Published in: Bański, Piotr/Kupietz, Marc/Barbaresi, Adrien/Biber, Hanno/Breiteneder, Evelyn/Clematide, Simon/Witt, Andreas (eds.): Proceedings of the LREC 2018 Workshop “Challenges in the Management of Large Corpora (CMLC-6)” 07 May 2018 – Miyazaki, Japan. - Paris: European language resources association (ELRA), 2018. Pp. 20-25

How to Get the Computation Near the Data: Improving Data Accessibility to, and Reusability of Analysis Functions in Corpus Query Platforms

Marc Kupietz, Nils Diewald, Peter Fankhauser
Institut für Deutsche Sprache
R5, 6-13, D-68161 Mannheim
{kupietz, diewald, fankhauser}@ids-mannheim.de

Abstract

The paper discusses use cases and proposals to increase the flexibility and reusability of components for analysis and further processing of analysis results in corpus query platforms by providing standardized interfaces to access data at multiple levels.

Keywords: Corpus Query Platform, Interoperability, Reusability, Extensibility, Legal Issues

1. Introduction

Compared to other disciplines that deal with big research data, in linguistics it is on the one hand particularly important and on the other hand particularly difficult to equip research software with satisfactory functionalities for data analysis. It is important because the research data usually is not only too big to move, but also – due to IPR and license restrictions – often not allowed to move (Kupietz et al. 2010, 2014; Kamocki et al. 2016). For this reason, researchers who want to analyse corpora cannot simply download the data and apply their own software tools. Providing generally satisfactory functionalities in central corpus analysis platforms, however, has proven difficult. One reason for this is that the question, how language data should be analysed, depends very much on the research aims, and is itself a key research field with rapid development. As a consequence, provided functionalities will typically not cover all areas of interest and will be outdated quickly. Finding solutions for this dilemma is even more important as a quick and general availability of methods for analysis and for the visualization of analysis results, and accordingly a certain nudge towards methodological canonicalization is likely to foster scientific progress in disciplines that deal with language data.

In this paper we will review current approaches (including our own approach with KorAP) to interoperable language analysis and discuss their limitations. We then describe typical analysis use cases from which we derive the main data structures needed for their support. On this basis, we outline a basic API, and discuss the resulting data access requirements.

2. Previous work

2.1. CLARIN

A lot of valuable and groundbreaking work with respect to the reusability of language resources and technology has been done within the European Research Infrastructure for Language Resources and Technology CLARIN. Apart from corpus encoding standards, and best practices, particularly relevant for the subject of this paper are the experiences with the web-service orchestration and tool-chaining platform for corpus processing WebLicht (Hinrichs, Hinrichs and Zastrow 2010). WebLicht itself, however, is based on a distributed network of web-services between which data is transferred while analysis functions are immobile. Thus the WebLicht approach does not address license and IPR restriction problems, typical in corpus linguistics.

In general, the task addressed in this paper will complement the work done in CLARIN, in the sense that CLARIN work focuses on commonly used base infrastructures and existing resources and technologies, while the task here is focussed on a very specific class of use cases and applications.

2.2. KorAP

Increasing data reusability and accessibility for different applications using different methods was one of the key targets of KorAP, the corpus analysis platform developed at the IDS starting in 2011 (Kupietz et al. 2014). From the beginning of the KorAP project, it has been clear that the core developers at the IDS would not be able to develop and maintain all desired functionalities to satisfy all potential users.

At an early stage of the KorAP's design phase (Bański et al. 2012, p. 2906), the intended approach to solve the problem was to roughly follow Jim Gray's (2004) principle, *if the data cannot move put the computation near the data*, by providing a mobile code sandbox where users can run their own “KorAPp” code with controlled output in order to meet license restrictions (Kupietz et al. 2010). However, due to high expected development costs of such a sandbox, its maintenance and the required hardware, this approach was only pursued in a manual way: DeReKo samples were sent to interested users to let them adapt their software to the DeReKo encoding format, the software then was applied manually on IDS servers, and the controlled output was sent back to the users (Kupietz et al. 2014).

Proposed levels of access

To simplify this work as much as reasonable, since 2014 (Bański et al.) KorAP follows an alternative multi-level approach for making the data accessible to user code depending on the respective task, with the following levels of access:

- **Corpus**: Corpus-level access is typically most suitable for tasks that (a) require whole texts as input and/or (b) need to add new annotations to the corpus data. A typical example for such a task is topic domain or text type classification. It can be carried out by means of (manually applied) mobile code as described above.

- **Multiple (Backend) Instances:** This level, or rather approach, is ideally applicable for tasks relying on multiple, physically separated corpora and where identical functionalities are expected as e. g. for contrastive studies on comparable corpora (Cosma et al. 2016, Kupietz et al. 2017). In addition, however, this approach can also be a complementary alternative to standardized data access. For example if a corpus query tool A is used, but functionalities of tool B are required, in some cases the easiest solution can be to convert the corpus data to the format required by tool B and run an instance of tool B in addition to tool A.
- **Web-API**: This level seems ideally applicable for tasks that require search results. In the case of KorAP, the interface is specified by the Kustvakt REST-API and the KoralQuery protocol (Bingel and Diewald 2015); optionally executed close to the data to avoid network latencies ("fast lane").
- **Open Source**: This level is ideal for tasks that require or suggest extensions to core functionalities. In the case of KorAP, such extensions can be proposed for review and submitted via Gerrit[1 2]. For functionalities for which alternatives are specifically welcomed, an interface definition (in the Java sense) could streamline this extension process.

Currently, the access at these different levels is, however, mostly KorAP-specific and not standardized. Thus, the proposed levels of data-access primarily serve to complement and to facilitate further steps of canonicalization and standardization.

2.3. Other corpus query tools

Probably all existing corpus query tools support mechanisms for exporting query results in different formats to allow further processing. The corpus workbench (Evert and Hardy 2011), for example, provides a tabulate command to aggregate query results into tables that can then be used for further statistical analysis.

A corpus query tool that goes very far with providing interfaces for data access is the CorpusExplorer[3] (Rüdiger 2018) which offers an extensible large variety of export formats and an SDK to allow users to develop their own functions for analysis and further processing.

Nederlab (Brugman et al. 2016) provides an R based visualization service (Komen 2015) as a separate service component to further process search results, especially for visualizations, and is meant to support even user provided custom R modules.

3. Use cases

This list of example use cases for exchangeable analysis modules is not intended to be complete. It rather reflects our own interests, the experiences with the DeReKo user community, and mainly serves as an overview of possible application classes and their requirements.

3.1. Collocation Analysis

An obvious and simple use case for interoperable analysis modules is collocation analysis (Sinclair 1991), which is offered in nearly every corpus platform and one of the most widely used and most often varied methodologies in corpus linguistics.

The standard case: cohesion of word pairs

In the standard case, collocation or co-occurrence analysis measures the association between co-occurring words. I. e. it assigns an association score to word pairs of the vocabulary observed in a corpus based on their respective total frequencies and their co-occurrence frequency within a certain context-window, e. g. [-5, 5] around the target word, so that high scores indicate strong attraction, low scores indicate weak attraction, and scores can be used to rank the word pairs according to the strength of their attraction (Evert 2009).

The results of collocation analysis can be used for finding recurrent syntagmatic patterns and, by means of comparing vectors of association scores (*collocation profiles*) for finding paradigmatic relationships between words (Perkuhn 2007). For both scenarios exchangeable analysis functions and interfaces for further processing or visualizing the results would be desirable.

Higher-order collocation analysis

In higher-order collocation analysis (Keibel and Belica 2007) the analysis is applied recursively by taking the cohesive pairs found in one step as a single node and using some form of the found concordances as the corpus in the subsequent step.

It is already more difficult to define a sufficiently general standardized API for this simple extension, however, it might be a good example of functions that are more easily standardizable on an API level by using callbacks.

3.2. Corpus Comparison

The general goal of corpus comparison (Kilgariff 2001) is to analyse language use depending on text-external variables, such as mode (oral vs. written), genre, register, discipline, or time, in order to understand the correlation between text-internal features with text-external variables. Analysis techniques comprise multivariate analysis (e. g. Biber 1993), cluster analysis, and classification (Teich and Fankhauser 2010) together with feature selection and ranking (Fankhauser et al. 2014, Teich et al. 2014).

In terms of data structures, the typical workflow for corpus comparison is as follows. For feature selection, a query on the corpus is used to select a virtual subcorpus and features of interest. The query result consists of sequences of words, or more generally sequences of features (such as part-of-speech *n*-grams), from which bags of words can be derived. Crucial for this kind of analysis is that the query results are contextualized with the text-external variables.

[1] KorAP's Gerrit: `https://korap.ids-mannheim.de/gerrit/`

[2] Gerrit Code Review in general: `https://www.gerritcodereview.com/`

[3] `https://notes.jan-oliver-ruediger.de/software/corpusexplorer-overview/`

3.3. Provenance of analysis results: Linking back to the concordances

A feature that is generally desirable across many if not all use cases is the possibility to link back from aggregated representations of query or analysis results to the corpus texts that were the basis of the analysis. A typical example would be to allow users to click on tokens displayed in map-visualizations of distributional-semantic neighbourhoods (Keibel and Belica 2007, Fankhauser and Kupietz 2017), or in frequency graphs (Kupietz et al. 2017b: 327) in order to execute a query that shows the corresponding or underlying concordances. Such a feature is highly desirable because in typical exploratory workflows (Tukey 1977) the results of further processing of corpus analyses do not have the function to thoroughly display observations but rather to illicit the abduction of new hypotheses that need to be verified on the basis of the observed data (Keibel and Kupietz 2009, Jockers 2013, Perkuhn and Kupietz 2018: 87-88).

4. Data modelling

As exemplified in the use cases above, the two main data structures for text analysis are sequences of words and bags of words.

Sequences of words maintain the actual order of words in context of their use up to a certain length. Thereby, sequences comprise concordances, word *n*-grams, and, when adorned with frequencies, *n*-gram language models.

Bags of words disregard the order of words, but often represent larger contexts, such as documents or entire subcorpora, by means of vectors over the entire vocabulary of a corpus.

Word co-occurrences constitute an interesting case in between. On the one hand, they can be modelled by means of bigram sequences w_1w_2, on the other hand, as bags of words indexed by w_1 or w_2.

Both, sequences of words and bags of words, should be equipped with a (not necessarily unique) context identifier, which allows to associate them with text external variables, such as metadata, about the document or the subcorpus. Likewise, the words themselves can be equipped with identifiers, in order to associate word or position specific annotations, such as lemma or part-of-speech, and more generally link back to the corpus text.

5. Interfaces

To make analysis modules in corpus query systems reusable, standardized interfaces are required. As all use cases focus on very large corpus data and the introduced perspective is on data that requires restricted access due to license and IPR restrictions, a scenario of standardized web-service APIs seems to be obvious, although interface definitions could be adapted for programming library interfaces as well. The interface definition can be separated in three parts: the request protocol, the response format, and, in case the analysis results should be represented by the corpus query system or processed further, the analysis format.

Request Protocol

The request protocol specifies endpoints and parameters for data access. Regarding the presented example use cases, these are at least

- **Query endpoint**: Requires corpus definition (or document identifier), query definition (or positional identifiers), context definition (in characters, words, annotations, etc.), metadata fields to retrieve, annotations in the match to retrieve
- **Statistical endpoint**: Calculation of, e.g., numbers of tokens or occurrences in a defined corpus
- **Grouping endpoint**: Grouping of matches with frequency information

The request protocol probably requires batch processing for large request sets and a paging/cursor mechanism for large result sets. Existing APIs to get inspiration from include OpenSearch[4], SRU[5], and PortableContacts[6].

Response Format

The response format represents the accessible corpus data in a machine readable format, preferable in JSON (or JSON-LD, following recommendations from ISO TC37 SC4 WG1-EP) or XML for further processing. Existing formats to get inspiration from include RSS, Atom, and ActivityStreams[7].

Analysis Format

The results of the presented analysis methods can be serialized as data tables, therefore they may use a CSV format for further processing, or a serialization to JSON or XML. Existing APIs to get inspiration from include the Web Language Model API[8]. For visual integration in user interfaces, the data may be passed as image data or presented in (sandboxed) iframes. Existing APIs to get inspiration from include OpenSocial[9].

6. Data Access Requirements and Current Implementations

The use cases introduced above require means of access to corpus data that are not necessarily in the focus of corpus query engines, which are typically optimized for structured queries (formulated in corpus query languages like CQP, Christ 1994) on structured textual data, rather than for providing data structures suitable for further analysis. In addition, they may provide functionalities to define subcorpora or restrict access to the underlying corpus, because full access is limited due to legal constraints. To meet these requirements, most corpus query engines either rely on uniformly layouted documents (e. g. searching plain XML files using XQuery or XPath) or indexed corpus data. As operations on non-indexed data can computationally be expensive for very large corpora, indexed representations are prefered for most use cases involving data analysis.

Recent developments in corpus query engines focus on inverted indices, as used by BlackLab[10], MTAS[11] or KorAP

[4] `http://www.opensearch.org/`

[5] `https://www.loc.gov/standards/sru/`

[6] `http://portablecontacts.net/`

[7] `http://activitystrea.ms/`

[8] `https://azure.microsoft.com/de-de/services/cognitive-services/web-language-model/`

[9] `https://github.com/opensocial`

[10] `http://inl.github.io/BlackLab/`

[11] `https://meertensinstituut.github.io/mtas/`

(Diewald and Margaretha 2016) - although alternative approaches have proven to be useful as well (e. g. relational database models in ANNIS, Zeldes et. al 2009; or graph databases in graphANNIS, Krause et al. 2016). While inverted indices perform well on the task of structural queries (by providing for fast retrieval of textual units using a dictionary of surface terms, annotations, or metadata fields), they are not well suited for fast retrieval of contextual information necessary for the required data modelling for data analyses. The match information of a search, retrieved from the postings lists of an inverted index contain, at minimum, a document identifier (that can be used to recreate context on the document level) and optionally positional information (that can be used to recreate context of occurrences on the token or character level). However, the recreation of contexts (for example to retrieve all meta information of a document or to generate snippets) normally requires post-processing steps, involving additional data structures (see Manning et al. 2008, p. 158). Implementations like Apache Lucene[12] (the inverted index architecture behind BlackLab, MTAS, and KorAP) can provide fast access to stored metadata fields as well as primary data, that can be, in conjunction with positional information, be used to recreate textual context for a match. These additional data representations however have the disadvantage of introducing redundancy. Instead of a raw primary data string, a stored field may contain an annotation enriched representation as well (cf. COSMAS II, Bodmer 1996), introducing even more redundancy. This is useful to return context including annotational markup (see the use case for corpus comparison) or to limit the context according to annotation boundaries, like sentences. More elaborate engines make use of an additional forward index (see for example BlackLab[13], or prototype versions of KorAP[14]) to provide fast access to the raw data by positional information.

Additional data access is required for the proposed use cases regarding corpus statistics, for example to retrieve the number of tokens in a specified subcorpus, or the number of occurrences of context tokens in a subcorpus. Some of these information can be precalculated and stored in document associated meta fields, but the corpus query engine needs to provide methods for performant data aggregation as well.

To make data analysis components exchangeable, corpus query engines will be required to not only provide fast search capabilities, but also grant fast access to all underlying corpus data (primary data, annotation data, and metadata), based on document and positional information, while still respecting IPR restrictions.

7. Preliminary Conclusions

Improving the interoperability and extensibility of analysis and further processing functions in different corpus query engines seems desirable and feasible. However, the development and maintenance costs for supporting more sophisticated applications via canonical APIs seem – given the extensive experiences within the CLARIN project, the heterogeneity of corpus query tools and even the limited scope of use cases given in this paper – quite high. We, thus, recommend to start the canonicalization with functions that are of strong common interest as well as easily convergable. To this end, we presented some widely used applications for corpus analysis, and identified the main data structures to support them.

As flanking measures, we recommend to follow and to extend a multi-level approach as sketched above, already at a not (fully) standardized stage. This means for example to support standard corpus encoding and annotation formats, so that corpora and corpus analysis tools can be added and exchanged with manageable efforts and to support the open source level by encouraging the extension of corpus query systems with new analysis functions on the part of external users and developers. As always, standardization makes sense upto a point where the total costs for implementing and maintaining the required generality exceeds the total costs of achieving the desired results at the relevant places and maintaining their required reproducibility without a standard. Instead of guessing, where this break-even point will be, it seems reasonable to start with some safe candidates while not neglecting other (promising) ones.

8. References

Bański, P., Fischer, P. M., Frick, E., Ketzan, E., Kupietz, M., Schnober, C., Schonefeld, O., Witt, A. (2012). The New IDS Corpus Analysis Platform: Challenges and Prospects. In: Calzolari, N. et al. (eds.): Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12). Istanbul, Turkey, May 2012. European Language Resources Association (ELRA), 2012: 2905-2911.

Bański, P., Diewald, N., Hanl, M., Kupietz, M. and A. Witt (2014). Access Control by Query Rewriting: the Case of KorAP. In: *Proceedings of the 9th conference on the Language Resources and Evaluation Conference* (LREC 2014), European Language Resources Association (ELRA), Reykjavik, Iceland, May 2014: 3817-3822.

Biber, D. (1993). The Multi-Dimensional Approach to Linguistic Analyses of Genre Variation: An Overview of Methodology and Findings, Computers and the Humanities, 26: 331-345.

Bingel, J. and Diewald, N. (2015). KoralQuery – a General Corpus Query Protocol. In: Proceedings of the Workshop on Innovative Corpus Query and Visualization Tools at NODALIDA 2015, Vilnius, Lithuania, May 11-13, pp. 1-5.

Bodmer, F. (1996). Aspekte der Abfragekomponente von COSMAS-II. LDV-INFO. Informationsschrift der Arbeitsstelle Linguistische Datenverarbeitung, 8:112–122.

Brugman, H., Reynaert, M., van der Sijs, N., van Stipriaan, R., Sang, E. T. K., and van den Bosch, A. (2016). Nederlab: Towards a Single Portal and Research Environment for Diachronic Dutch Text Corpora. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Slovenia. pp.1277-1281.

Christ, O. (1994). A modular and flexible architecture for an integrated corpus query system. In Proceedings of COM-

[12] http://lucene.apache.org/core/

[13] http://inl.github.io/BlackLab/file-formats.html

[14] See https://github.com/KorAP/Krawfish-prototype

PLEX'94, 3rd Conference on Computational Lexicography and text Research, Budapest, Hungary. pp. 23-32.

Cosma, R., Cristea, D., Kupietz, M., Tufiş, D., Witt, A. (2016). DRuKoLA – Towards Contrastive German-Romanian Research based on Comparable Corpora. In: Bański, P. et al. (eds.):4th Workshop on Challenges in the Management of Large Corpora. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Slowenien. Paris: European Language Resources Association (ELRA), 2016: 28-32.

Diewald, N. and Margaretha, E. (2016). [200E?]Krill: KorAP search and analysis engine. In: Journal for Language Technology and Computational Linguistics (JLCL), 31 (1). 63-80.

Evert, S. (2009). Corpora and collocations. In Lüdeling, A. and Kytö, M. (eds.), Corpus Linguistics. An International Handbook. Mouton de Gruyter, Berlin.

Evert, S. and Hardie, A. (2011). Twenty-first century Corpus Workbench: Updating a query architecture for the new millenium. In Proceedings of the Corpus Linguistics 2011 conference, University of Birmingham, UK.

Fankhauser, P., Knappen, J. and Teich, E. (2014). Exploring and Visualizing Variation in Language Resources. Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). Reykjavik, Iceland

Fankhauser, P. and Kupietz, M. (2017). Visualizing Language Change in a Corpus of Contemporary German. In: Proceedings of the 9th International Corpus Linguistics Conference. Birmingham: University of Birmingham, 2017.

Gray, J. (2004). Distributed Computing Economics. In: Herbert A., Jones K.S. (eds) Computer Systems. Monographs in Computer Science. Springer, New York, NY

Hinrichs, E., Hinrichs, M, and Zastrow, T. (2010). WebLicht: Web-Based LRT Services for German. In: *Proceedings of the ACL 2010 System Demonstrations*. pp. 25–29.

Jockers, M. L. (2013). Macroanalysis: Digital Methods and Literary History. Champaign, IL: University of Illinois Press.

Kamocki, P., Kinder-Kurlanda, K., Kupietz, M. (2016). One Does Not Simply Share Data. Organisational and Technical Remedies to Legal Constraints in Research Data Sharing -- building bridges between Digital Humanities and the Social Sciences. In: Witt, A., Kamocki, P. (2016). When DH Meets Law: Problems, Solutions, Perspectives. In Digital Humanities 2016: Conference Abstracts. Jagiellonian University & Pedagogical University, Kraków, pp. 104-108.

Keibel, H. and Belica, C. (2007). CCDB: A Corpus-Linguistic Research and Development Workbench. In: Proceedings of the 4th Corpus Linguistics Conference (CL 2007). Birmingham: University of Birmingham.

Keibel, H. and Kupietz, M. (2009): Approaching grammar: Towards an empirical linguistic research programme. In: Minegishi, Makoto/Kawaguchi, Yuji (Eds.): Working Papers in Corpus-based Linguistics and Language Education, No. 3. Tokyo: Tokyo University of Foreign Studies (TUFS), 2009. pp. 61-76.

Kilgarriff, A. (2001). Comparing Corpora. International Journal of Corpus Linguistics, 6(1): 1-37.

Komen, E. R. (2015). An insider's guide to the Nederlab R visualization webserver. Technical report, Nederlab.

Krause, T., Leser, U., Lüdeling, A. (2016). graphANNIS: A Fast Query Engine for Deeply Annotated Linguistic Corpora. In: Journal for Language Technology and Computational Linguistics (JLCL), 31 (1). 1-25.

Kupietz, M., Belica, C., Keibel, H. and Witt, A. (2010). The German Reference Corpus DeReKo: A primordial sample for linguistic research. In: Calzolari, N. et al. (eds.): *Proceedings of LREC 2010*. 1848-1854.

Kupietz, M., Lüngen, H., Bański, P. and Belica, C. (2014). Maximizing the Potential of Very Large Corpora. In: Kupietz, M., Biber, H., Lüngen, H., Bański, P., Breiteneder, E., Mörth, K., Witt, A., Takhsha, J. (eds.): *Proceedings of the LREC-2014*-Workshop Challenges in the Management of Large Corpora (CMLC2). Reykjavik: ELRA, 1–6.

Kupietz, M., Witt, A., Bański, P., Tufiş, D., Cristea, D., Váradi, T. (2017). EuReCo – Joining Forces for a European Reference Corpus as a sustainable base for cross-linguistic research. In: Bański, P. et al. (eds.): Proceedings of the Workshop on Challenges in the Management of Large Corpora and Big Data and Natural Language Processing (CMLC-5+BigNLP) 2017. Birmingham, 24 July 2017. Mannheim: Institut für Deutsche Sprache, 2017: 15-19.

Kupietz, M., Diewald, N., Hanl, M., Margaretha, E. (2017b). Möglichkeiten der Erforschung grammatischer Variation mithilfe von KorAP. In: Konopka, M., Wöllstein, A. (eds.): Grammatische Variation. Empirische Zugänge und theoretische Modellierung. Berlin/Boston: de Gruyter.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). Introduction to Information Retrieval, Cambridge University Press.

Perkuhn, R. (2007). Systematic Exploration of Collocation Profiles. In: Proceedings of the 4th Corpus Linguistics Conference (CL 2007), Birmingham. University of Birmingham.

Perkuhn, R. and Kupietz, M. (2018). Visualisierung als aufmerksamkeitsleitendes Instrument bei der Analyse sehr großer Korpora. In: Bubenhofer, N. and Kupietz, M. (eds.): Visualisierung sprachlicher Daten: Visual Linguistics – Praxis – Tools. Heidelberg: Heidelberg University Publishing.

Rüdiger, J. O. (2018). CorpusExplorer v2.0 – Visualisierung prozessorientiert gestalten. In: Bubenhofer, N. and Kupietz, M. (eds.): Visualisierung sprachlicher Daten: Visual Linguistics – Praxis – Tools. Heidelberg: Heidelberg University Publishing.

Sinclair, J. (1991). Corpus, Concordance, Collocation. Oxford: Oxford University Press.

Teich, E., Fankhauser P. (2010). Exploring a Scientific Corpus Using Datamining. In: Gries, S. T., Wulff, S. and Davies, M. (eds.): Corpus-linguistic applications. Current studies, new directions. Amsterdam/New York, NY, 2010, VI, Rodopi, pp. 233-246

Teich, E., Degaetano-Ortlieb, S., Fankhauser, P., Kermes, H., and Lapshinova-Koltunski, E. (2014). The Linguistic Construal of Disciplinarity: A Data Mining Approach Using Register Features. Journal of the Association for Information Science and Technology (JASIST).

Tukey, J. W. (1977). Exploratory Data Analysis. Pearson.

Zeldes, A., Ritz, J., Lüdeling, A., and Chiarcos, C. (2009). ANNIS: A search tool for multi-layer annotated corpora. In Mahlberg, M., González-Díaz, V., and Smith, C., editors, Proceedings of the Corpus Linguistics 2009 Conference, Liverpool, UK.