Corpus Query Lingua Franca (CQLF)

Piotr Bański[1], **Elena Frick**[1], **Andreas Witt**[1, 2]

[1]Institute for the German Language (IDS Mannheim)
[2] Heidelberg University

Email: {banski|frick|witt}@ids-mannheim.de

Abstract

The present paper describes Corpus Query Lingua Franca (ISO CQLF), a specification designed at ISO Technical Committee 37 Subcommittee 4 "Language resource management" for the purpose of facilitating the comparison of properties of corpus query languages. We overview the motivation for this endeavour and present its aims and its general architecture. CQLF is intended as a multi-part specification; here, we concentrate on the basic metamodel that provides a frame that the other parts fit in.

Keywords: text corpora, query languages, ISO standards

1. Introduction

The present paper describes Corpus Query Lingua Franca (ISO CQLF), a specification designed at ISO Technical Committee 37 Subcommittee 4 "Language resource management" for the purpose of facilitating the comparison of properties of corpus query languages. We overview the motivation for this endeavour and present its aims and its general architecture. CQLF is intended as a multi-part specification; here, we concentrate on the basic metamodel that provides a frame that the other parts fit in.

1.1. Aims and Motivation

A lot of effort has been devoted to developing various corpus query languages (cQLs) in order to adequately satisfy information needs of corpus linguists and lexicographers. In practice, this large amount of different cQLs (see e.g. Clematide 2015) poses a serious challenge for researchers seeking to maximally exploit the advantages offered by contemporary language resources; this is true especially for those who do not have a background in computational linguistics and programming. Apart from learnability problems concerning some cQLs, the real problem is that a number of linguistic resources are accessible through drastically different corpus management systems which provide their own cQLs. Thus, mastering one or two cQLs is not enough to benefit from the entire palette of existing corpus linguistic software packages.

Apart from that, as the respective fields develop, information needs grow and it may well happen that after having invested significant time and effort into learning and adopting a new cQL, a researcher will come to realise that the given cQL provides no way to further elaborate on the existing queries, and that another cQL has to be used.

There exists a need for interoperability across corpus query systems, realized among others by abstracting away from individual cQLs and seeing how far their queries can be compared. The issue of general interoperability of language resources and applications has been addressed at various levels of abstraction by such initiatives as e.g. CLARIN "Federated Content Search", ISOCat (now in redesign), or WSEP (Ide, 2013). With the purpose of enabling interoperability of corpus query tools (or at least gauging the extent to which such interoperability is possible), we have initiated the development of a standard for the presentation of features of corpus QLs, called Corpus Query Lingua Franca (CQLF), within ISO TC37 SC4 WG6. Currently at the Committee Draft stage, ISO CQLF (ISO/CD 24623-1) is the first standardization activity in the domain of linguistic QLs.[1]

The goal of CQLF is to provide both a metamodel that would serve as a target space on which individual QLs can be located with respect to their properties, and a set of well-defined properties which could be applied as a measure of compatibility and interoperability between cQLs. CQLF is not expected to serve as an interlingua and provide the technical basis for query translation *from* one QL *into* another. Such initiatives raise a host of problems, ranging from low-level technical descriptions (e.g. the inability to preserve information when translating between regular expressions on the one hand and wildcards on the other) to issues of epistemology ("Does the result of the query reformulated in cQL2 address exactly the information need expressed in cQL1?"). The immediate goal of CQLF is therefore more modest: to serve as the target space within which QLs can be located with respect to their basic properties. It can thus also serve as a measure of compatibility and interoperability, but without

[1] See (Herzog, 2015) for a succinct description of the newest developments within TC34 SC4.

the added claim to provide QL-to-QL mappings. A robust CQLF-based mapping system in action (with the epistemological burden appropriately controlled) would be ideal in the long run, but it makes a lot of sense to start smaller.

CQLF is also expected to be useful to corpus linguists and Digital Humanists at large for the purpose of identifying the QL suitable for the corpus linguistic tasks that they are faced with.

1.2. Background

The background of CQLF has been framed by three sources that we describe in turn.

A *statement of existing demand* has been provided by Martin Mueller's report for the Mellon Foundation "Towards a digital carrel: A report about corpus query tools" (2010) providing the general idea and the name of Corpus Query *Lingua Franca*. As Mueller noted in his summary, "it seems worth doing" to put "together an ISO proposal for a corpus query lingua franca that would allow different query engines to interoperate" and to make "recommendations for representing queries. If it leads to a standard, it will make development easier. If it does not lead to a standard it may at least help articulate the points where interoperability becomes difficult or breaks down" (Mueller, 2010). While the present proposal is crucially not a proposal for a single query language "to rule them all", we consider it an expression of what is feasible to achieve in this area, in terms of facilitating interoperability and highlighting the points where interoperability should not be expected.

Subcommittee 4 of the ISO Technical Committee 37 (TC37 SC4), where CQLF was successfully accepted as a New Work Item and gradually developed into Committee Draft provides the *institutional background* for this initiative. A group of international experts have participated in the standardization process contributing to the creation of the present proposal.

The *practical background* for experimentation on implementing CQLF has been supplied by the Leibniz-Association-funded project "Korpusanalyseplattform der nächsten Generation" at the Institute for the German Language in Mannheim (IDS Mannheim) where a new corpus analysis platform KorAP has been developed (cf. Bański et al., 2012; Bański et al., 2013, Diewald et al., 2016).

At an early stage of the development of KorAP, a sizeable collection of use cases for querying corpora was compiled, and a number of cQLs were evaluated against it. That has provided a view on limitations and strengths of existing cQLs, and helped to identify the differences between them. The result of this work was one of the sources used to bootstrap the design of CQLF (see also Frick et al., 2012). Furthermore, part of the KorAP project focused on integrating CQLF concepts into a query management system, by providing a facility to translate the compatible corpus queries in various cQL into a common protocol, uniformly processed by the search backend. This in fact yielded the first reference implementation for CQLF (for more details see Bingel and Diewald, 2015).

1.3. Architecture of the Standard

CQLF is intended to eventually consist of three parts:

- Part 1, "Metamodel", overviewed here, provides the specification of the abstract CQLF metamodel and circumscribes the outer limits of QL compatibility.
- Part 2, "CQLF Ontology: Single-stream Architectures", describes the set of properties allowing to classify query languages within CQLF; this part has been implemented as an OWL ontology and is currently at the alpha stage.
- Part 3 "CQLF Ontology: Multi-Stream Architectures" is intended to address the tasks of querying multiple primary data streams, for example in multi-modal corpora and parallel corpora.

2. Components of the CQLF Metamodel

CQLF is designed to be a modular construction with several components. Each component is characterised with respect to some aspect of the data models describing corpus objects that are the target or context of queries. A schematic view of the components of a CQLF metamodel is presented in Figure 1 below. The top-level components are referred to as CQLF Classes and correspond to the major division into data models built upon a single data stream vs. those which use more than one data stream (be it binary or text-based), in parallel. The present paper focuses on the Single-stream class (to be introduced below), which consists of three CQLF Levels that correspond to the major kinds of data organization in linguistic corpora.

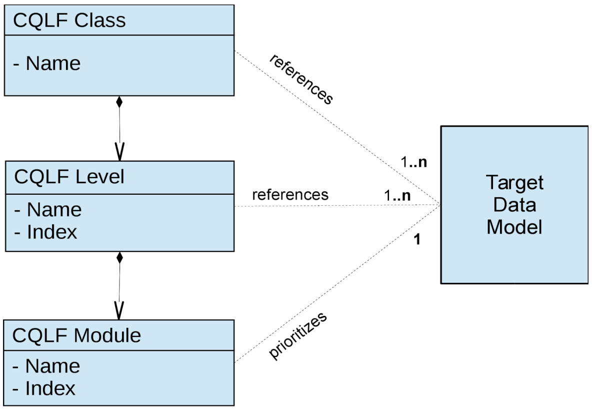

Figure 1: CQLF metamodel

The *index* variables of CQLF Level and Module are convenience details that allow for easier reference. The identification of levels and modules, while grounded in formal properties of data models and in functional characteristics of query languages, is also partially utilitarian, where it neglects some distinctions that could otherwise be recognised, in order to provide a simpler mechanism for determining conformance with the overall model and for stating the most important similarities and differences between cQLs.

The relationships sketched in Figure 1 together with the basic divisions made on the basis of the existing cQL, yield a basic taxonomy of cQLs represented in the diagram in Figure 2 below.

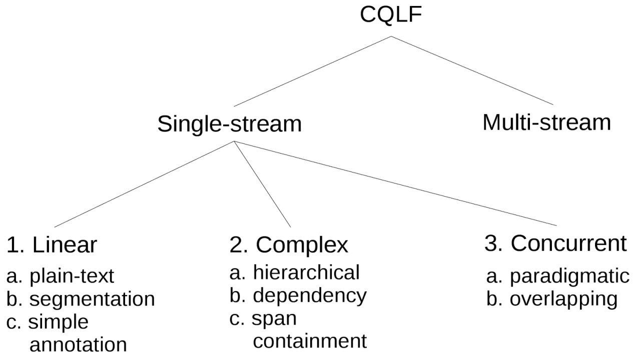

Figure 2: CQLF components

Within the Single-stream class, each consecutive level introduces a more complex search dimension. The level system is based on the distinction between different major types of data organization and consequently different types of annotations (modelled by various parts of the ISO LAF family of standards), as well as queries that correspond to them. Below, we look at each level in turn.

Level 1 (Linear) addresses, in any combination, plain-text search, search in segmented data (segmentation annotations) as well as search in simple annotations. What is relevant for this level is that annotations (if present) form a single layer of objects that exhaustively or partially describe the primary data stream. Tokenization is treated as a privileged level of segmental annotation (see (Odebrecht et al., submitted) for arguments for this point of view).

Level 2 (Complex) consists of three modules. The Hierarchical Module concerns tree-based representations (primarily used for phrase-structure description). The Dependency Module focuses on the identification of relationships in which words function as nodes. The two modules can be composed into a mixed representation, with dependency relationships imposed on phrase structure. The third module of this level concerns simplified hierarchical relationships encoded as span containment. This can mimic phrase structure (with extra devices for resolving cases of mutual containment, from which the hierarchy may not be derived) and is often non-recursive (as illustrated by e.g. CQP, cf. (Evert, 2010)).

By definition, hierarchical and dependency annotations involve spans (anonymous or with simple annotations), either as terminal leaves in trees, or as objects over which relations are defined. Similarly, in order to create representations in which annotations of the same data stream are aligned (or contrasted), there must exist some kind of span (even if sentence- or paragraph-sized) that is a member of the relevant relation. Levels 2 (Complex) and 3 (Concurrent) therefore assume the existence of segmental annotations (and by a minimal extension simple annotations), and additionally allow for plain-text search.

Level 3 (Concurrent) consists of two modules: Paradigmatic, responsible for prototypical cases where different annotation layers provide data packages competing for the same location, and Overlapping, which addresses concurrent annotations that are built upon spans which differ in their start and/or end offsets.

Note that by assuming that concurrent annotations involve a single underlying data stream, the possibility of representing alignments by means of ordered sets of offset anchors (because each case of alignment is measured at a single offset in the single underlying data stream) is effectively eliminated, and reliance must therefore be placed on a subset of a simple span algebra, with values restricted to "partial/complete/no overlap". This is different from cases where multiple streams of primary data are aligned, and where ordered pairs or n-tuples of offsets, pointing into each of the parallel data streams, must carry important information. Such cases are not in the scope of the present specification and are the subject of CQLF Part 3, "Multi-Stream Architectures".

The target data models are, minimally:

- For Level 1: ISO LAF (ISO 24612:2012), ISO MAF (ISO 24611:2012);
- For Level 2: ISO LAF, ISO SynAF-Metamodel (ISO 24615-1:2014);
- For Level 3: ISO LAF and other complex models.[2]

The above requirements are stated as the baseline for assessing the position of the given cQL within the matrix proposed here. Since CQLF is meant to be maximally inclusive, the references to ISO-defined data models are provided as hints rather than preconditions.

Note that in order to keep the number of levels small, some concessions have been made: plain-text search has been classified together with search within segments (whether token-level or larger) due to the fact that some matching mechanisms and syntactic operators are shared between the two. Similarly with search in simple annotations. In the same vein, hierarchical and dependency-based description again share many mechanisms (with mixed models well attested: for example, the Tiger XML (König et al., 2003) model is hierarchy-based, with "secondary edges" introducing dependencies between constituents; etc.). Containment-based relationships are quasi-hierarchical (hierarchy can in most cases be derived from containment).

Both Level 2 (Linear) and Level 3 (Concurrent) depend on simple annotations (alternatively, on segmental annotations), hence on Level 1. Level 3 may, but does not need to, build on Level 2. These interdependencies are illustrated in Figure 3.

[2] Note that ISO LAF does not provide recommendations for putting together concurrent annotations other than by creating a super-graph over the graphs describing individual stand-off annotations.

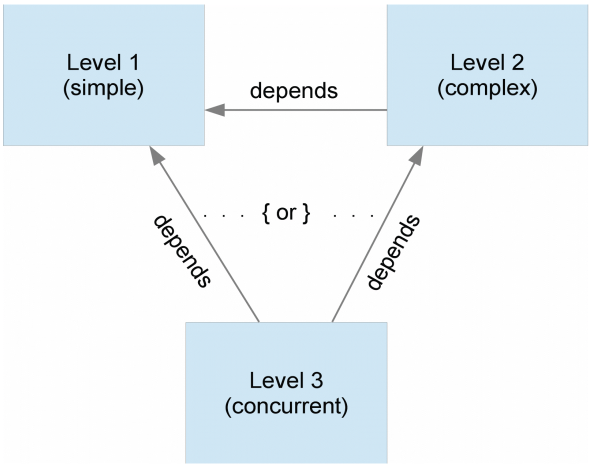

Figure 3: Dependencies among CQLF Levels. Note that "{or}" is an OCL (Object Constraint Language) way of expressing the relevant alternative: Level 3 may depend either on Level 1 or on Level 2, or on both of them at the same time.

3. Conformance

The aim of the conformance statements for the CQLF Metamodel is to classify the existing corpus QLs into major categories, corresponding to the CQLF components (classes, levels and modules). This classification is intended to be maximally inclusive and coarse-grained, and is a prelude to a more thorough feature-by-feature classification in the forthcoming parts of the standard.

In order to claim conformance with the CQLF Metamodel, it is enough for a QL to qualify as conformant at any leaf node of CQLF component hierarchy illustrated in Figure 2, as long as the following conditions are also obeyed:

- In order to be able to claim conformance at Level 1 (Linear) of the CQLF Metamodel, a corpus QL must provide support for plain-text search and/or search in segmentation annotations and/or search in simple (non-hierarchical and non-dependency) annotations.
- In order to be able to claim conformance at Level 2 (Complex) of the CQLF Metamodel, a corpus QL must provide support for querying hierarchical annotations and/or dependency annotations and/or annotations (segmentation or simple) related by a containment relationship, and it must be conformant at Level 1.
- In order to be able to claim conformance at Level 3 of the CQLF Metamodel, a corpus QL must provide support for querying the relationships/alignment between simple annotations (minimally) or between a mixture of simple annotations, hierarchical annotations, and dependency annotations, and it must be conformant at Level 1.

It is essential to draw a distinction between CQLF conformance and CQLF-Metamodel conformance; the former must contain a list of references to the (sets of) features defined in CQLF Part 2: CQLF Ontology: "Single-stream architectures". This reference will be in the form of a list of Data Categories (or Data Category Selections) rooted in an instantiation of ISO DCR.

Below, we list several self-documenting examples of conformance statements, extracted from the CQLF ontology.

(1) **Single-stream → 1. Linear → plain-text**

[FRAME]
Look for all plain text spans corresponding to string A

[SIMPLE USE CASE]
Look for all plain text spans corresponding to string A

[QUERY]
hasSyntax: $ grep "this" fileName
isFormulatedIn: grep
comment: This query finds the string "this" in raw text.

(2) **Single-stream → 1. Linear → simple annotation**

[FRAME]
Look for all objects annotated with feature A.

[SIMPLE USE CASE]
Look for all objects annotated with feature A.

[QUERY]

hasSyntax:	MORPH(PRN rel dat)
isFormulatedIn:	Cosmas II
comment:	This query finds all relative pronouns in dative.
hasSource:	[URL skipped]

(3) **Single-stream → 2. Complex → containment**

[FRAME]
Look for all [spans | span sequences] A containing a [span | span sequence] B

[SIMPLE USE CASE]
Look for all multi-token spans A containing span sequence B (startsWith, endsWith, isAround or fullMatch relation).

[QUERY]

hasSyntax:	<s/> containing []* [tag="N.*"] []* [tag="N.*"] []*
isFormulatedIn:	Sketch Engine
comment:	This query looks for all sentences containing more than one noun
hasSource:	[URL skipped]

(4) **Single-stream → 3. Concurrent → overlapping**

[FRAME]
Look for all [spans | span groups] A overlapping a [span | span group] B (incl. full, right, left overlaps and different kinds of containment
relations)

[SIMPLE USE CASE]
Look for all spans annotated with feature A overlapping a span annotated with feature C (incl. full, right, left overlaps and different kinds of containment relations)

[QUERY]

hasSyntax:	Topic="ab" & Inf-Stat="new" & #1 _i_ #2
isFormulatedIn:	ANNIS Query Language
comment:	This query searches for Topic spans with the value "ab" (aboutness topic) that include (_i_) Inf-Stat with the value "new"
hasSource:	[URL skipped]

Note that the present specification does not specify conformance conditions for CQLF Class "Multi-stream". These will be the subject of Part 3, "CQLF Ontology: Multi-stream architectures".

4. Summary and Outlook

The present paper presented an overview of the architecture of CQLF and sketched its further projected development.

CQLF is designed to provide a useful basis for establishing the potential extent and the limits of interoperability between different corpus query systems, and to help reduce the gap between end users with a linguistic or literary background and powerful search environments using modern technology.[3]

The aim is to provide a metastandard that will form the basis for locating any individual corpus QL within a single matrix of a few well-defined properties, and then, gradually, to elaborate on, and extend, the inventory of these properties, as well as to define the relationships between them. An initial set of properties has been designed according to both formal and functional criteria, and may be used as guidelines in the development of a new QL or a new functionality of an existing QL. These properties are part of the 2nd part of CQLF, "CQLF Feature Ontology" (implemented in OWL and currently in the alpha stage), which includes a relatively small ontology to be embedded in an infrastructure that will allow for extensions by individual cQL developers, and for look-ups by end users.

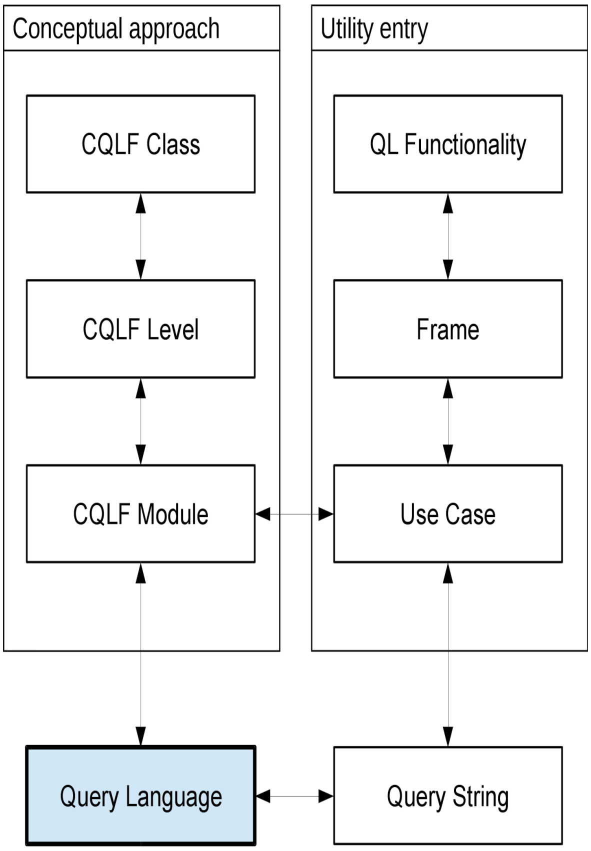

Figure 4: CQLF infrastructure

Figure 4 illustrates the CQLF infrastructure that we envision, with two major entry points: one for developers entering information on their cQLs in order, among others, to claim conformance with CQLF, and the other for end users seeking to identify the range of cQLs that satisfy their requirements.

5. Acknowledgements

We would like to acknowledge the helpful remarks that we have received from the DIN mirror of ISO TC37 SC4, and in particular Stefan Evert and Kerstin Eckart. We are also grateful to Michael Cunningham for his patient assistance with making the specification more accessible, and to Guadalupe Aguado de Cea for inspecting the set of proposed definitions. We wish to stress that all possible errors introduced at later stages of development are our sole responsibility. Finally, we wish to heartily thank three anonymous LREC reviewers for encouragement and useful remarks.

6. References

Bański, P.; Fischer, P. M.; Frick, E.; Ketzan, E.; Kupietz, M.; Schnober, C.; Schonefeld and O., Witt, A. (2012). The New IDS Corpus Analysis Platform: Challenges and Prospects. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey, May 2012. pp. 2905-2911. - European Language Resources Association (ELRA). Available at http://www.lrec-conf.org/proceedings/lrec2012/pdf/789_Paper.pdf

[3] We treat (Evert and Hardie, 2015) as proof that CQLF concepts can be useful for advanced research in our field.

Bański, P.; Bingel, J.; Diewald, N.; Frick, E.; Hanl, M.; Kupietz, M.; Pęzik, P.; Schnober, C. and Witt, A. (2013). KorAP: the new corpus analysis platform at IDS Mannheim. In Vetulani, Zygmunt, Uszkoreit, Hans (Eds.): *Human Language Technologies as a Challenge for Computer Science and Linguistics. Proceedings of the 6th Language and Technology Conference.* pp. 586-587 - Poznań: Fundacja Uniwersytetu im. A.. Available at http://ids-pub.bsz-bw.de/frontdoor/index/index/docId/3261

Bingel, J., Diewald, N. (2015). KoralQuery – A General Corpus Query Protocol. In *Proceedings of the Workshop on Innovative Corpus Query and Visualization Tools at NODALIDA 2015*, May 11-13, 2015, Vilnius, Lithuania. Available at http://www.ep.liu.se/ecp/111/001/ecp15111001.pdf

Clematide, S. (2015). Reflections and a Proposal for a Query and Reporting Language for Richly Annotated Multiparallel Corpora. In *Proceedings of the Workshop on Innovative Corpus Query and Visualization Tools at NODALIDA 2015*, May 11-13, 2015, Vilnius, Lithuania.

Diewald, N.; Hanl, M.; Margaretha, E.; Bingel, J.; Kupietz, M.; Bański, P. and Witt, A. (2016). KorAP Architecture – Diving in the Deep Sea of Corpus Data. In proceedings of LREC-2016.

Evert, S. (2010). The IMS Open Corpus Workbench (CWB) CQP Query Language Tutorial. Version 3.0. Available at http://cwb.sourceforge.net/files/CQP_Tutorial.pdf

Evert, S. and Hardie, A. (2015). Ziggurat: A new data model and indexing format for large annotated text corpora. In Bański, P.; Biber, H.; Breiteneder, E.; Kupietz, M.; Lüngen, H.; Witt, A. (eds.) (2015): Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora (CMLC-3). Mannheim: Institut für Deutsche Sprache, pp. 21-27. Available at http://ids-pub.bsz-bw.de/frontdoor/index/index/docId/3826

Frick, E.; Schnober, C. and Banski, P. (2012). Evaluating Query Languages for a Corpus Processing System. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey, May 2012. pp. 2905-2911 - European Language Resources Association (ELRA). Available at http://www.lrec-conf.org/proceedings/lrec2012/summaries/800.html

Herzog, G.; Heid, U.; Trippel, T.; Banski, P.; Romary, L.; Schmidt, T.; Witt, A. and Eckart, K. (2015). Recent Initiatives towards New Standards for Language Resources. In *Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology*, pp. 154–156, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

Ide, N. (2013). Web Service Exchange Protocols: Preliminary Proposal. Slides presented at ISO TC37 SC4 WG1 meeting on 2 September 2013, in Pisa, Italy. Available at http://www.anc.org/LAPPS/EP/Meeting-2013-09-26-Pisa/overview-ep-2013-09-26-pisa.pdf

ISO/CD 24623-1. Language resource management – Corpus Query Lingua Franca (CQLF) – Part 1: Metamodel.

König, E.; Lezius, W. and Voormann, H. (2003). TIGERSearch 2.1: User's Manual. IMS, University of Stuttgart. Available at http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/TIGERSearch/manual.html

Mueller, M. (2010). Towards a digital carrel: A report about corpus query tools. *Technical report. Unpublished report submitted to the Mellon Foundation.* Available at http://panini.northwestern.edu/mmueller/corpusquerytools.pdf

Odebrecht, C.; Belz, M.; Zeldes, A. and Lüdeling, A. submitted manuscript. RIDGES Herbology - Designing a Diachronic Multi-Layer Corpus. Available at https://www.linguistik.hu-berlin.de/de/institut/professuren/korpuslinguistik/mitarbeiter-innen/carolin/odebrechtetalridges-submitted.pdf