Fenster, war meine erneute Bew
, damit **der junge Mann** nicht frustriert wird: "Der Kollege war stets um
n es nur **der alte Name** der Kirche war, über der die evangelische errichte

der veralteten und vielfach redundanten Auswahl). Ich hätte die Änderung nicht gemacht, wenn ich mit den Vorlagen hätte weiterarbeiten müssen, weil das für mich – wie oben beschrieben – einen erheblichen Mehraufwand bedeutet hätte. Dann wäre wahrscheinlich noch Monate oder sogar Jahre **der alte Zustand** geblieben. Was also tun? en:Agree to disagree? Ich habe nichts dagegen, wenn du die Vorlage wieder dort einpflegst und damit die von dir genannten Vorteile wiederherstellst; ich selbst möchte das nicht machen. In Ordnung? Beste Grüße, -- 09:06, 8. Aug.

Benutzer Diskussion:Andropov/Archiv01 by Andropov, u.a. (2017-07-01) [WUD17/A99/36539]

author	Andropov, u.a.
availability	CC-BY-SA
corpusEditor	wikipedia.org
corpusSigle	WUD17
corpusTitle	Wikipedia
creationDate	2017-06-12
docSigle	WUD17/A99
docTitle	Wikipedia, Benutzerdiskussionen mit Anfangsbuchstabe A, Teil 99
editor	wikipedia.org
externalLink	Wikipedia

Figure 2: Interaction path adding "extras on demand"

usability of the interfaces is of great importance and is — in our experience — an aspect that is often neglected in the development of research software. In the following, we would like to present examples of how we are continuously improving, expanding and evaluating our platforms in order to meet the evolving demands of users and data.

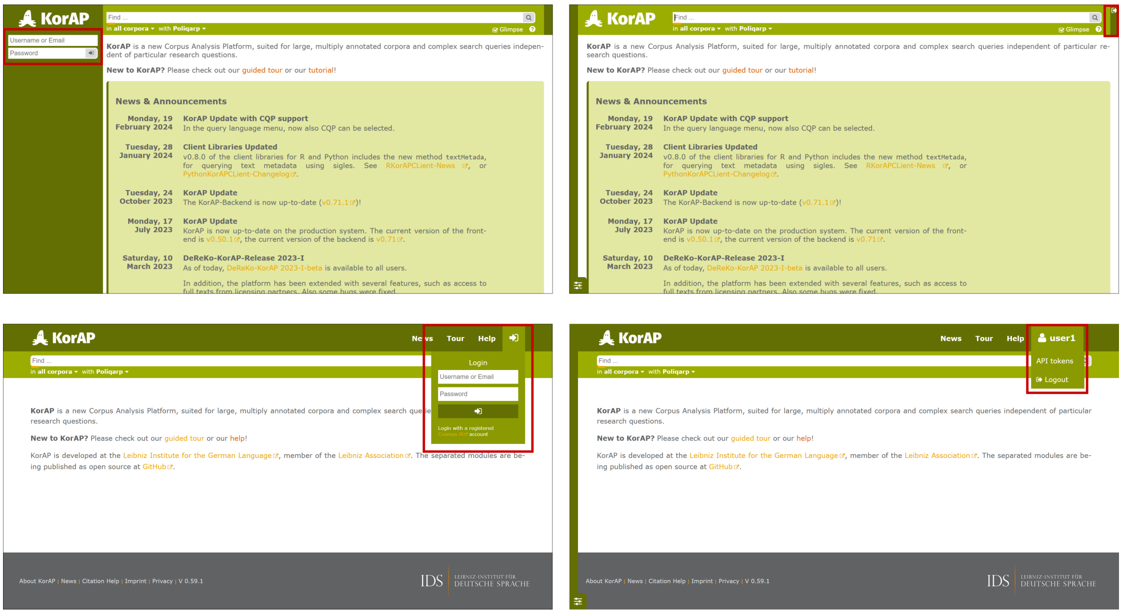
3.1 Improving

Kalamar [DBK19] is the user front-end for the KorAP corpus search and analysis engine. It was developed as a separate component [DHM+16] that communicates with the backend using the publicly accessible web interfaces of KorAP [KDM22]. In this respect, Kalamar acts as an API client.

Due to the wide range of interested researchers who use the web frontend, each of whom only needs part of KorAP's functionality but should not be distracted by too many functions, the design of the user interface follows the principle of "extras on demand" (see [Tid06], p. 45f). This means that basic functionalities are readily accessible, while advanced or less frequently needed options are only displayed on request (see Figure 2). More generally, this approach aligns with the broader principle of progressive disclosure, which recommends presenting information and branching options only when they become relevant to the user's current task. According to established models of human-computer interaction, users typically pursue their goals in user interfaces by following sequential action paths that are shaped by their current task context [DFAR04]. This approach reduces cognitive load and helps to meet different user requirements [Nie06, SPC+17].

As part of improving the user interface, the navigation structure was recently revised to enable a more contemporary and intuitive user experience. The implementation was based on established design principles of visual perception — in particular the principles of proximity, similarity, and continuity — which are considered a central component of modern human-computer interaction [LHB10].

One key measure was the introduction of an additional navigation bar at the top of the page. This now bundles basic functions such as logging in and out, a documentation link and further pages. Previously, for example, login and logout elements were located in different parts of



KorAP is a new Corpus Analysis Platform, suited for large, multiply annotated corpora and complex search queries independent of particular research questions.

New to KorAP? Please check out our guided tour or our tutorial!

News & Announcements

Monday, 19 February 2024	**KorAP Update with CQP support** In the query language menu, now also CQP can be selected.
Tuesday, 28 January 2024	**Client Libraries Updated** v0.8.0 of the client libraries for R and Python includes the new method `textMetada`, for querying text metadata using sigles. See RKorAPClient-News, or PythonKorAPClient-Changelog.
Tuesday, 24 October 2023	**KorAP Update** The KorAP-Backend is now up-to-date (v0.71.1)!
Monday, 17 July 2023	**KorAP Update** KorAP is now up-to-date on the production system. The current version of the frontend is v0.50.1, the current version of the backend is v0.71.
Saturday, 10 March 2023	**DeReKo-KorAP-Release 2023-I** As of today, DeReKo-KorAP 2023-I-beta is available to all users. In addition, the platform has been extended with several features, such as access to full texts from licensing partners. Also some bugs were fixed.

Login with a registered Cosmas-II account

KorAP is a new Corpus Analysis Platform, suited for large, multiply annotated corpora and complex search queries independent of particular research questions.

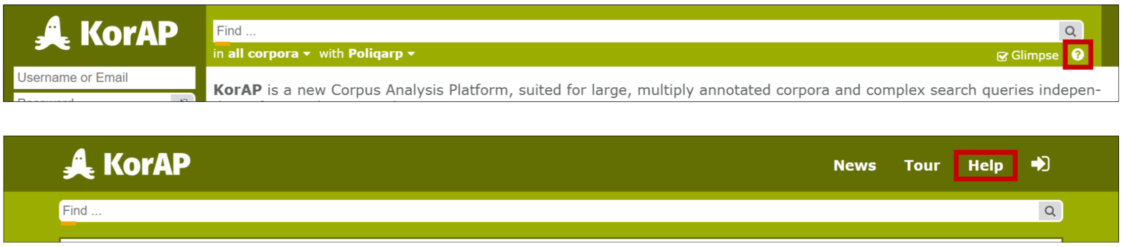
New to KorAP? Please check out our guided tour or our help!

KorAP is developed at the Leibniz Institute for the German Language, member of the Leibniz Association. The separated modules are being published as open source at GitHub.

About KorAP | News | Citation Help | Imprint | Privacy | V 0.59.1

IDS LEIBNIZ-INSTITUT FÜR DEUTSCHE SPRACHE

Figure 3: Comparison of login and logout components: old version (top) and new version (bottom)



KorAP is a new Corpus Analysis Platform, suited for large, multiply annotated corpora and complex search queries indepen-

Figure 4: Help button: old version (top) and new version (bottom)

the application — contrary to the design principle of proximity, which recommends a spatial grouping of functionally related elements (see Figure 3).

The visual appearance of the navigation components was also adapted to the findings of design theory: Related content was made recognizable as belonging together through consistent coloring and clearly defined spacing. This makes it easier to quickly grasp the user interface and improves user orientation. For example, the previously isolated help button, which was located below the search bar in the form of a question mark, has been integrated into the top navigation bar and thus positioned more prominently (see Figure 4). The logo, which serves as a link to navigate to the home page, has also been moved from the search bar to the top navigation bar — a placement that meets current user expectations.

Subtleties such as the display of a profile icon next to the user name also enable the recognition of the login status at a glance (see Figure 5). These design elements follow proven principles of user guidance and reduce the cognitive load of recurring use.

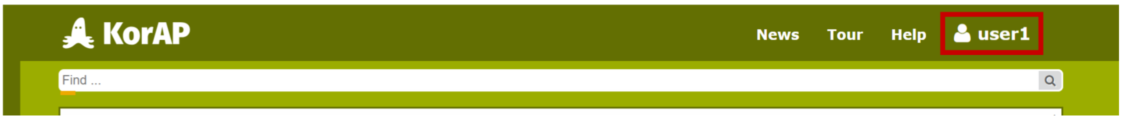
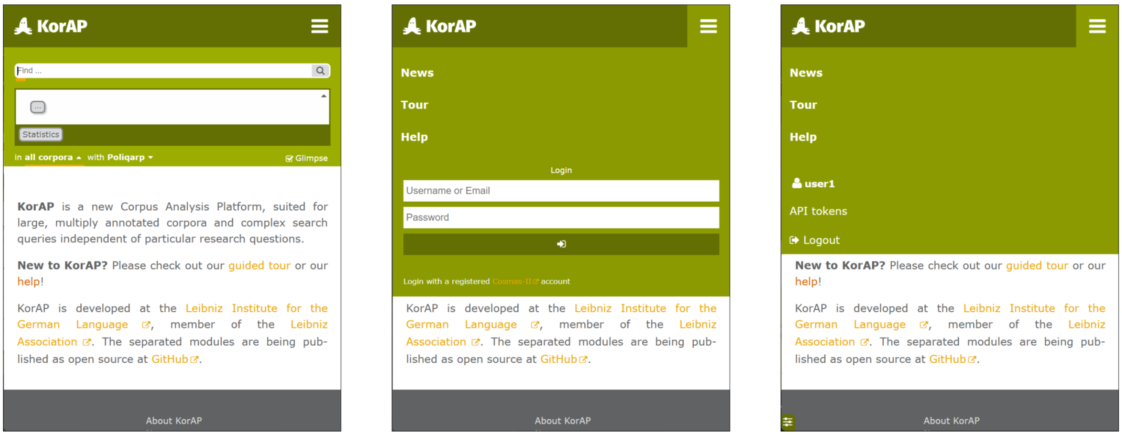


Figure 5: Indication of login status via profile icon and user name



KorAP is a new Corpus Analysis Platform, suited for large, multiply annotated corpora and complex search queries independent of particular research questions.

New to KorAP? Please check out our guided tour or our help!

KorAP is developed at the Leibniz Institute for the German Language, member of the Leibniz Association. The separated modules are being published as open source at GitHub.

Login with a registered Cosmas-II account

KorAP is developed at the Leibniz Institute for the German Language, member of the Leibniz Association. The separated modules are being published as open source at GitHub.

New to KorAP? Please check out our guided tour or our help!

KorAP is developed at the Leibniz Institute for the German Language, member of the Leibniz Association. The separated modules are being published as open source at GitHub.

Figure 6: Mobile view: Burger menu closed (left), opened with user logged out (center), and opened with user logged in (right)

To improve accessibility, attention was also paid to a high-contrast display of fonts and icons as well as sufficient font sizes. These adaptations meet accessibility requirements and make the application easier to use for all user groups.

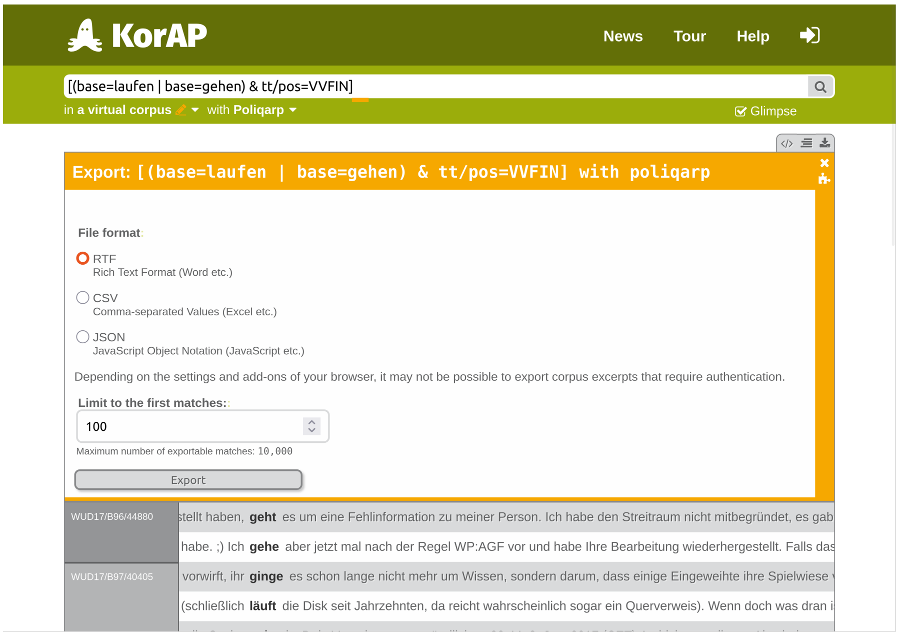
Last but not least, the user interface has been adapted to the new navigation in terms of responsive design to ensure that all functions can still be used comfortably on smaller screens (see Figure 6).

Overall, the revised navigation contributes to the effectiveness, efficiency, and user-friendliness of the application and creates a more consistent design basis for future enhancements.

3.2 Extending

Recent developments in linguistics (especially in corpus linguistics), but also the emergence of new forms of corpus data (e.g., from the field of computer-mediated communication), make it necessary for developers of corpus analysis platforms to constantly provide users with new methods for data exploration (such as new visualization options or the integration of external resources).

Kalamar allows the integration of additional functions in the form of web-based plugins. Plugins are web services that are developed and operated independently from the Kalamar User Interface and can easily be integrated via widgets. Communication between widgets and the user interface, for example with regard to the user's current query, takes place via standardized client-side browser interfaces (i.e. *service workers*). Communication between widgets and the database, for example to retrieve additional data, is done via the KorAP web API.



[(base=laufen | base=gehen) & tt/pos=VVFIN]

in **a virtual corpus** with **Poliqarp**

☑ Glimpse

Export: [(base=laufen | base=gehen) & tt/pos=VVFIN] with poliqarp

File format:

● RTF
Rich Text Format (Word etc.)

○ CSV
Comma-separated Values (Excel etc.)

○ JSON
JavaScript Object Notation (JavaScript etc.)

Depending on the settings and add-ons of your browser, it may not be possible to export corpus excerpts that require authentication.

Limit to the first matches:

100

Maximum number of exportable matches: 10,000

Export

WUD17/B96/44880 stellt haben, **geht** es um eine Fehlinformation zu meiner Person. Ich habe den Streitraum nicht mitbegründet, es gab

habe. ;) Ich **gehe** aber jetzt mal nach der Regel WP:AGF vor und habe Ihre Bearbeitung wiederhergestellt. Falls das

WUD17/B97/40405 vorwirft, ihr **ginge** es schon lange nicht mehr um Wissen, sondern darum, dass einige Eingeweihte ihre Spielwiese

(schließlich **läuft** die Disk seit Jahrzehnten, da reicht wahrscheinlich sogar ein Querverweis). Wenn doch was dran i

Figure 7: Export plugin

The plugin architecture adheres to KorAP's principle of designing small, independent components [DHM+16]. This ensures maintainability, which, given the diverse scientific backgrounds of the users and the limited developer resources, is one of the most challenging tasks in developing research software in computational linguistics [DMK21]. It does not only provide modularity but also a foundation that allows other projects or institutions to develop their own plugins for Kalamar, which can be maintained and extended by the project members themselves and customized for their users' needs. Plugins have to be registered and can be made accessible for the users through a marketplace,[13] which allows the installation and deinstallation of plugins. The security is granted through a restricted security policy of the Kalamar plugin system and KorAP's support of OAuth 2.0. All plugins need to be registered with an OAuth Client ID in case they require OAuth access.

Right now, a number of supported plugins have been developed at IDS including the export and the GlemmService plugin. These are currently not installed separately by the user but are configured as part of Kalamar and are therefore accessible for all users.

Example 1: Support for Exporting Data

In contrast to the COSMAS II export we decided to keep KorAP's export quite simple. Over the years we received many requests addressing additional configuration requirements. To meet the different needs, COSMAS II offers a wide range of user options for configuration, including result and corpus views, different sorting options, various context settings for the result and detailed possibilities regarding the layout of the exported file. From our user feedback, we re-

[13] The marketplace is not yet public.

alized that the manifold possibilities are quite overwhelming for beginners and also tend to be confusing for experienced users. Additionally, maintaining and upgrading to different file format versions is significantly resource-intensive. Therefore we offered a reduced export selecting JSON and CSV file formats in addition to RTF, which allow automatic post-processing of the data. To accommodate additional export configuration requirements, it is now possible to design and implement custom export plugins tailored to individual user's or project's needs.

As illustrated in Figure 7, the export widget is located on top of the search results and embedded only, when results exist (following the aforementioned "extras on demand" principle). The location of the widget and the icon used for export is part of the plugin's configuration. The plugin sends a request to KorAP's backend for the results to be exported and subsequently converts them into the requested file format. Plugins can be implemented not only to send or receive data from a remote service, but also to toggle on or off a remote service or invoke an assistant that interacts with the query line.

Example 2: Integration of External Resources

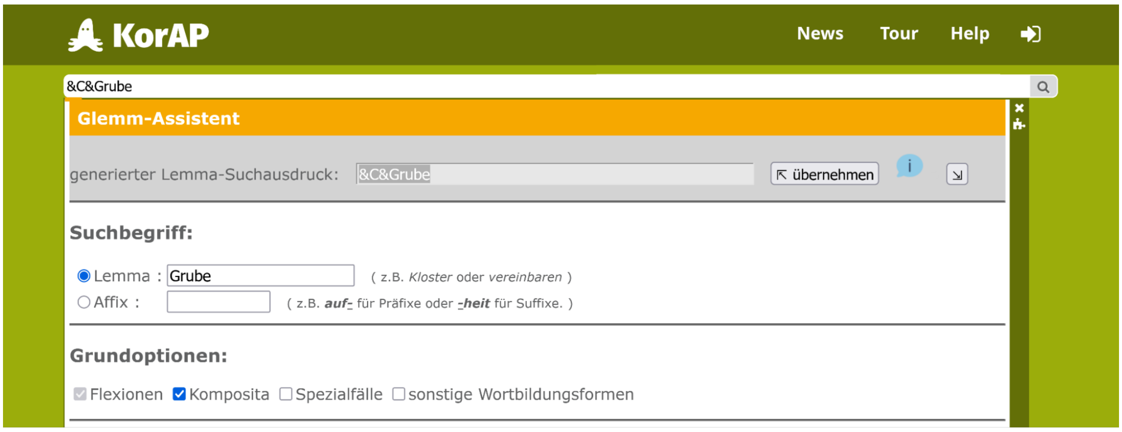
As the size of the corpora searched increases, the word form lists generated by search expressions come into focus for developers. In COSMAS II, these lists are generated from at least three types of search expressions: expressions with simplified wildcards (e.g. "`Ab*ung+mittel?`"), regular expressions (e.g. "`#REG(^Ab[a-z]*ung[s]?mittel[sn])`") and lemma expressions (e.g. "`&C&Mittel`", i.e. compounds of the lemma "Mittel"). These lists may well reach sizes of several 10 or 100 thousands of word forms (= types), and even more in a large corpus (e.g. 304,374 list entries for the compounds of "`&Haus`" on a selection of DeReKo).

While some users will be happy to get more hits, others may even find it useless if they are unable to edit the list and weed out unwanted word forms before performing the search. Browsing through numerous results in search of the desired hits can be very time-consuming or even impossible, since there is no best-fit order COSMAS II and KorAP could sort the results accordingly. Having too many false positives makes a quantitative approach difficult. Last but not least, as with COSMAS II, the longer the list, the longer the user has to wait for the results (up to several minutes). To address this issue, COSMAS II offers editable word form lists that can be sorted and whose entries can be switched on and off and saved for repeated use. It is also necessary to provide filters which allow users to switch off ranges of word forms using substring matching or word frequency.

The Kalamar plugin system presented above is a welcome opportunity to adopt modules from COSMAS II which support the issue just outlined without having to carry out integration work in KorAP. First, a web service called GlemmService (because it obtains its data from the so called "German Lemmatizer" [Bel94]) was integrated, which enriches a search query for each basic form found with the corresponding inflectional forms and/or compounds. This service can be toggled on or off using a Kalamar widget and is invoked during query processing.

The GlemmService also delivers a wizard which is loaded via another widget located below the search input window. This wizard assists users to insert the desired combination of lemmatization options into the search expression (see Figure 8, excerpt from the upper part).

A further service currently under development is the Word Form List Service which will enable users to visualize, sort and edit word form lists generated e.g. by the GlemmService to suit their



&C&Grube

Glemm-Assistent

generierter Lemma-Suchausdruck: &C&Grube ⇱ übernehmen

Suchbegriff:

● Lemma : Grube (z.B. *Kloster* oder *vereinbaren*)

○ Affix : (z.B. ***auf-*** für Präfixe oder ***-heit*** für Suffixe.)

Grundoptionen:

☑ Flexionen ☑ Komposita □ Spezialfälle □ sonstige Wortbildungsformen

Figure 8: Adding a search expression via the wizard loaded into the search widget area (upper part)

needs as detailed above. To illustrate the interaction between the plugin and Kalamar, it should also be noted that this plugin would benefit from a cache resource offered by Kalamar to store the edited lists.

3.3 Evaluating

Evaluation is a fundamental aspect of software development and serves to systematically measure the quality and suitability of software applications. In computational linguistics, evaluation is likewise an established practice (cf. recent LREC conference proceedings). In corpus linguistics, however, evaluation often remains neglected in favour of the pursuit of more data and innovative methods for corpus processing. This trend is especially evident today, as much of the field's attention is focused on the evaluation of Large Language Models (LLMs). In contrast, there are only a few studies that systematically examine, for instance, the search functionality of existing corpus query tools in terms of accuracy, speed, and performance — either in comparison with other tools (see e.g., [CRM20]) or across varying corpus sizes (see e.g., [Sch12]).

More commonly, evaluations focus on the functional aspect of linguistic software: *What can users actually do with the tool?* Examples of studies comparing different systems based on a set of software features include [FSB12, BBK17, Davnd].

With regard to user-oriented evaluations of linguistic software, studies such as [SF07, SZR08, Lla12] provide insights into the process of collecting user requirements and evaluating the usability of user interfaces.

With our study, we aim to contribute to this research field by presenting a systematic evaluation of the DGD user interface, using think-aloud experiments. Considering the time and costs involved in conducting useful and meaningful usability testing, we think it is important that we share the experiences in the field of usability evaluation and demonstrate how established evaluation methods from software engineering can be adapted and applied to the development of linguistic research software.

Goals and Methodology

With the primary goal of identifying limitations of the user interface and potential starting points for improving DGD design, the following questions served as the basis for the evaluation:

- Do new DGD users find the desired information and use the intended navigation paths and user interface elements?
- Do they locate the desired information quickly, or do they require several attempts because they intuitively search elsewhere or do not understand the underlying navigation principles?
- Which terms used by the platform are unclear or confusing to new users?

The think-aloud method, which we selected as our approach for this evaluation study, goes beyond simply observing user behaviour. It not only makes it possible to watch what participants do, but also encourages them to verbalize their thoughts while interacting with the software. This provides insights into their reasoning processes and helps to uncover what participants do not understand or how they might misinterpret certain aspects of the user interface.

Initially, the study design included ten participants who had never worked with DGD before but had experience in using corpus analysis platforms. However, after conducting sessions with the first three participants, a substantial amount of data had already been collected, and it was decided to analyze this data material before proceeding with further experiments. The analysis provided enough of valuable findings, identifying several usability issues in the DGD design. Consequently, the decision was made not to recruit any further participants, despite recommendations in the literature suggesting a minimum of four or five participants for such studies (cf. [Vir92]).

It also should be noted that the think-aloud experiments described in this paper were a supplementary component of a larger user study, which also included an extensive user survey with 669 participants and ten contextual interviews (see [FFH$^+$16]). The three think-aloud experiments presented in the following can thus be considered as a pilot study to test the suitability of this method for evaluating corpus search platforms.

Three participants (two men, one woman; aged 25–34) took part in the think-aloud experiment. All held academic degrees in Linguistics and worked as research associates at the IDS. All had experience working with corpora, though their backgrounds varied: one participant specialized in written language corpora and regularly used other corpus analysis tools, but had no experience with spoken data. Another worked extensively with spoken language, particularly in phonetics, conversation analysis, and signal processing, and was familiar with different transcription and annotation tools but not with corpus search software. The third participant had extensive experience with various corpus analysis tools and even developed similar software.

The experiment consisted in observing test persons working with DGD on some predefined tasks, asking them to speak aloud their thoughts, feelings and opinions about their interaction with the software at all times. During the think-aloud experiment, it is not allowed to interrupt the test person. In a traditional "strong" think-aloud experiment, the administrator intervenes only after a certain time when the test person stops talking, and only by reminding him or her to "please keep talking" [OMHA10]. In our experiment, we have chosen the "relaxed" form of the think-aloud experiment where the administrator is not bound to the time and to limited verbal statements, which obviously simplifies the test management efforts. To get as much information as possible we combined the "relaxed" method with the "coaching" method that allows to make a break after every task and ask the test person for task-specific feedback (with the question "How was it?"). If the feedback was felt to be too limited, the administrator was allowed to ask some additional questions (e.g. "Why did you first go to ...?", "Did you expect that ...?"). At the end of the experiment, the test person was asked about his/her general impression of DGD and his/her opinion about the design and intuitive usability of the program.

The experiment was audio and video recorded with the camera directed to the monitor to capture the probands' mouse movements on the screen, and we used the same technical environment (Windows PC, Mozilla Firefox, 24"-monitor) for every experiment. The DGD developer was not present in the room during the experiment to avoid any influence on the test person. The whole experiment took one hour and included five task blocks with a total of 16 tasks covering the core application functions: registration, browsing (getting corpus information), querying corpora, using filters, downloading data and managing virtual collections.

The tasks were first tested in a pilot think-aloud experiment and revised afterwards, especially focussing on clear, non-ambiguous and simple instructions. The challenge was also to prepare tasks with the appropriate level of difficulty. Very simple tasks like "Go to Page X, Press Button Y" would not provide the feedback we were interested in. On the other hand, relatively complex tasks like "Create a virtual collection on the topic you are interested in" would let the test persons think about the content ("What do I do?" instead of "How do I do it?") and would distract them from verbalising their behaviour.

To analyse the think-aloud data we used two quantitative measures — *accuracy* and *completeness* (cf. definition of effectiveness in ISO 9241-11:2018[14]). We checked whether the test person could complete the task at all and counted the deviations in performing tasks. The time the probands needed to perform a task was not measured in order to avoid the feeling of scrutiny. Furthermore, the relevant data was annotated with ELAN [Slo14] and analyzed qualitatively.

Quantitative Analysis

In all three think-aloud experiments, more than 80 percent of tasks could be completed by the test persons (cf. Table 1). A task was classified as completed when the proband could find the information required in the task, even if he or she did accomplish it in a manner different from the one previewed by investigators.

Every task was divided into a sequence of simple actions. In order to complete the task, the test person was supposed to perform all actions in the specified sequence. If the proband intuitively performed other actions by e.g. going to another tab or choosing the wrong menu button, this was counted as a deviation. Overall, we recorded a few deviations, the accuracy was between 0.75 and 0.78. Some actions could not be evaluated because the test person had chosen a different way to complete the task and did not come to the view where these actions could be performed

[14] ISO 9241-11:2018(en): Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts; https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en

[15] The last task was omitted due to time constraints.

Table 1: Summary of the quantitative analysis

	Experiment 1	**Experiment 2**	**Experiment 3**
Total number of tasks	15[15] (100%)	16 (100%)	16 (100%)
Number of completed tasks	13 (86.7%)	14 (87.5%)	14 (87.5%)
Number of tasks not completed	2 (13.3%)	2 (12.5%)	2 (12.5%)
Completeness	0.87	0.88	0.88
Total number of actions	53 (100%)	59 (100%)	59 (100%)
Number of actions not evaluated	0	9 (15.3%)	8 (13.6%)
Number of actions evaluated	53 (100%)	50 (84.7%)	51 (86.4%)
Number of deviations	13	11	11
Accuracy (number of actions evaluated – number of deviations) / (total number of actions – number of actions not evaluated)	0.75	0.78	0.78

at all.

One significant outcome of our experiments was the identification of tasks that multiple participants either failed to complete or completed with similar deviations. These tasks were thoroughly analyzed to determine potential causes of failure.

Example 1: In one of the tasks, participants were asked to determine the number of tokens in a specific corpus. Two of them intuitively searched for this information in the “Corpus Description”, where it was not available. Instead, the token count could only be found by scrolling down to the “Quantification” section. Based on this observation, a recommendation was made to display the token count directly within the “Corpus Description” section to improve accessibility.

Example 2: During another task, all three participants struggled to use the context filter to refine KWIC (*Keyword in Context*) results by specifying multiple optional tokens. Although the participants indicated in the survey that they were familiar with regular expressions, it took them several attempts to create a working pattern. Based on these observations, it was recommended to improve the documentation by adding more illustrative examples. Additionally, a dropdown menu containing sample expressions could be placed near the filter input field, allowing users to test predefined examples. This would reduce the need for users to formulate expressions from scratch and instead enable them to adapt existing samples to their specific needs.

Qualitative Analysis

The qualitative analysis of the experiment data consisted of the selection of relevant user statements containing positive or negative remarks, as illustrated in Table 2.

The fact that positive user statements are rare should not necessarily be seen as an indication of bad usability per se, since the nature of the experiment itself could lead the test person to focus more on critical feedback when confronted with situations different from the ones expected. We

Table 2: Sample user statements from think-aloud experiments

	User statement	Concerned GUI element
1.	This is too small for me, my eyes are not the best. I make it bigger.	general font size
2.	What puts me off here is that I have to enter a lot of information.	registration form
3.	I go back to the home page. I think this is taking too long.	speed
4.	I like it that something like this is offered.	additional corpus materials
5.	I would have expected this under "corpus description". But I can't find it here.	information about corpus duration

therefore did not compare the number of negative and positive statements, but focussed on the reasons for user criticism (e.g. software speed, too small font etc.) and identified the areas of DGD to which these comments referred. Particular attention was given to instances where multiple participants expressed negative feedback regarding the same software components. After identifying the corresponding components, we reviewed their design and developed suggestions for their improvement.

Summary

The outcomes of the think-aloud experiments primarily helped us to unveil software limitations (e.g. missing links between some DGD views, missing "stop" or "search field clean" buttons) and to identify the software parts where the users' understanding of the software strongly differs from those of the DGD developer (e.g. the registration procedure, the structure and terminology of the main menu). Based on the collected results, we were able to elaborate concrete recommendations on how to improve the DGD usability. As a result, many new features and GUI elements have been added to the system over the past few years.

The use of this user-oriented evaluation methodology in the field of corpus linguistics is new, but we could successfully adopt it for testing the corpus search platform DGD. Looking ahead, we plan to apply this approach to KorAP and to conduct such evaluations regularly across all our corpus search platforms. Usability should not be underestimated in research software development. Developers benefit significantly from the knowledge their users share with them and user satisfaction should be regarded as a key factor influencing the development of research applications.

4 Dealing with Legacy Software

Maintaining legacy software presents a unique set of challenges, particularly for linguistic research institutions that curate extensive corpora and provide critical services for internal and external researchers. These services must ensure reliability and reproducibility, as scientific rigor depends on consistent findings and sound results. Maintaining the software infrastructure underlying these tools presents persistent challenges. Legacy systems, often developed years ago, form the backbone of critical services, yet their upkeep demands balancing innovation with stability. This tension is exacerbated by the evolving needs of researchers, who require flexible tools to analyze ever-growing datasets while adhering to strict methodological standards.

A primary obstacle lies in regular software updates and addressing Common Vulnerabilities and Exposures (CVEs)[16]. Many systems rely on outdated dependencies, making them susceptible to security risks. Patching these vulnerabilities often requires extensive code revisions, particularly in monolithic architectures where updates risk destabilizing core functionalities. Automated update mechanisms are limited by the heterogeneity of deployed systems, necessitating manual intervention. This process consumes significant resources, diverting attention from feature development. Furthermore, tracking and prioritizing CVEs demands specialized expertise, which is scarce among teams focused primarily on linguistic research rather than software engineering.

Running services reliably introduces additional complexity, especially when handling diverse user inputs. Requests range from simple keyword searches to complex workflows involving APIs that connect user-provided corpus data with custom code snippets for data processing. Ensuring consistent execution across these scenarios requires rigorous input validation and sandboxing, which complicates system design. For instance, allowing users to conduct corpus analyses through APIs using our supported R[17] and Python[18] clients for KorAP [KDM20] demands secure execution environments to prevent resource exhaustion or malicious behavior. Maintaining reproducibility further requires precise versioning of processing pipelines, as even minor changes in algorithms or dependencies can alter results, undermining scientific validity.

Security concerns escalate with features involving user-written code. While empowering researchers to implement custom analyses, these capabilities create attack vectors if not properly sandboxed. Even simple, overlooked vulnerabilities could affect resource allocation, allow privilege escalation, or denial-of-service attacks. Mitigating these risks requires continuous auditing and adopting least-privilege execution models, which demand expertise rarely found in linguistics-focused teams, while logging and monitoring for anomalous behavior adds overhead, straining already limited computational resources.

Example 1: COSMAS II

After REFER and COSMAS I and II, KorAP is now the 4th generation of corpus research systems for written corpora developed at IDS. REFER was written in Fortran 77 and assembler in the 1980s [Brü89], COSMAS I then in C and finally offered a user interface designed in HTML[19]. COSMAS II is also realized in C on the server side with all database construction tools and a script-based access, while the user interfaces were developed successively in C++, Java with Struts and finally Java with Java Server Faces; to enable communication between Java and C, a separate JNI wrapper (Java Native Interface) needed to be programmed. Finally, KorAP is based

[16] https://www.cve.org/

[17] https://github.com/KorAP/RKorAPClient

[18] https://github.com/KorAP/PythonKorAPClient

[19] See Cyril Belica: “The overall corpus linguistic concept of the COSMAS platform”; https://www.ids-mannheim.de/digspra/kl/projekte/cosmas-i/gesamtkonzept

on a newly designed microarchitecture that uses programming languages and frameworks such as Mojolicous[20] (Perl), JavaScript, Java, Go, etc. and provides open APIs with supported clients written in R and Python.

Each of these new beginnings was motivated by the fact that the use of new programming environments and frameworks meant that more powerful software could be developed with less development effort, larger and more complex data volumes could be managed and new concepts could be implemented more quickly. Nevertheless, development time is a factor that should not be underestimated when planning a new search system. A pragmatic approach is therefore to ensure that the system to be developed and its predecessor can be operated in parallel over a longer period of time. If this requirement is met, the new system can primarily implement new ideas as a kind of proof of concept, while all the proven functionalities of the predecessor are gradually transferred to the new system before it can be replaced.

COSMAS II replaced its predecessor in 2003 and has been in use ever since. 10 years later, it became apparent that DeReKo (the corpus to be managed) would grow much faster than originally envisaged in the software concept. This was a crucial feature where COSMAS II was slowly becoming a legacy system, as it would soon no longer be able to accommodate DeReKo in a single database. In fact, today the entire DeReKo is spread across four databases, so that a search across the entire DeReKo, if desired, must be carried out by users in several steps. Looking at the life cycles of the three user interfaces mentioned, it is noticeable that they were technically obsolete even after a shorter period of time.

Nevertheless, the term *legacy system* is not necessarily to be understood negatively, as we will see in this section. Nor does it refer exclusively to the service life of the software, but also to the occurrence of a number of symptoms [RSVT24]. How some of these symptoms relate to COSMAS II will be presented and discussed below:

- Maintainability of the sources: Some of the programming languages (GNU-C, Java and JNI) and technologies (Tomcat and JSP) involved are 30 years old or more, but can still be used without any problems as they are continuously maintained and further developed by a strong community. The very straightforward list of technologies used is also an advantage for understanding and maintaining the overall system.

- Hardware locking: The programs were written in GNU-C from the outset with portability in mind. This made it relatively easy to port them to different architectures over time (approx. 5–6 ports so far). Most recently, the sources of COSMAS II were ported from SOLARIS to Linux, whereby only a very small part contained Linux-incompatible functions that had to be adapted.

- The central free[21] *Managing Gigabytes* library [WMB99], which takes over the indexing and compression of the corpus data, is characterized by very high quality and stability and has been in use for over 20 years without any failures. As it is open source, troubleshooting would be possible in the event of a problem, but probably not easy (it is no longer officially supported). To prevent the older data width in the data structures from overflowing, size-limited, parallel sub-databases were introduced. This partitioned architecture in turn has

[20] https://mojolicious.org/

[21] Licensed under GNU GPL v2

the advantage that the indexed data can be accessed more efficiently (e.g. by parallel threads and Data Skiping). From the outset, it was decided not to modify the sources of this library in order to facilitate the integration of newer versions. In addition, a program layer for general read and write access to index structures was introduced, which makes the functioning of the MG library transparent; in the early development phase of COSMAS II, it would thus have been conceivable to replace the MG library with another one.

- The knowledge about the architecture of COSMAS II, the processes within the server and during the database building are still available in-house. Necessary changes and enhancements to the software can be made at the designated points, thus avoiding unnecessary redundancies and code changes that are contrary to the design. This is an important technical aspect in order to avoid a deterioration in the maintainability and the execution of the sources.
- Documentation: In the initial phase, COSMAS II was developed and funded as part of two projects, so that the concepts and initial results were published, while the documentation of the programs and functions shifted more and more mainly to the programm sources and scripts: the usual function and interface documentation, references to design aspects, discussions and URLs to background information can be found there. Changes were often marked with the date and initials of the author; over time, the projects were first versioned using Apache Subversion SVN[22], and later on using git[23]. How detailed the documentation is depends strongly on the individual developers involved in the project. The longer someone has worked on the project, the more extensive the explanations often are. On the other hand, it can be said that external developers usually have little or no time to devote to this aspect.
- Security vulnerabilities have been closed so far. Either the self-written source code was affected and could be enhanced, or it involved free program libraries for which new versions were available. Although COSMAS II is proprietary software, it contains only open-source program libraries and is independent of commercial libraries and their maintenance.
- Data security: Anomalies in data access usage can be monitored script-based and approaches to data theft can be prevented by appropriate extensions in the server code.
- Maintenance of the corpus data in the database: Faulty corpus data can be reindexed subsequently, while metadata, frequencies, word annotations etc. can also be corrected at any time using our own tools on the indexed data structures.

Example 2: grammis

The modernization of grammis also exemplifies these challenges. Originally built on PHP[24] 5.6 with a tightly coupled CakePHP[25] framework and an Oracle database[26], its architecture hindered

[22] https://de.wikipedia.org/wiki/Apache_Subversion
[23] https://git-scm.com
[24] https://www.php.net/
[25] https://cakephp.org/
[26] https://www.oracle.com/database/

scalability. The framework's rigid ORM layer struggled with Oracle's SQL dialect, causing frequent query-generation errors. Upgrading PHP versions broke backward compatibility with legacy extensions, while CakePHP's outdated dependency tree resisted modern package manager integration. Migrating to a microservices architecture required decoupling database interactions, rewriting business logic with a modern HTMX[27] framework, and re-implementing the backend in Go[28], all while maintaining feature parity for active users. Addressing the Oracle dependency also highlighted tensions between open-source ideals and practical constraints. While IDS as a research institute must prioritize open-source solutions, legacy contracts and performance-critical workloads tie it to Oracle's proprietary database. A migration to PostgreSQL, the adoption of a CI/CD pipeline, and upskilling researchers in modern practices such as infrastructure-as-code, albeit competing with the primary research objectives, are our upcoming tasks.

5 Summary and Conclusion

Empirical linguistic research depends on large, well-annotated corpora of both written and spoken language, accompanied by rich metadata. IDS maintains two flagship corpora — DeReKo and the AGD — and provides multiple online platforms to query and analyze these resources.

These platforms are primarily aimed at users with a background in the humanities, which is taken into account when designing the access options. Therefore, different access options are offered depending on the level of technical and computational linguistic knowledge and taking into account the different research questions of the users. Multi-level access is offered for the written corpora, with the lowest level of access being via the KorAP and COSMAS II user interfaces. The spoken corpora allow low-threshold access with DGD and more sophisticated data access with ZuRecht.

The development of this software poses many challenges, which we have illustrated in this article using various examples and which should be taken into account in the development of future systems. Some of these challenges are not apparent to the user, such as dealing with legal (e.g. intellectual property or personal rights), technical (e.g. regarding the handling of legacy software) and scientific issues (e.g. regarding the reproducibility of results). Other challenges have a direct impact on the user experience and require continuous improvement, expansion and evaluation of the user interfaces of corpus analysis systems.

Looking forward, the insights gained from spoken-language evaluation studies will inform upcoming enhancements to written-language platforms. Continued emphasis on user-centered design, open standards, and modular architectures will be crucial for navigating future challenges in order to meet both the growing data volumes and the changing user requirements.

Acknowledgements: We thank the anonymous reviewers for their valuable feedback, which helped improve the clarity and quality of this paper. The authors disclose the use of LLMs as a writing aid for selective translations and formulations throughout this paper, which is based on the authors' notes, ideas and concepts. The dominant models used include GPT-4 and DeepL. The authors retain full responsibility for the content.

[27] https://htmx.org/

[28] https://go.dev/

Bibliography

[al-94] D. al-Wadi. COSMAS – ein Computersystem für den Zugriff auf Textkorpora. Version R.1.3-1. Institut für Deutsche Sprache, Mannheim, 1994.

[Ant22] L. Anthony. AntConc (Version 4.0.3). Waseda University, Tokyo, Japan, 2022.

[BBK17] M. Brouwer, H. Brugman, M. Kemps-Snijders. MTAS: A Solr/Lucene Based Multi Tier Annotation Search Solution. In *Selected Papers from the CLARIN Annual Conference 2016. Linköping Electronic Conference Proceedings 136*. Pp. 19–37. 2017.

[Bel94] C. Belica. A German Lemmatizer. Technical report MLAP93-21/WP2, IDS, Mannheim, 1994.

[BFF$^+$12] P. Bański, P. M. Fischer, E. Frick, E. Ketzan, M. Kupietz, C. Schnober, O. Schonefeld, A. Witt. The New IDS Corpus Analysis Platform: Challenges and Prospects. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Pp. 2905–2911. European Language Resources Association (ELRA), Istanbul, Turkey, May 2012.

[BFS21] J. Batinić, E. Frick, T. Schmidt. Accessing Spoken Language Corpora: An Overview of Current Approaches. *Corpora* 16(3):417–445, Nov. 2021. doi:10.3366/cor.2021.0229

[Bod05] F. Bodmer. COSMAS II – Recherchieren in den Korpora des IDS. *SprachReport* 21(3):2–5, 2005.

[Brü89] T. Brückner. *REFER – Benutzerhandbuch*. Institut für Deutsche Sprache, Mannheim, Germany, 1989.

[CRM20] M. Coole, P. Rayson, J. Mariani. LexiDB: Patterns & Methods for Corpus Linguistic Database Management. In Calzolari et al. (eds.), *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Pp. 3128–3135. European Language Resources Association, Marseille, France, May 2020.

[Davnd] M. Davies. Comparison of the BYU Corpus Architecture, Corpus Workbench (SketchEngine), and Corpus Workbench (BNCweb). https://www.english-corpora.org/architecture.asp, n.d.

[DBK19] N. Diewald, V. Barbu Mititelu, M. Kupietz. The KorAP User Interface. Accessing CoRoLa via KorAP. *On design, creation and use of the Reference Corpus of Contemporary Romanian and its analysis tools. CoRoLa, KorAP, DRuKoLA and EuReCo. Revue Roumaine de Linguistique* 64(3), 2019.

[DFAR04] A. Dix, J. Finlay, G. D. Abowd, B. Russell. *Human-Computer Interaction*. Pearson, Harlow, England, 2004.

[DHM+16] N. Diewald, M. Hanl, E. Margaretha, J. Bingel, M. Kupietz, P. Bański, A. Witt. KorAP Architecture – Diving in the Deep Sea of Corpus Data. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Pp. 3586–3591. Portorož, Slovenia, 2016.

[DMK21] N. Diewald, E. Margaretha, M. Kupietz. Lessons Learned in Quality Management for Online Research Software Tools in Linguistics. In Lüngen et al. (eds.), *Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-9)*. Pp. 20–26. Limerick, Ireland (online), 2021.

[DND17] J. de Does, J. Niestadt, K. Depuydt. Creating Research Environments with BlackLab. In *CLARIN in the Low Countries*. Ubiquity Press, Dec. 2017. doi:10.5334/bbi.20

[DR23] A. Deppermann, S. Reineke. Zur Verwendung von Metadaten in der interaktionsanalytischen Arbeit mit Korpora – Am Beispiel einer Untersuchung anhand des Korpus FOLK. In Beißwenger et al. (eds.), *Korpusgestützte Sprachanalyse. Grundlagen, Anwendungen und Analysen*. Pp. 245–259. Narr Francke Attempto, Tübingen, 2023.

[DS24] N. Diewald, H. Stallkamp. Kalamar. Zenodo, Nov. 2024. doi:10.5281/zenodo.14179138

[EH15] S. Evert, A. Hardie. Ziggurat:A New Data Model and Indexing Format for Large Annotated Text Corpora. In Bañski et al. (eds.), *Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora*. Pp. 21–27. Institut für Deutsche Sprache, Lancaster, July 2015.

[FFH+16] C. Fandrych, E. Frick, H. Hedeland, A. Iliash, D. Jettka, C. Meißner, T. Schmidt, F. Wallner, K. Weigert, S. Westpfahl. User, Who Art Thou? User Profiling for Oral Corpus Platforms. In Calzolari et al. (eds.), *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Pp. 280–287. European Language Resources Association (ELRA), Portorož, Slovenia, May 2016.

[FHW23] E. Frick, H. Helmer, F. Wallner. ZuRecht: Neue Recherchemöglichkeiten in Korpora gesprochener Sprache für Gesprächsanalyse und Deutsch als Fremd- und Zweitsprache. *Korpora Deutsch als Fremdsprache* 3(1):44–71, 2023. doi:10.48694/kordaf.3730

[FS25] E. Frick, T. Schmidt. Querying Spoken Language Data. In Bański et al. (eds.), *Harmonizing Language Data. Standards for Linguistic Resources*. Digital Linguistics 4, pp. 339–376. De Gruyter, 2025.

[FSB12] E. Frick, C. Schnober, P. Bański. Evaluating Query Languages for a Corpus Processing System. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), 2012.

[Kam18] P. Kamocki. The Argument for 'Non-Consumptive Use' in the EU: How Copyright Could Be Redefined to Allow Text and Data Mining. In *Intellectual Property Perspectives on the Regulation of New Technologies*. Pp. 237–258. Edward Elgar Publishing, 2018.
doi:10.4337/9781786436382.00016

[KBD^{+}24] M. Kupietz, P. Bañski, N. Diewald, B. Trawiński, A. Witt. EuReCo: Not Building and Yet Using Federated Comparable Corpora for Cross-Linguistic Research. In Zweigenbaum et al. (eds.), *Proceedings of the 17th Workshop on Building and Using Comparable Corpora (BUCC) @ LREC-COLING 2024*. Pp. 94–103. ELRA and ICCL, Torino, Italia, May 2024.

[KBKW10] M. Kupietz, C. Belica, H. Keibel, A. Witt. The German Reference Corpus DeReKo: A Primordial Sample for Linguistic Research. In *Proceedings of the 7th Conference on International Language Resources and Evaluation*. Pp. 1848–1854. European Language Resources Association (ELRA), Valletta, Malta, 2010.

[KDM20] M. Kupietz, N. Diewald, E. Margaretha. RKorAPClient: An R Package for Accessing the German Reference Corpus DeReKo via KorAP. In Calzolari et al. (eds.), *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*. Pp. 7017–7023. 2020.

[KDM22] M. Kupietz, N. Diewald, E. Margaretha. Building Paths to Corpus Data. In Fišer et al. (eds.), *CLARIN*. Digital Linguistics 1, pp. 163–190. De Gruyter, 2022.
doi:10.1515/9783110767377-007

[KL14] M. Kupietz, H. Lüngen. Recent Developments in DeReKo. In Calzolari et al. (eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Pp. 26–31. European Language Resources Association (ELRA), Reykjavik, Iceland, May 2014.

[LHB10] W. Lidwell, K. Holden, J. Butler. *Universal Principles of Design*. Rockport Publishers, Beverly, Massachusetts, 2010.

[Lla12] L. C. Llanos. Designing a Search Interface for a Spanish Learner Spoken Corpus: The End-User's Evaluation. In Calzolari et al. (eds.), *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Pp. 241–248. European Language Resources Association (ELRA), Istanbul, Turkey, May 2012.

[LZ10] L. Lemnitzer, H. Zinsmeister. *Korpuslinguistik. Eine Einführung*. narr Studienbücher. Narr Francke Attempto Verlag, 2 edition, 2010.

[Mac07] B. MacWhinney. The Talkbank Project. In Beal et al. (eds.), *Creating and Digitizing Language Corpora: Volume 1: Synchronic Databases*. Pp. 163–180. Palgrave Macmillan UK, London, 2007.

[MDKK24] E. Margaretha Illig, N. Diewald, P. Kamocki, M. Kupietz. Managing Access to Language Resources in a Corpus Analysis Platform. In Vandeghinste and Thalassia (eds.), *CLARIN Annual Conference Proceedings 2024. 15 – 17 October 2024, Barcelona, Spain.* Pp. 163–167. CLARIN, Utrecht, 2024.

[MS09] S. Merkel, T. Schmidt. Korpora gesprochener Sprache im Netz – eine Umschau. *Gesprächsforschung* 10:71–93, 2009.

[Nie06] J. Nielsen. Progressive Disclosure. https://www.nngroup.com/articles/progressive-disclosure/, 2006.

[OMHA10] E. Olmsted-Hawala, E. Murphy, S. Hawala, K. Ashenfelter. Think-Aloud Protocols: A Comparison of Three Think-Aloud Protocols for Use in Testing Data-Dissemination Web Sites for Usability. In *Conference on Human Factors in Computing Systems – Proceedings*. Volume 4, pp. 2381–2390. Apr. 2010.
doi:10.1145/1753326.1753685

[RDS23] S. Reineke, A. Deppermann, T. Schmidt. Das Forschungs- und Lehrkorpus für Gesprochenes Deutsch (FOLK). In Deppermann et al. (eds.), *Mündlich, schriftlich, multimedial*. Pp. 71–102. De Gruyter, Berlin, Boston, 2023.

[RSVT24] S. Rosenkranz, D. Staegemann, M. Volk, K. Turowski. Explaining the Business-Technological Age of Legacy Information Systems. *IEEE Access* 12:84579–84611, June 2024.
doi:10.1109/ACCESS.2024.3414377

[Rüd20] J.-O. Rüdiger. *CorpusExplorer: Eine Software zur korpuspragmatischen Analyse*. PhD thesis, Universität Kassel, Kassel, 2020.

[Sch12] R. Schneider. Evaluating DBMS-based Access Strategies to Very Large Multi-Layer Corpora. In *Proceedings of the LREC 2012 Workshop: Challenges in the Management of Large Corpora (CMLC)*. Istanbul, Turkey, 2012.

[Sch14] T. Schmidt. The Database for Spoken German — DGD2. In Calzolari et al. (eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Pp. 1451–1457. European Language Resources Association (ELRA), Reykjavik, Iceland, May 2014.

[SF07] D. Santos, A. Frankenberg-Garcia. The Corpus, Its Users and Their Needs: A User-Oriented Evaluation of COMPARA. *International Journal of Corpus Linguistics* 12(3):335–374, Oct. 2007.
doi:10.1075/ijcl.12.3.03san

[SFF$^+$23] T. Schmidt, C. Fandrych, E. Frick, M. Schwendemann, F. Wallner, K. Wörner. Zugänge zu mündlichen Korpora für DaF und DaZ – Projekt, Datengrundlagen, technische Basis. *Korpora Deutsch als Fremdsprache* 3(1):1–12, 2023.
doi:10.48694/kordaf.3721

[SL22] R. Schneider, C. Lang. Das grammatische Informationssystem grammis – Inhalte, Anwendungen und Perspektiven. *Zeitschrift für germanistische Linguistik* 50(2):407–427, 2022.
doi:10.1515/zgl-2022-2060

[Slo14] H. Sloetjes. ELAN: Multimedia Annotation Application. In Durand et al. (eds.), *The Oxford Handbook of Corpus Phonology*. Pp. 305–320. Oxford University Press, Oxford, 2014.

[SPC$^+$17] B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, N. Elmqvist. *Designing the User Interface*. Pearson, Harlow, England, 2017.

[SS14] U.-M. Stift, T. Schmidt. Mündliche Korpora am IDS: vom deutschen Spracharchiv zur Datenbank für gesprochenes Deutsch. In *Ansichten und Einsichten. 50 Jahre Institut für Deutsche Sprache*. Pp. 360–375. IDS Verlag, Mannheim, Germany, 2014.

[SW09] T. Schmidt, K. Wörner. EXMARaLDA – Creating, Analysing and Sharing Spoken Language Corpora for Pragmatic Research. *Pragmatics* 19(4):565–582, 2009.

[SZR08] J.-P. Soehn, H. Zinsmeister, G. Rehm. Requirements of a User-Friendly, General-Purpose Corpus Query Interface. In *Proceedings of the LREC 2008 Workshop Sustainability of Language Resources and Tools for Natural Language Processing*. Pp. 27–32. Marrakech, Morocco, 2008.

[Tid06] J. Tidwell. *Designing Interfaces. Patterns for Interaction Design*. O'Reilly & Associates, Inc., 2006.

[Vir92] R. A. Virzi. Refining the Test Phase of Usability Evaluation: How Many Subjects Is Enough? *Human Factors* 34(4):457–468, Aug. 1992.
doi:10.1177/001872089203400407

[Wil19] J. Wilbur. ELAN as a Search Engine for Hierarchically Structured, Tagged Corpora. In Pirinen et al. (eds.), *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*. Pp. 90–103. Association for Computational Linguistics, Tartu, Estonia, Jan. 2019.
doi:10.18653/v1/W19-0308

[WMB99] I. H. Witten, A. Moffat, T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images, Second Edition*. Morgan Kaufmann, San Francisco, USA, 1st edition edition, May 1999.

[ZJLC09] A. Zeldes, R. Julia, A. Lüdeling, C. Chiarcos. ANNIS: A Search Tool for Multi-Layer Annotated Corpora. In *Corpus Linguistics 2009*. Liverpool, 2009.