Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 7015–7021
Marseille, 11–16 May 2020

RKorAPClient: An R Package for Accessing the German Reference Corpus DeReKo via KorAP

Marc Kupietz, Nils Diewald, Eliza Margaretha
Leibniz-Institut für Deutsche Sprache
R5 6–13, 68161 Mannheim, Germany
{kupietz | diewald | margaretha}@ids-mannheim.de

Abstract

Making corpora accessible and usable for linguistic research is a huge challenge in view of (too) big data, legal issues and a rapidly evolving methodology. This does not only affect the design of user-friendly graphical interfaces to corpus analysis tools, but also the availability of programming interfaces supporting access to the functionality of these tools from various analysis and development environments. RKorAPClient is a new research tool in the form of an R package that interacts with the Web API of the corpus analysis platform KorAP, which provides access to large annotated corpora, including the German reference corpus DeReKo with 45 billion tokens. In addition to optionally authenticated KorAP API access, RKorAPClient provides further processing and visualization features to simplify common corpus analysis tasks. This paper introduces the basic functionality of RKorAPClient and exemplifies various analysis tasks based on DeReKo, that are bundled within the R package and can serve as a basic framework for advanced analysis and visualization approaches.

Keywords: Corpus Tools, Corpus Analysis, Reference Corpora, Data Visualization, R, OAuth, Web Services

1. Introduction

With the establishment of corpus linguistic methodology in all areas of linguistics and the associated increased demands on methodological validity and sophistication, research institutions that wish to offer corpora for use by third parties are faced with the question of how they can 1.) satisfy the broad spectrum of user demands; 2.) take into account the fact that the analysis methodology itself is subject of rapid ongoing research, without 3.) infringing on the legitimate interests of the rights of authors and other right holders that are notoriously affected when dealing with linguistic research data.

The open source corpus analysis platform KorAP[1] (Bański et al., 2012) is part of the CLARIN / CLARIAH infrastructure and one of the main scientific tools for querying and analysing the German Reference Corpus DeReKo, which is with 46.9 billion tokens (Leibniz-Institut für Deutsche Sprache, 2020) and 45.000 registered users one of the most important empirical bases for German linguistics (Kupietz et al., 2010, 2018b). Under the umbrella of the EuReCo initiative (Kupietz et al., forthcoming), it also provides access to the Contemporary Corpus of the Romanian Language CoRoLa[2] (Romanian Academy, 2017; Barbu Mititelu et al., 2018; Cristea et al., 2019) and the Hungarian National Corpus HNC (Hungarian Academy of Sciences, 2018; Oravecz et al., 2014).

KorAP attempts to address the challenge of corpus data accessibility with a number of complementary approaches. At the level of the Web user interface, for example, the systematic application of an *extras on demand* strategy (Tidwell, 2006, p. 45f) is intended to simultaneously meet the needs of frequent, occasional, expert, and novice users (Diewald et al., 2019). However, since corpus linguistic methodology, including appropriate visualization of analysis results, itself is subject of ongoing research, it is fundamentally impossible to cover all potential demands with a single scientific tool and its Web GUI. For this reason Kupietz et al. (2018a) proposed support for multiple levels of data access for user-supplied code: the corpus level, the API level, the multiple (backend) instance level, and the open source level. On these levels we would like to provide access to the corpus data for externally developed software, so that it can be used as efficiently as possible in different application scenarios, without contravening typical license conditions.

In this paper we introduce the RKorAPClient package, a new way to easily access KorAP functionality programmatically at the API level and hence, as shown in various examples, the German Reference Corpus DeReKo. The following sections should provide an insight into the conceptual background of RKorAPClient the motivation of design decisions, its location in the KorAP architecture (for more, see Diewald et al., 2016) and at the same time serve as an introduction to the core functionalities of RKorAPClient, also for new users.

2. API Access

KorAP provides all its functionality via Web APIs[3]. The major user interface to KorAP, Kalamar (Diewald et al., 2019), is a web-based client interacting with the API, translating user interactions into Web API calls and visualizing the responses as part of the user interface. This means that each implemented functionality of the user interface requires a previously implemented API, which is often more advanced than what is accessible to the user through the frontend.

2.1. Functional Extensions: Server-side vs. Client-side

While KorAP has already supported a wide range of search functions (both for occurrence search and for creating virtual corpora), the native support of analysis functions such as sorting and aggregation of results is still very limited. On

[1] For accessing DeReKo: https://korap.ids-mannheim.de/, Source code: https://github.com/KorAP/

[2] http://corola.racai.ro/

[3] Documentation: https://github.com/KorAP/Kustvakt/wiki

the basis of already available functions (accessible via Web API; see Kupietz et al., 2019), however, analyses can already be carried out externally by sorting or aggregating all results subsequently in a separate second step after the initial step of searching, but with potential performance losses. To simplify this second step for users (before a high-performance, native solution is available as part of the search engine), there are two possible options (see Figure 1):

1. Provide a *server-side implementation* as part of the Web API;
2. Provide a *client-side implementation* as a software library.

The first option is tempting: users would not notice the limitations of the system and could, once a native solution exists, benefit directly from the changes of the underlying Web service without having to change API calls in their analysis or development environment. This solution would have a significant disadvantage though: the convenience with which the user would use the non-native analysis functions would hide the actual complexity of the execution: Collecting all search hits and the subsequent sorting and aggregation require considerable resources in terms of computing and storage capacity. In this case, the Web service would have to provide this capacity.
The second option would also allow for a comfortable use of non-native analysis functions and would limit changes in the analysis or development environment to an update of the client library once a native solution is introduced on the server side. The advantage over the first solution would be that the complexity (i.e., the need for computing and storage capacity) is on the user's side and therefore does not have to be provided by the Web service. A disadvantage, on the other hand, is the increased data transfer, as all results need to be transferred from the Web service to the client application (see Figure 1).

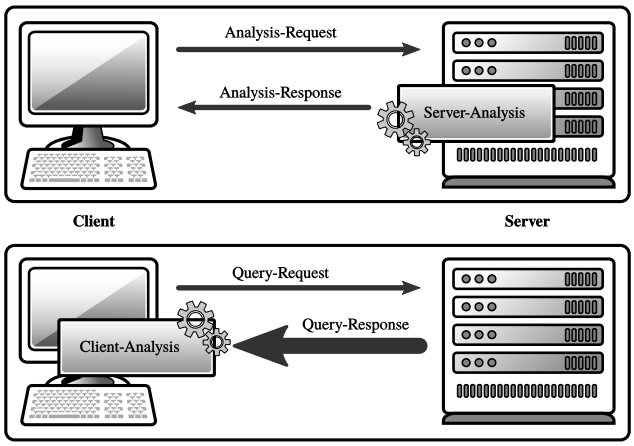

Figure 1: Server-side analysis vs. client-side analysis

2.2. API Client Libraries

An API client library is a wrapper for Web API requests specific to a certain programming language. It can be used in standalone scripts as well as form the basis for specialized web interfaces to the data. Since API interfaces in KorAP are the prerequisite for new functions in the Web interface, users of API client libraries can be supported almost without additional effort, so that no conflict of interest arises with the users of the Web interface with regard to further development.

3. The RKorAPClient Library

RKorAPClient is an R package for client-side analysis, aimed at researchers who want to perform quantitative linguistic analysis on corpus data using KorAP, without having to deal with the details of the underlying API protocol.
It is published under a BSD-2 license and available on CRAN[4] and GitHub[5]. It is implemented using S4 classes and based upon the tidyverse collection of R packages for data science (Wickham and Grolemund, 2016). Currently, it provides four basic functions that are required to access the KorAP Web API and several convenience functions on top of these which summarise and simplify typical workflows.

3.1. Basic Functions

The basic functions are oriented to those of a database client. The `KorAPConnection` constructor is used to initialize a connection object with the target server, authorization information (an OAuth access token, see Section 5.2.) and some more settings like caching policy, verbosity, etc. Being provided with default values, all of these initialization settings are optional.
The `corpusQuery` method is used to trigger the actual request to the KorAP server. In addition to the search expression and the definition of an underlying virtual sub-corpus, the query language can be specified (e.g., *Poliqarp+*, Przepiórkowski, 2004; *COSMAS-II*, al-Wadi, 1994; *ANNIS*, Rosenfeld, 2010; *FCS-QL*, OASIS Standard, 2013). The optional `vc` (virtual corpus) and `ql` (query language) parameters are also passed on to the server.
The `fetchNext` method, which operates on a `corpusQuery` object, is used to retrieve the next chunk of query results. This method returns an updated `corpusQuery`, in which, among other things, the data and metadata of all search hits can be successively accumulated.
Finally, the `corpusStats` function is used to query the size of virtual corpora (in tokens, sentences and texts) in order to be able to calculate relative frequencies.

3.2. Additional Query Functions

The most important additional functions at present are the `corpusQuery` methods `fetchRest` and `fetchAll`, which are used to retrieve search results, and the `frequencyQuery` method based on a `KorAPConnection`. The latter combines `corpusQuery` with `corpusStats` by directly calculating relative frequencies and also confidence intervals. Like it's base functions and most other functions, it is *vectorized* (Burns, 2011) regarding the parameters `query` and `vc` (virtual corpus), which means that it not only supports scalars as arguments, but also vectors over which it iterates automatically. If the vectors for `query` and `vc` are not of the same length, the `expand` parameter defaults to `TRUE`, so that the

[4] https://cran.r-project.org/package=RKorAPClient
[5] https://github.com/KorAP/RKorAPClient

function iterates over all combinations of `query` and `vc`. For example:

```
new("KorAPConnection") %>%
  frequencyQuery(
    query = c('so "genannte.?"', '"sogenannte.?"'),
      vc = paste("pubDate in", 2005:2007)
  ) %>%
  ipm()
```

returns the data frame with six rows shown in Table 1.[6] Whether the total token sizes of the respective virtual corpora (as in Figure 2) or the sum of hits for different queries on the same virtual corpus is used for the calculation of relative frequencies (as in Figure 3) is controlled by the `as.alternatives` parameter (see Figure 3 and section 4.3. for examples).

3.3. Plot Functions

The package currently also provides some extensions to ggplot2 (Wickham, 2016) and plotly (Sievert et al., 2017) as well as helper functions to directly generate interactive HTML and JavaScript plots using Highcharts[7] via the highcharter wrapper package (Kunst, 2019) (see Figure 6). Although such functions are usually not part of an API client for good reasons, we decided to include them in order to simplify the integration of frequently used plot types and to help R-beginners, who have no experience with plot packages, to get started quickly and successfully.[8]

For example, **`geom_freq_by_year_ci`** simplifies the plot of frequency curves with confidence intervals (see Listing 1 and Figure 2).[9] In addition, the function **`RKorAPClient::ggplotly`** converts ggplot2 objects created like this to plotly, in such a way that all displayed data points are linked to their corresponding queries in KorAP's Web interface via the `webUIRequestUrl` values returned from `frequencyQuery`. This is methodologically important to provide a quick overview of the observations on which the respective data points are based and to make it verifiable that they are not artifacts (Kupietz et al., 2017, p. 326f).

4. Demo Applications

In order to keep the hurdle to using the R package as low as possible, it contains some short demos for the most typical linguistic application scenarios, which often only need to be adapted to the respective concrete application.[10]

4.1. Diachronic Frequency Distributions

The programming language R is particularly suitable for statistical analysis of multidimensional data, for example for diachronic observation of linguistic phenomena in different

[6] Poliqarp is used as the default query language. In Poliqarp double quotes denote regular expressions.

[7] https://www.highcharts.com/

[8] If this decision turns out to be unmaintainable, we may need to spin off the plot and miscellaneous functions to external packages.

[9] Similar functions are provided for simplifying the generation of Highcharts plots.

[10] All examples in this paper are based on DeReKo-2019-I-W1 (Leibniz-Institut für Deutsche Sprache, 2019b), a virtual corpus containing a subset of DeReKo-2019-I (Leibniz-Institut für Deutsche Sprache, 2019a) with approximately 11 billion words.

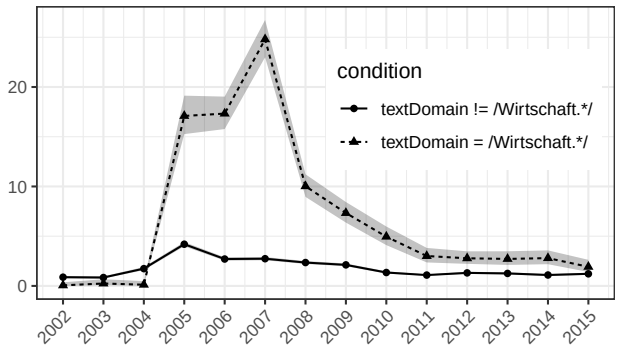

Figure 2: Observed frequencies per million of the lemma *Heuschrecke* in business-like newspaper columns (dashed line) and outside these (solid line). The ribbons around the plots indicate 95% confidence intervals. The confidence level can be configured using the optional `conf.level` parameter of the `frequencyQuery` method.

```
library(RKorAPClient)
library(ggplot2)
kco <- new("KorAPConnection")
expand_grid(
  condition = c("textDomain = /Wirtschaft.*/",
                "textDomain != /Wirtschaft.*/"),
  year = (2002:2015)
) %>%
  cbind(frequencyQuery(
    kco,
    "[tt/l=Heuschrecke]",
    paste(.$condition, "& pubDate in", .$year)
  )) %>%
  ipm() %>%
  ggplot(aes(
    x = year,
    y = ipm,
    linetype = condition,
    shape = condition
  )) +
  geom_freq_by_year_ci()
```

Listing 1: Complete R code to perform a frequency query for the lemma *Heuschrecke* over year slices from 2002 until 2015, and to plot the results as shown in Figure 2, above. Calls of functions provided by RKorAPClient are printed in red.[11]

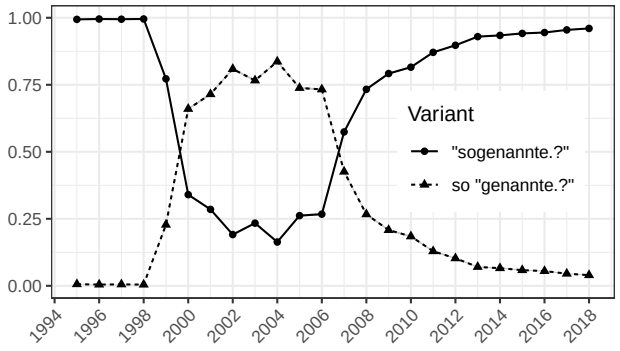

Figure 3: Proportions of observed uses of the joint and separate spelling variants of *sogenannt* over time.

	query	totalResults	vc	total	ipm	conf.low	conf.high
1	so ”genannte.?”	52480	pubDate in 2005	377248775	139.11	137.93	140.31
2	so ”genannte.?”	46897	pubDate in 2006	382454596	122.62	121.52	123.74
3	so ”genannte.?”	29943	pubDate in 2007	439384267	68.15	67.38	68.93
4	”sogenannte.?”	22854	pubDate in 2005	377248775	60.58	59.80	61.37
5	”sogenannte.?”	16243	pubDate in 2006	382454596	42.47	41.82	43.13
6	”sogenannte.?”	38086	pubDate in 2007	439384267	86.68	85.81	87.56

Table 1: Output of a frequency query with expanded combinations of `query` and `vc` parameters converted to instances per million. For space reasons, the column `webUIRequestUrl` holding a link to a corresponding query to the Web GUI, was omitted.

contexts. The metaphor of the *Heuschrecke* (German for *locust*) to name reckless private equity firms was introduced by German politician Franz Müntefering in a newspaper interview on 19 April 2005 (see Ziem, 2008, p. 192f). The distribution of this term over time and in different contexts can be ascertained and visually processed (see Figure 2) with just a few lines of R code using RKorAPClient (see Listing 1) and accessing DeReKo – here referring to the lemma layer of the TreeTagger (Schmid, 1994) annotation foundry (see line 11 in Listing 1).

Another significant type of frequency analyses over time is – in the case of DeReKo and KorAP – the empirical observation of spelling usage performed by the *Rat für deutsche Rechtschreibung* (German Council for Orthography). Typical of this application scenario is the comparison of frequencies of different spelling variants over time (see e.g., Fischer and Lang, 2019). As shown in Figure 3, the demos include an example, which plots the ratio of the jointly and separately written variants of *sogenannt* (German for *so-called*) in DeReKo, for which the recommended variant has changed twice, in 1996 and 2006 (Rat für deutsche Rechtschreibung, 2006, p. 41, p. 249).

4.2. Regional Frequency Distributions

A typical application scenario for the R client package is also the comparison of frequencies across different geographical regions. The demos contain an easily adaptable example that provides a function `geoDistrib(query)` and visualizes e.g., the geographical distribution of non-standard use of the preposition *wegen* (German for *because of / due to*) in conjunction with a dative noun (instead of a genitive noun), by means of relative frequencies in press products published in different German-speaking regions (see Figure 4).

4.3. Further Types of Analysis and Visualization

With Mosaic plots, the demo section of RKorAPClient contains just another popular form of multidimensional data visualization used in linguistics (see e.g., Wolfer and Hansen-Morath, 2018), generated by the vcd package (Meyer et al., 2006). Figure 5 shows the dependencies of indicative vs. subjunctive use on newspaper sections. In this representation the size of boxes in the mosaic are proportional to the number of observed verbs of the respective moods in the respective newspaper sections. In our example, it can be seen that the use of subjunctive in DeReKo newspaper articles is significantly more common in economics sections compared to sports and culture.

The plot is based on the following function call that uses DeReKo's MarMoT annotations on morphological features (Müller et al., 2013) :

```
1 frequencyQuery(
2   query = c("[marmot/m=mood:subj]",
3             "[marmot/m=mood:ind]"),
4   vc = c("textDomain=Wirtschaft",
5          "textDomain=Kultur",
6          "textDomain=Sport"),
7   as.alternatives = TRUE)
```

The demo section of RKorAPClient, contains many more examples, including interactive visualizations as shown in Figure 6[12], and is expected to grow. The code of the ex-

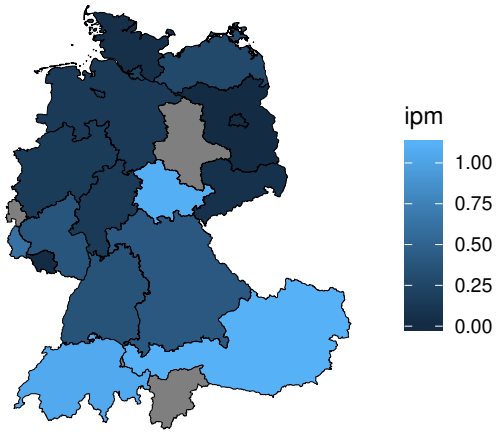

Figure 4: Observed frequencies per million of the query `wegen dem [tt/p=NN]`, meaning *wegen dem* followed by a common noun, according to the TreeTagger part-of-speech annotation, in different regions of the German-speaking area.

The examples do not require authenticated access to DeReKo and are for illustrational purposes only.

[11]The `expand_grid` (tidyr) call (starting line 6 in Listing 1) generates all combinations of years and text domain conditions in a data frame. The advantage of doing this explicitly here is that the columns `year` and `condition` of the generated data frame can be re-used in the `ggplot` call in lines 18, 20-21.

[12]Note that the more detailed query `corpusQuery(kco, 'LeserIn | LeserInnen', "textType = /Zeit.*/", fields="corpusTitle") %>% fetchAll()` reveals that almost 90% of the total hits for this non-standard, gender-neutral variant stem from a single newspaper (taz).

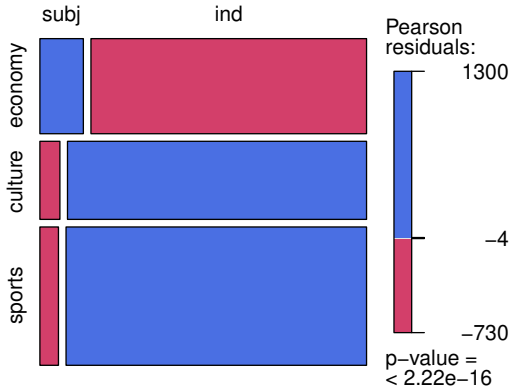

Figure 5: Mosaic plot visualization of indicative vs. subjunctive verb use in relation to different newspaper sections.

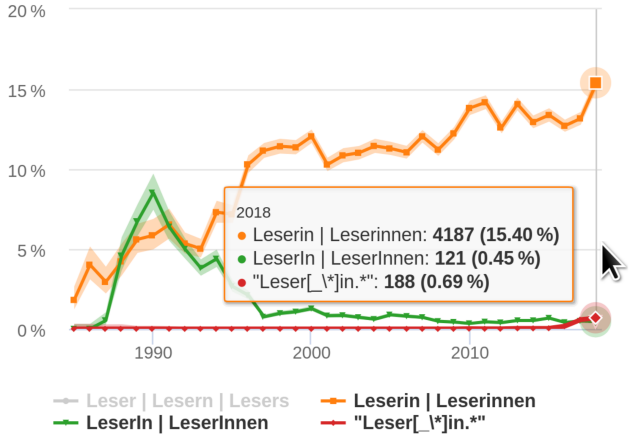

Figure 6: Screenshot of an interactive visualization of the frequency development of differently gendered variants of the lemma *Leser* (German for *reader*), using the integrated **hc_freq_by_year_ci** helper function to generate Highcharts plots. The data series for querying the grammatically male gender variant (greyed out label) is switched off here.

amples should provide help for users to create further analyses and visualizations, supplementary to the documentation. As with any free and open source project, contributions by users of RKorAPClient are very welcome.

5. Authorization and Authentication

One of the biggest challenges in making corpus data available is the various copyrights and licensing conditions that restrict third party access to the data. As mentioned in the introduction, the RKorAPClient can also be seen as part of a technical and organisational solution for typical licensing restrictions that aims to provide linguists with access to corpus data that is as flexible and useful as possible, without infringing on the interests of rights holders.

In DeReKo most corpus data require a license agreement signed by the end user and are therefore only accessible with authentication. Nevertheless, KorAP API allows non-authenticated users to still gain benefit from limited search results such as public metadata of licensed corpora, that can be freely disclosed, which in the case of DeReKo includes, for example, place of publication, source, text type, topic domain, or column (for newspaper articles) (Kupietz et al., 2019). Although the search results are limited to public metadata, this is quite sufficient to cover a wide range of research questions, like all kinds of frequency distribution analyses (see the RKorAPClient demos for examples). In cases where it is necessary to view the textual data as well, this can be done in an authenticated manner via the Web interface.[13]

5.1. Server-side Implementation

OAuth 2.0 (Hardt, 2012) is a protocol enabling authenticated users to give authorizations to external applications to access their private data or act on their behalf. User authentication and authorizations given to applications are represented by OAuth tokens. In KorAP, OAuth tokens are issued via Kalamar. KorAP allows applications having valid OAuth tokens to perform actions according to the authorization scopes associated with the tokens, such as searching and retrieving annotations. Since OAuth tokens represent user authentication, restricted corpus data including text snippets and annotations can be accessed using KorAP API calls via external applications, for instance for statistical analysis using the RKorAPClient.

5.2. RKorAPClient Implementation

The R client package supports both, unauthenticated and authenticated access. In the case of unauthenticated access (i.e., when no access token was pre-configured and the `KorAPConnection` is initialized without providing an **`accessToken`** parameter) subsequent queries will only return so-called *public* metadata fields, that are not affected by copyright or license restrictions.[14] For authenticated access, the KorAP connection needs to be initialized with a valid `accessToken` parameter:

```
kco <- new("KorAPConnection", accessToken="<token>")
```

The access token is then applied for all subsequent requests using the `KorAPConnection` instance `kco`.

By means of the method **`persistAccessToken`**, the access token can also be made persistent beyond the current session. It is then used automatically for each new connection until it is deleted using the **`clearAccessToken`** method. Access tokens are bound to the base request URL of the connection so that different tokens can be used to access different KorAP instances. In order to keep the tokens as private as possible, **`persistAccessToken`** stores them via the keyring package, that preferably uses the operating system's credential store.

6. Conclusion

API access to search and analysis functions of a corpus platform is an important mechanism to enable methods of analysis and post-processing that are not yet natively available (i.e., provided by the corpus platform itself) or are so specific that they are not expected to be ever made available

[13] As explained in Section 3.3., RKorAPClient provides a corresponding linking.

[14] This is partially handled already on the client side by not requesting known restricted fields in order to make sure not to provoke query rewrites, which is KorAP's mechanism for allowing only authorized access to restricted data (Bański et al., 2014).

natively. Client libraries are a good way to provide such additional capabilities that go beyond the natively provided functionality and allow the user to easily access and process the data without leaving their familiar development or analysis environment (e.g., programming language). In case of R-KorAPClient extra features are available for visualizing the processed data.

The aim of the provision of RKorAPClient as part of the KorAP project is to offer simple API access methods as building blocks for more complex tasks, as well as complex methods, which are expected to be provided natively by KorAP in the near future. In this way, modules using RKorAPClient can take advantage of future developments (i.e., native implementations of complex methods) without the need for any changes in their development or analysis environments, but with benefits especially in terms of performance.

In order to facilitate the integration into further development environments, implementations for other programming languages are planned, which will follow the model of RKorAPClient to offer comparable feature sets.

7. Bibliographical References

al-Wadi, D. (1994). *COSMAS – Ein Computersystem für den Zugriff auf Textkorpora. Version R.1.3-1. Benutzerhandbuch*. Institut für Deutsche Sprache, Mannheim. Mit einem Geleitwort von Prof. Dr. Gerhard Stickel.

Barbu Mititelu, V., Tufiş, D., and Irimia, E. (2018). The reference corpus of the contemporary romanian language (CoRoLa). In Calzolari, N., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC'18)*, pages 1235–1239, Miyazaki/Paris. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2014/pdf/360_Paper.pdf.

Bański, P., Diewald, N., Hanl, M., Kupietz, M., and Witt, A. (2014). Access Control by Query Rewriting: the Case of KorAP. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3817–3822, Reykjavik/Paris. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2014/pdf/743_Paper.pdf.

Bański, P., Fischer, P. M., Frick, E., Ketzan, E., Kupietz, M., Schnober, C., Schonefeld, O., and Witt, A. (2012). The New IDS Corpus Analysis Platform: Challenges and Prospects. In Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2905–2911, Istanbul/Paris. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2012/pdf/789_Paper.pdf.

Burns, P. (2011). *The R inferno*. Lulu.com.

Cristea, D., Diewald, N., Haja, G., Mărănduc, C., Barbu Mititelu, V., and Onofrei, M. (2019). How to find a shining needle in the haystack. Querying CoRoLa: solutions and perspectives. In Cosma, R. and Kupietz, M., editors, *On design, creation and use of of the Reference Corpus of Contemporary Romanian and its analysis tools. CoRoLa, KorAP, DRuKoLA and EuReCo*, volume 64(3) of *Revue Roumaine de Linguistique*, pages 279–292. Editura Academiei Române, Bucharest. https://www.lingv.ro/images/RRL%203%202019%2007-Cristea.pdf.

Diewald, N., Barbu Mititelu, V., and Kupietz, M. (2019). The KorAP user interface. Accessing CoRoLa via KorAP. In Cosma, R. and Kupietz, M., editors, *On design, creation and use of of the Reference Corpus of Contemporary Romanian and its analysis tools. CoRoLa, KorAP, DRuKoLA and EuReCo*, volume 64(3) of *Revue Roumaine de Linguistique*, pages 265–277. Editura Academiei Române, Bucharest. https://www.lingv.ro/images/RRL%203%202019%2006-%20Diewald.pdf.

Diewald, N., Hanl, M., Margaretha, E., Bingel, J., Kupietz, M., Banski, P., and Witt, A. (2016). KorAP architecture – Diving in the deep sea of corpus data. In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3586–3591, Portorož/Paris. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2016/pdf/243_Paper.pdf.

Fischer, P. M. and Lang, C. (2019). Ein Tool zur Visualisierung des Gebrauchs von Schreibvarianten. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*, pages 269–270, Erlangen. German Society for Computational Linguistics & Language Technology.

Hardt, D. (2012). RFC 6749: The OAuth 2.0 authorization framework. Specification. https://tools.ietf.org/html/rfc6749.

Kunst, J. (2019). highcharter: A wrapper for the "highcharts" library. https://cran.r-project.org/package=highcharter.

Kupietz, M., Belica, C., Keibel, H., and Witt, A. (2010). The German Reference Corpus DeReKo: A Primordial Sample for Linguistic Research. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 1848–1854, Valletta/Paris. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2010/pdf/414_Paper.pdf.

Kupietz, M., Cosma, R., Cristea, D., Diewald, N., Trawiński, B., Tufiş, D., Váradi, T., and Wöllstein, A. (forthcoming). Recent developments in the European Reference Corpus (EuReCo). In Granger, S. and Lefer, M.-A., editors, *Proceedings of the Using Corpora in Contrastive and Translation Studies*, Louvain-la-Neuve. Presses universitaires de Louvain.

Kupietz, M., Diewald, N., and Fankhauser, P. (2018a). How to get the computation near the data: Improving data accessibility to, and reusability of analysis functions in corpus query platforms. In Bański, P., Kupietz, M., Barbaresi, A., Biber, H., Breiteneder, E., Clematide, S., and Witt, A., editors, *Proceedings of the LREC 2018 Workshop "Challenges in the Management of Large Corpora" (CMLC-6)*, pages 20–25, Miyazaki/Paris. European Language Resources Association (ELRA). http://nbn-resolving.de/urn:nbn:de:bsz:mh39-75346.

Kupietz, M., Diewald, N., Hanl, M., and Margaretha, E. (2017). Möglichkeiten der Erforschung grammatischer Variation mithilfe von KorAP. In Konopka, M. and Wöllstein, A., editors, *Grammatische Variation. Empirische Zugänge und theoretische Modellierung*, pages 319–329. De Gruyter, Berlin. http://nbn-resolving.de/urn:nbn:de:bsz:mh39-59681.

Kupietz, M., Lüngen, H., Kamocki, P., and Witt, A. (2018b). The German Reference Corpus DeReKo: New Developments – New Opportunities. In Calzolari, N., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC'18)*, pages 4353–4360, Miyazaki/Paris. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2018/pdf/737.pdf.

Kupietz, M., Margaretha, E., Diewald, N., Lüngen, H., and Fankhauser, P. (2019). What's new in EuReCo? Interoperability, comparable corpora, licensing. In Bański, P., Barbaresi, A., Biber, H., Breiteneder, E., Clematide, S., Kupietz, M., Lüngen, H., and Iliadi, C., editors, *Proceedings of the Workshop "Challenges in the Management of Large Corpora" (CMLC-7)*, pages 33–39, Mannheim. Leibniz-Institut für Deutsche Sprache. http://nbn-resolving.de/urn:nbn:de:bsz:mh39-90261.

Meyer, D., Zeileis, A., and Hornik, K. (2006). The strucplot framework: visualizing multi-way contingency tables with vcd. *Journal of Statistical Software*, 17(3):1–48.

Müller, T., Schmid, H., and Schütze, H. (2013). Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA. Association for Computational Linguistics.

OASIS Standard (2013). searchRetrieve: Part 5. CQL: The Contextual Query Language Version 1.0. http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/os/part5-cql/searchRetrieve-v1.0-os-part5-cql.html.

Oravecz, C., Váradi, T., and Sass, B. (2014). The Hungarian gigaword corpus. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1719–1723, Reykjavik/Paris. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2014/pdf/681_Paper.pdf.

Przepiórkowski, A. (2004). The IPI PAN corpus: Preliminary version. Technical report, Institute of Computer Science, Polish Academy of Sciences, Warsaw.

Rat für deutsche Rechtschreibung, editor (2006). *Deutsche Rechtschreibung. Regeln und Wörterverzeichnis. Amtliche Regelung*. Narr Francke Attempto Verlag, Tübingen.

Rosenfeld, V. (2010). An implementation of the Annis 2 query language. Technical report, Humboldt-Universität zu Berlin, Berlin.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK. https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger1.pdf.

Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M., and Despouy, P. (2017). plotly: Create interactive web graphics via 'plotly.js'. *R package version*, 4(1):110.

Tidwell, J. (2006). *Designing Interfaces. Patterns for Interaction Design*. O'Reilly & Associates, Inc.

Wickham, H. (2016). *ggplot2: elegant graphics for data analysis*. Springer.

Wickham, H. and Grolemund, G. (2016). *R for data science: import, tidy, transform, visualize, and model data*. O'Reilly Media, Inc.

Wolfer, S. and Hansen-Morath, S. (2018). Visualisierung sprachlicher Daten mit R. In Bubenhofer, N. and Kupietz, M., editors, *Visualisierung sprachlicher Daten*. heuip. https://doi.org/10.17885/heiup.345.474.

Ziem, A. (2008). Universale Prägung und kulturelle Varianz. Überlegungen zu einem integralen Bedeutungsmodell im kognitiven Paradigma. *Zeitschrift für Literaturwissenschaft und Linguistik*, 38(3):185–198.

8. Language Resource References

Hungarian Academy of Sciences (2018). Hungarian National Corpus.

Leibniz-Institut für Deutsche Sprache (2019a). German Reference Corpus DeReKo. Deutsches Referenzkorpus, DeReKo-2019-I. PID: http://hdl.handle.net/10932/00-04BB-AF28-4A4A-2801-5.

Leibniz-Institut für Deutsche Sprache (2019b). Virtual Corpus W1 based on DeReKo-2019-I.

Leibniz-Institut für Deutsche Sprache (2020). German Reference Corpus DeReKo. Deutsches Referenzkorpus, DeReKo-2020-I. PID: http://hdl.handle.net/10932/00-04B6-B898-AD1A-8101-4.

Romanian Academy (2017). Reference Corpus of Contemporary Romanian Language. Romanian Academy, CoRoLa.