To Boldly Query What No One Has Annotated Before? The Frontiers of Corpus Querying

Markus Gärtner, Kerstin Jung
Institute for Natural Language Processing
University of Stuttgart
{markus.gaertner,kerstin.jung}@ims.uni-stuttgart.de

Abstract

Corpus query systems exist to address the multifarious information needs of any person interested in the content of annotated corpora. In this role they play an important part in making those resources usable for a wider audience. Over the past decades, several such query systems and languages have emerged, varying greatly in their expressiveness and technical details. This paper offers a broad overview of the history of corpora and corpus query tools. It focusses strongly on the query side and hints at exciting directions for future development.

1 Introduction

Annotated corpora have always been the backbone for many fields in NLP and other disciplines related to linguistics. Whether serving as an invaluable source of empirical evidence for foundational research or doubling as gold-standard training input for fueling the furnaces of our machine learning factories, their importance cannot be overemphasized. But especially for the empirically motivated user base, corpora are only ever as good as the means available to explore them. And the primary means of exploring linguistically annotated corpora have always been (dedicated) corpus query tools and corpus query languages in their manifold shapes.

In this paper we intend to give a thorough chronology of the major interplay between corpus progression and query tool evolution, with a strong focus on the latter. We start with an overview on relevant aspects of corpora and how they changed over the past ~30 years in Section 2. Section 3 elaborates on the observable phases in query tool development. In Section 4 we discuss alternative corpus query approaches based on general purpose data(base) management solutions and provide pointers to related work in Section 5. Section 6 summarizes some of our observations and with Section 7 we finally hint at our vision for future directions in corpus query system development.

2 Once Upon a Corpus – Trends in Corpus Evolution

Though corpus linguistics dates back further, major online catalogs such as those from LDC[1] and ELRA[2] list corpora starting from the early 1990s. In the following decades corpus trends have varied along several dimensions, both technical and content-related. This section discusses such features and gives examples for their evolution. Since this overview is an introduction to digital corpus query systems, we mainly focus on written and annotated corpora.

With a focus on written corpora, **character encoding** is a decisive factor when estimating the publication date. Starting from plain ASCII (Everts, 2000[3], Graff and Cieri, 2003) and language/script specific encodings, such as ISO/IEC 8859 (Armstrong-Warwick et al., 1994; Federico et al., 2000), nowadays many corpora come with a (mostly) language independent UTF-8 encoding (Ion et al. (2012); Prasad et al. (2019) and compare Schäfer (2015) with Schäfer and Bildhauer (2012)), which is also able to capture symbols relevant for transcription and annotation.

Similar to character encoding, the preferences regarding the **representation format** for corpus content changed over time. Many corpora established in the 1990s come in an SGML format (Liberman, 1989; Amaryllis, 2001; Graff, 1995). In the next decade, XML-based corpora followed (Chiao et al. (2006) and compare Hajič et al. (2001) and Pajas

[1] Linguistic Data Consortium, https://catalog.ldc.upenn.edu/
[2] European Language Resources Association, http://catalogue.elra.info/
[3] Earlier version published 1997 by ELRA: ISLRN 628-817-117-400-1

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6307–6321
July 5 - 10, 2020.

and Štěpánek (2005)) and since corpora were also made accessible over the web, relational database management systems (RDBMSs) became a valuable backend for corpus storage (Davies, 2005). Today we face a multitude of formats ranging from sophisticated and specialized XML encodings to simple tabular formats and often a corpus comes with more than one representation (Petran et al., 2016; Bick, 2018). Especially since the first CoNLL shared tasks[4], their tabular format to encode sequence-based annotations and relations has been majorly developed (Nivre et al., 2016).

Regarding included **languages**, multilingual and (partly) parallel corpora appear early (Liberman, 1989; Armstrong-Warwick et al., 1994; Graff and Finch, 1994), however, there was a rise of parallel corpora in the first decade of the current century. Prominent examples are Europarl (Koehn, 2005), the CESTA Evaluation Package (Hamon et al., 2006) and the Prague Czech-English Dependency Treebank 1.0 (Cmejrek et al., 2005). On the other hand, with the rise of web corpora, language detection became more important to only crawl (or keep) web data for a specific language.

Corpus **size** is a less discriminative factor than one might think, since many early corpora came as collections of sub-corpora. Armstrong-Warwick et al. (1994) already contains 90 million words and LDC's Gigaword initiative started in 2003 (Graff and Cieri, 2003), while many small corpora for specific topics or containing manual annotations are constantly being created. Nevertheless, with recent web corpora, e.g. ENCOW16[5] and iWEB[6], several billion tokens pose new challenges for the design of both storage and search facilities.

While for spoken corpora **domain** selection is often tailored to the research question at hand (cf. Talkbank (MacWhinney et al., 2004)), for written corpora (and especially annotated ones) there is a bias towards news and official documents, which was superseded by multi-domain web corpora starting in the late 2000s (e.g. the WaCKy initiative (Baroni et al., 2009) and COW) and, in the follow up, the increasing number of corpora of computer-mediated communication and social media[7]. Like with the language setting, for web-corpora the challenge is no longer to include more languages or domains, but to identify and/or restrict them to a sensible subset. Collections of historical language data have also been available for some time, e.g. the Corpus of Middle English Prose and Verse[8] and with the rise of the Digital Humanities many further corpora are created and/or enhanced with linguistic annotations, such as the Drama Corpora Project[9], where some corpora have been enhanced with lemma information.

Most corpora come with **annotations**, the earlier ones mainly with flat and word-based annotations, mostly including part-of-speech, such as the ECI-ELSNET Italian & German tagged sub-corpus[10]. Regarding the structural aspect, stand-off syntactic annotations became more feasible with emerging treebanks, while over time the focus changed from phrase-based (Brants et al., 2004) to dependency tree structures (Hajič et al., 2001). The current decade has also seen an increase in the richness of annotation layers of morphological, syntactical and semantical description, including highly concurrent annotations belonging to the same description layer, e.g. Ide et al. (2010) or Schweitzer et al. (2018).

3 A Brief History of Querying

We observed three major phases or generations in the history of corpus query systems, which are roughly aligned to the last three decades. The following is meant as a comprehensive but not exhaustive chronology of corpus query systems and approaches. Space does not permit we provide in-depth descriptions for every system mentioned but instead refer to Section 5 for pointers to existing work that discusses and compares certain (families of) query systems in detail.

3.1 First Generation – Humble Beginnings

The history of corpus querying systems has been for the most part tightly connected to the gradual expansion of the targeted corpus resources. As such the initial wave of corpus query tools during the 1990s was mostly geared towards text corpora:

The **COSMAS**[11] lineage remains until today[12]

[4] https://www.conll.org/previous-tasks

[5] **CO**rpora from the **W**eb (COW), English sub-corpus, https://corporafromtheweb.org/

[6] https://www.english-corpora.org/

[7] Annual conference on computer-mediated communication and social media corpora started in 2013 https://sites.google.com/site/cmccorpora/

[8] https://quod.lib.umich.edu/c/cme/

[9] https://dracor.org/

[10] ISLRN 869-857-775-378-7

[11] Corpus Search, Management and Analysis System, http://www.ids-mannheim.de/cosmas2/

[12] The initial version COSMAS I has been in continuous service from 1992 till 2003 and COSMAS II ever since 2002

the public query front-end for the large corpus collection hosted at the IDS (Bodmer, 2005), offering keyword in context (KWIC) visualization in a browser-frontend and various query constraints.

In contrast the Linguistic DataBase program (**LDB**) (Halteren and Heuvel, 1990) features a very expressive tree-based query syntax and also ships with a tree editor. In addition it provides an ingenious event-based approach for extracting information from a corpus during search.

The **Corpus Workbench (CWB)** architecture (Christ, 1994) with the Corpus Query Processor (CQP) as its core component is maybe the most widely used corpus query system as of today, serving as the backend for many corpus exploration websites. Having been under continuous maintenance to keep up with the demands of the new century (Evert and Hardie, 2011), it provides a solid set of simple yet expressive search features, such as regular expressions over tokens and token content, flexible structural boundaries, support for parallel corpora or the ability to enrich a corpus during ingest with external data that can then be used for querying, e.g. WordNet (Miller, 1995) categories.

Emu (Cassidy and Harrington, 1996) was designed for speech corpora with multiple levels of segmentation. Primarily a hierarchical speech data management system, it also supports label- and position-based queries for collections of tokens.

Similarly the **MATE** Workbench (Mengel, 1999; Mengel et al., 1999; Heid and Mengel, 1999; Isard et al., 2000) also targets combinations of text and speech data in the form of XML annotation files. It provides full boolean operations over hierarchical and time-based constraints in a logic-style query language, but no direct support for quantifiers.

3.2 Second Generation – The Rush for Rapid Feature Expansion

At the dawn of the 21st century the second and larger wave of query systems emerged. Initially focused heavily on treebanks annotated for phrase-based syntax, a later trend shifted more towards supporting dependency syntax annotations, with an overall theme of increasing expressiveness with new approaches to query syntax and constraints.

TIGERSearch (König and Lezius, 2000; Lezius, 2002) was among the first with its logic-based query language to target phrase-based treebanks conforming to the TIGER model (Brants et al., 2004). It inspired many of the later query approaches, but was quickly surpassed wrt expressiveness due to limited negation or quantification[13].

The ICE Corpus Utility Program (**ICECUP**)[14] introduced a completely new direction of development. Wallis and Nelson (2000) emphasized the complexity required to transform a two-dimensional tree description into a linear sequence of textual expression and made an argument for a graphical query approach. Their fuzzy tree fragments act as visual (under-)specification of the targeted phrase-based tree structures and are then matched against instances in a corpus. The appeals of this approach are diverse: It enables example-based searching by allowing the user to start from an existing instance in the corpus, transform it into a query and then relax the constraints on that query to generalize it[15]. Not having to learn a formal query language and annotation schemes first, also lowers the barrier to entry for successful querying.

As a dedicated treebank query tool **TGrep2** (Rohde, 2001) offers a rich query syntax for phrase-based treebanks. Notable features are conjunction, disjunction and negation for relations, over 30 predefined basic link types and the ability for users to simplify complex queries by using macros.

Usually corpus query tools depend on the target data already being annotated. **Gsearch** (Corley et al., 2001) however lets the user query unstructured text data by parsing it on the fly with a chart parser. Gsearch queries contain phrase-based constraints with limited boolean operators and the results are emitted in SGML.

VIQTORYA[16] (Steiner and Kallmeyer, 2002) is another tool to query phrase-based treebanks. Its query syntax is very similar to TIGERSearch[17] and queries are translated for the RDBMS backend.

Outside the domain of monolingual corpora **ParaConc** (Barlow, 2002) combines typical concordancer functionality such as surface search and

[13]The developers decided to forgo universal quantification due to computational cost and tractability (TIGERSearch Help, section 10.3) but also proposed an extension of the language with universal quantification and the implication operator. Marek et al. (2008) mention a solution based on set operations over multiple queries. This “allows to express queries which need a universal quantifier if expressed in a single query”. Unfortunately the referenced term paper is not available online.

[14]Designed for ICE-GB, the British component of the International Corpus of English (Nelson et al., 2002).

[15]Described by Wallis and Nelson (2000) as the ’get me something like *that*’ query method.

[16]**V**isual **Q**uery **T**ool f**o**r Synt**a**ctically **A**nnotated Corpora

[17]Consisting of the same quantifier-free subset of first-order logic, but different precedence definition of internal nodes (cf. Steiner and Kallmeyer (2002) and Clematide (2015)).

KWIC result view with regex and tag search and applies it to parallel corpora as targets.

The **CorpusSearch** (Taylor, 2003; Randall, 2008) command line tool for phrase-based syntax expects tree search configurations provided via query files with a boolean query language over a variety of tree predicates and regular expressions. Limitations on disjunction and negation and lack of quantification[18] make it slightly less expressive.

With full first-order logic the Finite Structure Query (**FSQ**) tool by Kepser (2003) offers access to the complete TIGER model, including arbitrary secondary edges and support for regular expressions in a graphical user interface (GUI). It is however limited to rather small corpora due to poor scalability of the query evaluation process.[19]

To access multi-modal and highly cross-annotated data in the NITE Object Model Library (Carletta et al., 2003), Evert and Voormann (2002) specified the **NITE Query Language (NiteQL)** based on MATE. Information from various segmentation levels can be extracted and combined in a logic-style language, including limited quantification. To honor the nature of multi-modal data they also propose a level of "fuzziness" for time operators with a configurable *fuzziness interval*.

Based on the MdF (Monads-dot-Features) Database and its query language QL by Doedens (1994), **Emdros** (Petersen, 2004) implements a text database for annotated texts. Its query syntax uses bracket nesting to express hierarchical relations and it surpasses TIGERSearch in several aspects of expressiveness, e.g. existential negation[20].

While previously mentioned query systems were either freely available or bound to the licensing model of associated corpus resources (e.g. ICE-CUP), the popular **Sketch Engine** (Kilgarriff et al., 2004) commercialized[21] corpus management and exploration in a web-based platform (Kilgarriff et al., 2014). Extending the CQP, its own query language CQL offers efficient access to corpora available on the platform (Jakubíček et al., 2010).

Around the same time **ANNIS** was published (Dipper and Götze, 2005) and started a successful ecosystem with the corpus metamodel SALT, the converter framework PEPPER and ANNIS itself as search module with its query language AQL. AQL is a very expressive query language on top of the graph-based model of SALT and an extension of the TIGERSearch syntax. Notable improvements over TIGERSearch are the access to concurrent annotations for the same layers, a rich set of segment relations to choose from and the generalization of directed relations in a query to be applicable for any type of edge in the corpus graph (e.g. syntax, coreference or alignments in parallel corpora). Queries in ANNIS can be constructed textually or graphically in a browser environment. It has been under continuous development for about 15 years now (Zeldes et al., 2009; Krause and Zeldes, 2014), resulting in the richest collection of result visualizations available in any corpus query system.

The **Linguist's Search Engine** (LSE) (Resnik and Elkiss, 2005) applies the query-by-example concept in a browser-based setting: A user provides a natural language example containing the desired phenomenon and receives a parse tree usable for querying. Relaxation or removal of constraints from this tree then yields increasingly generalized instances from built-in or custom collections[22].

The emergence of XPath[23] as a way of querying the tree-structure of various XML-based corpora offered new directions for corpus query languages. Bird et al. (2006) introduced **LPath** as an extension of XPath to overcome its limitations regarding the lack of expressible horizontal relations, a feature crucial for querying linguistic data. A later extension turned it into a first-order complete variant named **LPath+** (Lai and Bird, 2005).

Faulstich et al. (2006) also used an extension of XPath called **DDDQuery** to query complex annotation graphs of historical texts[24]. While using a RDBMS as backend, they do not directly translate queries into SQL. Instead user queries are first transformed into a first-order logic intermediate representation which in turn is translated into SQL.

The Prague Dependency Treebank (PDT) (Hajič et al., 2001; Hajič, 2006) is a richly annotated corpus. Its unique characteristic is a tectogram-

[18] The way negation on arguments to *search-function calls* is handled allows to express certain quantified relations though.

[19] The author of FSQ discusses those limitations in (Kepser, 2004) and proposes a solution based on monadic second-order logic which was later implemented in MonaSearch.

[20] See Petersen (2005) for a brief comparison of the two systems including benchmarks on example queries.

[21] An open-source part under the label NoSketch Engine with the Manatee backend for indexing and search is also available at `https://nlp.fi.muni.cz/trac/noske`.

[22] The "Getting Started Guide" (`http://hdl.handle.net/1903/1324`) for LSE mentions TGrep2 as the search component. In Resnik and Elkiss (2005) this information is missing and the screenshots do not show textual TGrep queries anymore, so the actual query evaluation backend is unknown.

[23] `https://www.w3.org/TR/xpath`

[24] `http://www.deutschdiachrondigital.de/`

matical layer which also includes annotations for coreference, deep word order, topic and focus. To provide users with adequate tools for access to this complexity, **NetGraph** (Ondruška et al., 2002; Mírovský, 2006) allows creation of tree queries for various layers both textually and graphically.[25]

Stockholm TreeAligner (Lundborg et al., 2007; Marek et al., 2008) continues the trend of extending the TIGERSearch language and applies it to parallel corpora. Its main improvement is the (re)introduction and implementation of universal quantification to overcome this central weakness.

Classic query tools for text corpora such as CQP lack the ability to efficiently deal[26] with common features of annotations for morphologically rich languages, such as positional tagsets or non-disambiguated annotation instances. **POLIQARP**[27] (Przepiórkowski et al., 2004; Janus and Przepiórkowski, 2007) is an indexer and query tool loosely based on the CQP approach with a client-server architecture and a variety of available client implementations. Initially targeted towards rich word-level annotations, such as in the IPI PAN Corpus (Przepiórkowski, 2004), it was later extended to also cover syntactic-semantic treebanks.

What's wrong with my NLP? by (Riedel, 2008) is primarily meant as a visualization tool with the ability to highlight differences between two concurrent dependency annotations (e.g. a gold standard and automatic predictions) with search options based on surface forms, tags and as a neat feature also including aforementioned diffs.

Maryns and Kepser (2009a) extended the expressiveness of FSQ to monadic second-order logic in **MonaSearch**. It features a GUI for viewing text-only "flat" results and defining queries of enormous expressiveness. However, due to the limitations of the underlying MONA framework (requiring binary tree structures), the system can only target collections of proper trees.

PML-TQ[28] (Pajas and Štěpánek, 2009; Štěpánek and Pajas, 2010) is effectively the successor of NetGraph, being designed to handle the rich multi-level annotations in the PDT. Its graphical client[29] is directly integrated into the tree editor TrEd (Pajas, 2009) to support graphical query construction. Queries in PML-TQ are expressed as a mandatory selection part in bracket-syntax and an optional list of instructions to generate result reports. The latter of those two parts was groundbreaking in that it allows for an unprecedented freedom in selectively extracting information from any successful match during a search and creating various aggregations or statistics from it. Besides excellent result handling its query language is also quite powerful, including quantification and negation of sub-queries.

3.3 Third Generation – New Challenges

During the last decade the speed at which new query tools have been developed or published slowed down considerably. At the same time continued growth in size of corpus resources rendered some of the earlier approaches inapplicable (cf. (Kepser, 2004) for a discussion on the limitations of FSQ), calling for innovative alternatives. The three most common themes of this era were (i) scalability and adaptability of search backends to keep up with the explosive growth of corpora, (ii) reducing the barrier to entry for a wide(r) range of potential users and (iii) working towards unification or standardization of query languages.

GrETEL[30] (Augustinus et al., 2012) is another implementation of the example-based search concept for the LASSY corpus (van Noord et al., 2013). Users provide sentences or example fragments and mark the areas of interest. Examples are then parsed, the subtrees for the specified part(s) of the input extracted and subsequently translated into XPath queries to run against the corpus in XML format. Further query options include the ability to specify whether or not pos, lemma or surface form of tokens in the subtree should be considered for the query. Since the user is effectively shielded from the tree representation and formal query formulation, GrETEL requires neither knowledge of an actual query language nor about the annotation scheme or underlying theories of the corpus.

Fangorn (Ghodke and Bird, 2012) addresses the challenge of querying treebanks too large to be loaded into memory, a scenario prohibitive for

[25] Besides NetGraph the tree visualizer and editor software **TrEd** (Pajas, 2009) also can be used to search in PDT and other tree structures via user macros defined in Perl. It does however not offer a query language for non-programmers.

[26] This does not imply their expressiveness being insufficient for this task, but rather that such queries can become quite bloated and their construction cumbersome for users.

[27] **POL**yinterpretation **I**ndexing **Q**uery **A**nd **R**etrieval **P**rocessor

[28] **P**rague **M**arkup **L**anguage - **T**ree **Q**uery

[29] The modular architecture supports multiple scenarios, such as a client-server setup with an RDBMS backend or an integrated index-less query evaluator in Perl for local data.

[30] **Gr**eedy **E**xtraction of **T**rees for **E**mpirical **L**inguistics

query tools with custom evaluation engines. They use Apache LUCENE[31] in a client-server setup to manage large numbers of phrase structure trees. Its query language follows the LPath scheme but lacks regular expressions support on label content.

Unlike the majority of other systems in recent years, we developed **ICARUS**[32] (Gärtner et al., 2013) as a standalone desktop application for visualization and example-based search[33] with a custom query evaluation system and no indexing or dependency on another database technology. Initially designed for querying dependency treebanks it underwent multiple extensions to make it compatible with annotations for coreference (Gärtner et al., 2014) and prosody[34] (Gärtner et al., 2015) and also to incorporate automatic error mining as a means of exploration (Thiele et al., 2014). Its bracket-style query language is similar to PML-TQ but lacks quantifiers and a dedicated section for result preparation instructions. While queries can be defined both textually or graphically, the preferred way is to use the graphical query editor that also provides contextual help for getting started easily.

CLARIN Federated Content Search[35] (**CLARIN-FCS**) is a successful example of unifying query access to multiple distributed corpus resources hosted by different parties and with diverse *native* query frontends. Its query language FCS-QL is heavily based on POLIQARP but also only meant to cover a small intersection of the expressiveness of common corpus query tools.

On the level of standardization **CQLF**[36] (Bański et al., 2016) provides an initiative that aims at providing means for comparability and interoperability of corpus query languages. In its first phase[37] CQLF-1 defines classes and features for the description of query languages for single-stream data.

A unified serialization format for CQLF-1 is available with **KoralQuery** (Bingel and Diewald, 2015), a JSON-LD based and theory-neutral corpus query protocol. It serves as the internal query representation[38] of **KorAP**[39] (Bański et al., 2014; Diewald et al., 2016), the designated successor of COSMAS II. While CLARIN-FCS multiplexes a query defined in a common (limited wrt expressiveness) query language to multiple query processors, KorAP lets the user choose up-front among several query languages[40] that all can be processed by the system in a microservices architecture[41].

Similar to Fangorn, **SETS**[42] (Luotolahti et al., 2015) is geared towards very large treebanks, this time targeting dependency syntax with a query language inspired by TRegex[43]. It is browser-based with a RDBMS backend and uses an elaborate query evaluation process: SETS generates and compiles optimized code for matching tokens for each query and only retrieves the minimal token sets from the database needed for evaluating a query.

Multilingwis[44] (Clematide et al., 2016) provides exploration in multiparallel corpora (Graën et al., 2016). Focused on result presentation and reducing the required expert knowledge, it simplifies the process of finding translation variants.

Other notable events in this time period include the **modernization of CQP** "for the new millennium" (Evert and Hardie, 2011) and the introduction of **graphANNIS** (Krause et al., 2016), a graph database backend for ANNIS3 as an alternative to the former RDBMS-based relANNIS.

4 Technological Alternatives

Many of the systems we presented in Section 3 use various forms of database technology as their storage or evaluation backend. Typically every such database or information management system already ships with its dedicated query language, such as SQL for RDBMSs, SPARQL for the RDF format, XPath and XQuery for XML documents, CYPHER for Neo4j and other graph-based databases or Apache LUCENE with its own query dialect for accessing the text database.

[31] `https://lucene.apache.org/`

[32] **I**nteractive Platform for **C**orpus **A**nalysis and **R**esearch, **U**niversity of **S**tuttgart

[33] An integrated interface for plugging in dependency parsers allows users to generate parses for example sentences that can then be converted into queries and relaxed iteratively.

[34] With various similarity measures usable for expressing query constraints based on the PaIntE model by Möhler (2001)

[35] `https://www.clarin.eu/content/content-search`

[36] **C**orpus **Q**uery **L**ingua **F**ranca. Part of ISO TC37 SC4 Working Group 6 (ISO 24623-1:2018).

[37] CQLF is an ongoing long-term effort, with CQLF-2 currently being worked on at the stage of a committee draft.

[38] The high level of abstraction it implements and the verbosity required to express simple queries combined with JSON syntax results in limited human readability.

[39] **Kor**pus**a**nalyse**p**lattform der nächsten Generation ("Corpus analysis platform of the next generation")

[40] At the time of writing it supports the following query languages: Poliqarp, FCS-QL, AQL, CQP 1.2, COSMAS II

[41] KorAP builds on a variety of (storage) technologies, inluding several RDBMS variants, LUCENE and also the graph database Neo4j (`http://neo4j.com/`).

[42] **S**calable and **E**fficient **T**ree **S**earch

[43] A "Tree regular expression" language in TGrep2 style

[44] **Multilingu**al **W**ord **I**nformation **S**ystem

This does of course prompt the question on the necessity of developing dedicated corpus query languages when more often than not the actual query evaluation is just offloaded to an existing database technology. Already Jarke and Vassiliou (1985) mentioned a plethora of (technical) factors to be considered when deciding on a (database) query language. Mueller (2010) on the other hand takes the perspective of scholarly users, providing arguments especially targeting the aspects of usability from a humanistic point of view, describing the handling of search results as "Achilles heel of corpus query tools". Having previously examined those factors in (Gärtner and Kuhn, 2018), we also agree on the continuing necessity of dedicated corpus query systems and query languages to bridge the gap between formal/technical expressiveness and the usability factors decisive for corpus users. Especially future directions as the ones we propose in Section 7 demand architectures that are more complex than the mere translations of data and queries.

There have however also been approaches or use case analyses to completely store and query linguistic corpora with OWL (Burchardt et al., 2008), XQuery (Cassidy, 2002) or a via RDBMS (e.g. content of the DIRNDL corpus (Eckart et al., 2012) in its entirety has for a long time only been available through direct SQL queries), but historically speaking those cases generally represent a minority.

5 Related Work

A lot of work has been invested already into laying the theoretical foundations for various aspects of and approaches to corpus querying, as well as into evaluating and comparing existing query systems. We distinguish between three types of contributions, namely (i) requirement analyses, (ii) evaluations of individual query languages or approaches and (iii) actual performance comparisons between multiple systems (feature-based or benchmarks).

Several contributions listing **requirements** for corpus query systems have been previously mentioned in Section 4. In addition, Mírovský (2008) provides a list of required language features for querying PDT and Lai and Bird (2004) do so for treebanks in general, specifically related to navigation, closures over relations and going "beyond ordered trees" in order to query more complex structures. This list of functional requirements is later extended on in Lai and Bird (2010) with features such as temporal organization and non-navigational requirements. While not exclusive to corpus query systems, technical aspects related to feasibility (e.g. scalability or computational complexity) or long-term maintainability (e.g. interoperability and extensibility) are also frequently emphasized by Lai and Bird (2004), Kepser (2003) and others. Besides the usability-focused scholarly position of Mueller (2010) around aspects of answer time, maintenance cost and the management of search results, we previously discussed additional non-technical requirements related to the general readability or post-processing capabilities of a query language and its learnability in Gärtner and Kuhn (2018), the latter being a crucial factor for achieving wide-spread use in humanistic fields.

Formal **evaluations** of query languages are somewhat rare, e.g. (Lai and Bird, 2010) for LPath and LPath$^+$, (Kepser, 2004) for MonaSearch or in part (Kepser, 2003) for FSQ. Instead the vast majority of evaluations use example queries of varying complexity to compare different query languages or systems. Notable early work on query complexity was done by Lai and Bird (2004), comparing several query languages[45] based on a set of linguistic information needs of increasing complexity. The example queries they provide have proven to be a good baseline for comparing the capabilities of query languages and subsequently found their way into many later tool evaluations, such as (Petersen, 2006a) for Emdros or in Clematide (2015) when highlighting features of particular query languages. Yet another evaluation approach was used by Frick et al. (2012) when they applied the classes defined in CQLF-1 as evaluation criteria in the comparison of COSMAS II, POLIQARP and AQL.

Clematide (2015) provides a very thorough reflection and categorization of the various **families** of corpus query languages: text corpus, treebank, path-based[46] and logic-based. A point he makes that resonates well with other surveys is the importance of striking the right balance between usability and technical aspects in any practical situation.

In some cases actual **performance benchmarks** have been published, such as testing Emdros with different RDBMS backends (Petersen, 2006b),

[45]TGrep2, TIGERSearch, Emu, CorpusSearch, NiteQL, LPath

[46]We argue for a more differentiated view on path-based query languages: While Clematide (2015) considers PML-TQ to be part of this family, we propose to move it together with ICARUS into a *tree-based* category of query languages, as their use of bracketed tree-expressions to describe structural relations represent a slightly different approach.

comparisons between TIGERSearch and Emdros in Petersen (2005), MonaSearch and TIGERSearch in (Maryns and Kepser, 2009b) and Luotolahti et al. (2015) benchmarking SETS against ICARUS. However, due to the rapid change in technologies and the architectural differences between query systems, it tends to be very difficult to provide accurate and meaningful performance comparisons and readers are advised to carefully examine whether the reported use cases are applicable to their own.

6 Key Observations & Shortcomings

In this section we intend to condense some of our observations after analyzing a large number of query systems. We focus on the following two aspects suitable for pointing out challenges (stemming from past shortcomings) and motivating directions in development of future corpus query systems, protocols or architectures.

6.1 Shifting Design Goals

The different generations of corpus query systems listed in Section 3 are the results of design processes with generally very distinct goals. The first generation in Section 3.1 can be seen as the initial step to have *some* means of querying beyond the search functions of grep or any text editor.

Subsequently, the second time period described in Section 3.2 represents a general **exploration phase**: Approaches in almost every direction were implemented, either as proof of concept for new query features or to address very specific linguistic theories or phenomena. Many of those implementations however were not scalable to the degree demanded by the rapid growth[47] of corpora.

As such the general trend in Section 3.3 was to overcome those limitations and provide **scalable systems** with also increased usability. At the same time the overall expressiveness of query languages provided took a step backwards. Especially concepts like closures over relations, (universal) quantification or existential negation often got rationalized in favor of performance in younger systems. Our vision of a hybrid architecture sketched in Section 7 is intended to overcome those limitations by utilizing and combining the different strengths of systems involved (such as the robust performance of indexing systems and the expressiveness and flexibility of custom query evaluation engines).

[47] Growth continually occurred both in size (number of primary units) and complexity (number of annotation layers).

6.2 Fragmentation & Limited Reusability

With the enormous amounts of resources that have been invested into creating this zoo of corpus query languages and systems, it is surprising how little reuse and unification has occurred over the years. We attribute this trend to a variety of frequently recurring factors, particularly the following:

- Due to the **lack of standards** regarding the categorization of expressiveness of query languages it has always been extremely difficult to determine whether an existing system could meat all the requirements a new project, user scenario or corpus resource posed, leading to redundancy.[48]
- The **technological heterogeneity**[49] involved also represented a major issue that only slowly is being overcome by the emergence of standards for corpus storage and interchange formats or the shift to more modular architectures such as microservices or plugin-engines, making it much easier to adapt a system to new requirements.[50]
- Especially early query systems often emerged as an interface for a very particular corpus, a specific format or to support the phenomena a certain project was interested in. As such, the **limited resources** typically available for short-term funded projects rarely allowed for extending previous monolithically designed work. Newly implemented (and often isolated) solutions focusing on a narrow selection of very specific query features or annotations were a common result.

7 The Final Frontier – An Outlook

With several dozens of systems contributing their individual variations, the pool of available corpus query tools and languages has become quite large. Navigating this ocean in order to find the right tool for the job and then learn to use it can already be as much effort as manually investigating the data at hand. Fortunately the CQLF standardization initiative aims at providing developers with the means of locating their tools on a map of query features, so that prospective users may find them without an odyssey. While this effort is still in an early stage, we are looking forward to having catalogs available

[48] An aspect that CQLF is now addressing, removing the need of essentially reverse engineering a tool or studying its source code, as time constraints together with the lack of standardization often went along with poor documentation.

[49] Ranging from platform/language lock-ins to format/storage dependencies, often in a monolithic composition.

[50] Such as new query features, formats, storage/database solutions, standalone apps or various client-server architectures.

in the not too distant future, allowing us to browse for query languages based on our individual information needs. However, many questions regarding the future of corpus querying still remain, two of which we consider of particular importance and will discuss in the following sections.

7.1 One Language to Query Them All?

Today we have a cluttered buffet of corpus query languages to pick from depending on our information needs. Interestingly they all share the pros and cons of being designed as **formal languages** with the goal of taciturnity, meaning that for the untrained eye they usually represent just a weird salad of letters and special characters .[51] This is particularly noteworthy, as all modern corpus query tools feature a rich GUI and could easily employ a more verbose query language while at the same time shield users from the time overhead when creating queries by clever auto-completion or recommendation functions.

Likewise, today's corpus queries are not **self-contained** to the level of for instance SQL queries, which are composed of dedicated parts for scope selection, actual constraints and result preparation. Usually only the constraint part is present in corpus query languages, with only a few exceptions [52], leaving additional configurations (result size limit, search direction, case sensitivity) exclusively to external components, such as the GUI, hampering the reproducibility of search results severely.

A fully self-contained and **human-readable** query protocol that can embed any existing query language and augment it with (boilerplate) statements to bind the query content to actual corpora and annotation layers, provide information about the query dialect and its version and store configuration and result preparation instructions, would go a long way towards unification and potential interoperability of corpus query systems.

7.2 Towards a Hybrid Architecture?

The typical architecture of corpus query systems today is a monolithic one and contains from bottom to top (i) a backend storage or custom data model, (ii) a custom query evaluator or query interface to said backend and (iii) a query parser or translator to process the raw user query. Choices in technology or algorithms for (i) through (iii) definitively dictate the basic nature and structure of the information that can be queried. They usually make it very difficult, if not impossible, to implement changes or extensions retrospectively or from the outside. A strong dependency on indexing to access large corpora also presupposes a priori knowledge of what information is meant to be searchable, frequently confining corpus query tools to the role of being mere finding aids within a research process.

We would like to see them become true enablers instead, allowing queries to go far beyond of what a corpus has to offer with its bare annotations alone and for example include the following extensions to create more informed search solutions:

- Use knowledge bases and similar external resources to allow more generalized queries, e.g. "find verbal constructions containing a preposition in combination with some sort of *furniture*".
- Add (semantic) similarity measures (e.g. word embeddings) and other approaches for increased *fuzziness* to improve example-based search.
- Offer true scripting support for users to extent or customize the ability provided by a system. While this might affect performance in unpredictable and detrimental ways, raw (distributed) computing power and clever use of pre-filtering can offset the impacts on performance.

Naturally all of these proposed features (and especially the last one) require a drastically different and quite heterogeneous architecture. Taking the microservices approach of KorAP as an example, it is easy to imagine a hierarchically organized architecture of query translation and evaluation services working together (by partially answering queries, filtering the results or otherwise post-process them) to provide the optimal combination of freedom in expressiveness and performance guarantees. Space does not permit we provide a detailed description of such a hybrid approach. Instead we refer to (Gärtner, to appear) for an overview of our ongoing efforts to design and implement a hybrid corpus query architecture and associated query protocol. Twenty years ago this might have seemed utterly unrealistic, but advances in information management systems and distributed computing certainly put this vision within technical reach.

[51] Kaufmann and Bernstein (2010) investigated the usability of natural language queries for interfaces to the semantic web with positive results. It would be interesting to see similar studies on corpus query interfaces.

[52] cf. PML-TQ for exemplary post-processing instructions, allowing to treat results as tabular data and to perform various transformation and aggregation operations on it, including textual reports.

References

Amaryllis. 2001. Amaryllis corpus - evaluation package. ELRA. ISLRN: 786-395-313-491-8.

Susan Armstrong-Warwick, Henry S. Thompson, David McKelvie, and Dominique Petitpierre. 1994. Data in your language: The ECI Multilingual Corpus 1. In *Proceedings of the International Workshop on Sharable Natural Language Resources*, Nara, Japan.

Liesbeth Augustinus, Vincent Vandeghinste, and Frank Van Eynde. 2012. Example-based treebank querying. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 3161–3167, Istanbul, Turkey. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1442.

Piotr Bański, Joachim Bingel, Nils Diewald, Elena Frick, Michael Hanl, Marc Kupietz, Piotr Pęzik, Carsten Schnober, and Andreas Witt. 2014. KorAP: the new corpus analysis platform at IDS mannheim. Human language technology challenges for computer science and linguistics. 6th language & technology conference december 7-9, 2013, Poznań, Poland, pages 586 – 587, Poznań. Uniwersytet im. Adama Mickiewicza w Poznaniu.

Piotr Bański, Elena Frick, and Andreas Witt. 2016. Corpus Query Lingua Franca (CQLF). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Michael Barlow. 2002. ParaConc: Concordance software for multilingual parallel corpora. In *Proceedings of the Third International Conference on Language Resources and Evaluation. Workshop on Language Resources in Translation Work and Research., Las Palmas, Canary Islands - Spain*, pages 20–24.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Eckhard Bick. 2018. Arbobanko - A treebank for Esperanto. In *Proceedings of CICLing 2018 - 19th International Conference on Computational Linguistics*, Germany. Springer.

Joachim Bingel and Nils Diewald. 2015. *KoralQuery – A General Corpus Query Protocol*, volume 111, pages 1–5. Linköping University Electronic Press.

Steven Bird, Yi Chen, Susan B Davidson, Haejoong Lee, and Yifeng Zheng. 2006. Designing and evaluating an XPath dialect for linguistic queries. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 52–52. IEEE.

Franck Bodmer. 2005. COSMAS II - Recherchieren in den Korpora des IDS. *Sprachreport : Informationen und Meinungen zur deutschen Sprache*, 21(3):2 – 5.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic interpretation of a German corpus. *Research on Language and Computation*, 2(4):597–620.

Aljoscha Burchardt, Sebastian Padó, Dennis Spohr, Anette Frank, and Ulrich Heid. 2008. Constructing integrated corpus and lexicon models for multi-layer annotations in OWL DL. *Linguistic Issues in Language Technology*, 1:1–33.

Jean Carletta, Jonathan Kilgour, Tim O'Donnell, Stefan Evert, and Holger Voormann. 2003. The NITE object model library for handling structured linguistic annotation on multimodal data sets. In *In Proceedings of the EACL Workshop on Language Technology and the Semantic Web (3rd Workshop on NLP and XML, NLPXML-2003)*.

Steve Cassidy. 2002. XQuery as an annotation query language: a use case analysis. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Steve Cassidy and Jonathan Harrington. 1996. EMU: an enhanced hierarchical speech data management system. In *Proceedings of the Sixth Australian International Conference on Speech Science and Technology*, pages 361–366.

Yun-Chuang Chiao, Olivier Kraif, Dominique Laurent, Thi Minh Huyen Nguyen, Nasredine Semmar, François Stuck, Jean Véronis, and Wajdi Zaghouani. 2006. Evaluation of multilingual text alignment systems: the ARCADE II project. In *Proceedings of the fifth International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1975–1979, Genoa, Italy.

Oliver Christ. 1994. A modular and flexible architecture for an integrated corpus query system. In *Proceedings of COMPLEX'94: 3rd Conference on Computational Lexicography and Text Research*, pages 23–32, Budapest.

Simon Clematide. 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*, 111, pages 6–16. Linköping University Electronic Press, Linköpings universitet.

Simon Clematide, Johannes Graën, and Martin Volk. 2016. Multilingwis – a multilingual search tool for multi-word units in multiparallel corpora. In Gloria Corpas Pastor, editor, *Computerised and Corpus-based Approaches to Phraseology: Monolingual* *and Multilingual Perspectives/Fraseología computacional y basada en corpus: perspectivas monolingües y multilingües*, page n/a. Tradulex, Geneva.

Martin Cmejrek, Jan Curín, Jan Hajic, and Jirí Havelka. 2005. Prague Czech-English dependency treebank resource for structure-based MT. In *EAMT 2005 Conference Proceedings*, pages 73–78.

Steffan Corley, Martin Corley, Frank Keller, Matthew W. Crocker, and Shari Trewin. 2001. Finding syntactic structure in unparsed corpora the Gsearch corpus query system. *Computers and the Humanities*, 35(2):81–94.

Mark Davies. 2005. The advantage of using relational databases for large corpora: Speed, advanced queries, and unlimited annotation. *International Journal of Corpus Linguistics*, 10(3):307–334.

Nils Diewald, Michael Hanl, Eliza Margaretha, Joachim Bingel, Marc Kupietz, Piotr Bański, and Andreas Witt. 2016. KorAP architecture – diving in the deep sea of corpus data. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3586–3591, Portorož, Slovenia. European Language Resources Association (ELRA).

Stefanie Dipper and Michael Götze. 2005. Accessing heterogeneous linguistic data – generic XML-based representation and flexible visualization. In *Proceedings of the 2nd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 206–210, Poznan, Poland.

Christ-Jan Doedens. 1994. *Text Databases: One Database Model and Several Retrieval Languages*, volume 14 of *Language and Computers*. Brill Rodopi.

Kerstin Eckart, Arndt Riester, and Katrin Schweitzer. 2012. A discourse information radio news database for linguistic analysis. In Christian Chiarcos, Sebastian Nordhoff, and Sebastian Hellmann, editors, *Linked Data in Linguistics. Representing and Connecting Language Data and Language Metadata*, pages 65–75. Springer, Heidelberg.

Stefan Evert and Andrew Hardie. 2011. Twenty-first century Corpus Workbench: Updating a query architecture for the new millennium. In *Proceedings of the Corpus Linguistics 2011 conference*, Birmingham.

Stefan Evert and Holger Voormann. 2002. The NITE query language.

Karlheinz Everts. 2000. Das Karl-May-Korpus – Ein linguistisch annotiertes Korpus der Werke des Autors Karl May und einiger seiner Zeitgenossen. Aufbau und Analysen. Online. Version 5.

Lukas C. Faulstich, Ulf Leser, and Thorsten Vitt. 2006. Implementing a linguistic query language for historic texts. In *Current Trends in Database Technology – EDBT 2006*, pages 601–612, Berlin, Heidelberg. Springer Berlin Heidelberg.

Marcello Federico, Dimitri Giordani, and Paolo Coletti. 2000. Development and evaluation of an Italian broadcast news corpus. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000)*, Athens, Greece. European Language Resources Association (ELRA).

Elena Frick, Carsten Schnober, and Piotr Bański. 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. European Language Resources Association (ELRA).

Markus Gärtner. to appear. The corpus query middleware of tomorrow – A proposal for a hybrid corpus query architecture. In *Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-8)*.

Markus Gärtner, Anders Björkelund, Gregor Thiele, Wolfgang Seeker, and Jonas Kuhn. 2014. Visualization, search, and error analysis for coreference annotations. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 7–12, Baltimore, Maryland. Association for Computational Linguistics.

Markus Gärtner and Jonas Kuhn. 2018. Making corpus querying ready for the future: Challenges and concepts. In *Proceedings of the 27th International Conference on Computational Linguistics*, KONVENS 2018, Wien, Österreich.

Markus Gärtner, Katrin Schweitzer, Kerstin Eckart, and Jonas Kuhn. 2015. Multi-modal visualization and search for text and prosody annotations. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 25–30, Beijing, China. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.

Markus Gärtner, Gregor Thiele, Wolfgang Seeker, Anders Björkelund, and Jonas Kuhn. 2013. ICARUS – an extensible graphical search tool for dependency treebanks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Sofia, Bulgaria. Association for Computational Linguistics.

Sumukh Ghodke and Steven Bird. 2012. Fangorn: A system for querying very large treebanks. In *COLING 2012: Demonstration Papers*, pages 175–182, Mumbai, India.

Johannes Graën, Simon Clematide, and Martin Volk. 2016. Efficient exploration of translation variants in large multiparallel corpora using a relational database. In *4th Workshop on the Challenges in the Management of Large Corpora*, pages 20–23. s.n.

David Graff. 1995. European Language Newspaper Text LDC95T11. Web Download, Philadelphia: Linguistic Data Consortium.

David Graff and Christopher Cieri. 2003. English Gigaword LDC2003T05. Web Download, Philadelphia: Linguistic Data Consortium.

David Graff and Rebecca Finch. 1994. Multilingual text resources at the Linguistic Data Consortium. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Jan Hajič. 2006. Complex corpus annotation: The Prague Dependency Treebank. In *Insight into Slovak and Czech Corpus Linguistic*, pages 54–73, Bratislava, Slovakia. Jazykovedný ústav Ľ. Štúra, SAV.

Jan Hajič, Barbora Vidová Hladká, and Petr Pajas. 2001. The Prague Dependency Treebank: Annotation structure and support. In *Proceedings of the IRCS Workshop on Linguistic Databases*, pages 105–114. University of Pennsylvania, Philadelphia, USA.

Hans Van Halteren and Theo Van Den Heuvel. 1990. *Linguistic Exploitation of Syntactic Databases: The Use of the Nijmegen LDB Program (LANGUAGE AND COMPUTERS)*. Brill Rodopi.

Olivier Hamon, Andrei Popescu-Belis, Khalid Choukri, Marianne Dabbadie, Anthony Hartley, Widad Mustafa El Hadi, Martin Rajman, and Ismaïl Timimi. 2006. CESTA: First conclusions of the technolangue MT evaluation campaign. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Ulrich Heid and Andreas Mengel. 1999. Query language for research in phonetics. In *International Congress of Phonetic Sciences (ICPhS 99)*, pages 1225–1228, San Francisco.

Nancy Ide, Collin Baker, Christiane Fellbaum, and Rebecca Passonneau. 2010. The Manually Annotated Sub-Corpus: A community resource for and by the people. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 68–73, Uppsala, Sweden. Association for Computational Linguistics.

Radu Ion, Elena Irimia, Dan Ştefănescu, and Dan Tufiş. 2012. ROMBAC: The Romanian Balanced Annotated Corpus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Amy Isard, David McKelvie, Andreas Mengel, and Morten Baun Møller. 2000. The Mate Workbench - a tool for annotating XML corpora. In *Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications) - RIAO 2000, 6th International Conference, College de France, France, April 12-14, 2000. Proceedings*, pages 411–425.

Miloš Jakubíček, Adam Kilgarriff, Diana McCarthy, and Pavel Rychlý. 2010. Fast syntactic searching in very large corpora for many languages. In *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*, pages 741–747, Tohoku University, Sendai, Japan. Institute of Digital Enhancement of Cognitive Processing, Waseda University.

Daniel Janus and Adam Przepiórkowski. 2007. Poliqarp: An open source corpus indexer and search engine with syntactic extensions. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 85–88, Stroudsburg, PA, USA. Association for Computational Linguistics.

Matthias Jarke and Yannis Vassiliou. 1985. A framework for choosing a database query language. *ACM Comput. Surv.*, 17(3):313–340.

Esther Kaufmann and Abraham Bernstein. 2010. Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases. *Journal of Web Semantics*, 8(4):377–393.

Stephan Kepser. 2003. Finite structure query: A tool for querying syntactically annotated corpora. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 179–186, Stroudsburg, PA, USA. Association for Computational Linguistics.

Stephan Kepser. 2004. Querying linguistic treebanks with monadic second-order logic in linear time. *Journal of Logic, Language and Information*, 13(4):457–470.

Adam Kilgarriff, Vít Baisa, Jan Bušta, Miloš Jakubíček, Vojtěch Kovář, Jan Michelfeit, Pavel Rychlý, and Vít Suchomel. 2014. The Sketch Engine: ten years on. *Lexicography*, pages 7–36.

Adam Kilgarriff, Pavel Rychlý, Pavel Smrž, and David Tugwell. 2004. The Sketch Engine. *Information Technology*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand.

Esther König and Wolfgang Lezius. 2000. A description language for syntactically annotated corpora. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 2*, COLING '00, pages 1056–1060, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thomas Krause, Ulf Leser, and Anke Lüdeling. 2016. graphANNIS: A fast query engine for deeply annotated linguistic corpora. *JLCL*, 31(1):iii–25.

Thomas Krause and Amir Zeldes. 2014. ANNIS3: A new architecture for generic corpus query and visualization. *Digital Scholarship in the Humanities*.

Catherine Lai and Steven Bird. 2004. Querying and updating treebanks: A critical survey and requirements analysis. In *In Proceedings of the Australasian Language Technology Workshop*, pages 139–146.

Catherine Lai and Steven Bird. 2005. LPath+: A first-order complete language for linguistic tree query. In *Proceedings of the 19th Pacific Asia Conference on Language, Information and Computation*, pages 1–12, Taipei, Taiwan, R.O.C. Institute of Linguistics, Academia Sinica.

Catherine Lai and Steven Bird. 2010. Querying linguistic trees. *J. of Logic, Lang. and Inf.*, 19(1):53–73.

Wolfgang Lezius. 2002. *Ein Suchwerkzeug für syntaktisch annotierte Textkorpora*. Ph.D. thesis, IMS, University of Stuttgart. Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), volume 8, number 4.

Mark Liberman. 1989. Text on tap: The ACL/DCI. In *Speech and Natural Language: Proceedings of a Workshop Held at Cape Cod, Massachusetts, October 15-18, 1989*.

Joakim Lundborg, Torsten Marek, and Martin Volk. 2007. Using the Stockholm TreeAligner. In *6th Workshop on Treebanks and Linguistic Theories*.

Juhani Luotolahti, Jenna Kanerva, Sampo Pyysalo, and Filip Ginter. 2015. SETS: Scalable and efficient tree search in dependency graphs. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 51–55, Denver, Colorado. Association for Computational Linguistics.

Brian MacWhinney, Steven Bird, Christopher Cieri, and Craig Martell. 2004. Talkbank: Building an open unified multimodal database of communicative interaction. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Torsten Marek, Joakim Lundborg, and Martin Volk. 2008. Extending the TIGER query language with universal quantification. In *KONVENS 2008: 9. Konferenz zur Verarbeitung natürlicher Sprache*, pages 5–17.

Hendrik Maryns and Stephan Kepser. 2009a. MonaSearch – a tool for querying linguistic treebanks. In *Treebanks and Linguistic Theories 2009*, pages 29–40, Groningen.

Hendrik Maryns and Stephan Kepser. 2009b. Monasearch: Querying linguistic treebanks with monadic second-order logic. In *The 7th International Workshop on Treebanks and Linguistic Theories*.

Andreas Mengel. 1999. MATE deliverable D3. 1–specification of coding workbench: 3.8 improved query language (Q4M). Technical report, Technical report, Institut für Maschinelle Sprachverarbeitung, Stuttgart, 18 ....

Andreas Mengel, Ulrich Heid, Arne Fitschen, and Stefan Evert. 1999. Specification of coding workbench: Improved query language (q4m). Technical report, Technical Report MATE Deliverable.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41.

Jiří Mírovský. 2006. Netgraph: A tool for searching in Prague Dependency Treebank 2.0. In *Proceedings of TLT 2006*, pages 211–222, Praha, Czechia. ÚFAL MFF UK.

Jiří Mírovský. 2008. PDT 2.0 requirements on a query language. In *Proceedings of ACL-08: HLT*, pages 37–45, Columbus, Ohio. Association for Computational Linguistics.

Gregor Möhler. 2001. Improvements of the PaIntE model for F_0 parametrization. Technical report, Institute of Natural Language Processing, University of Stuttgart. Draft version.

Martin Mueller. 2010. Towards a digital carrel: A report about corpus query tools.

Gerald Nelson, Sean Wallis, and Bas Aarts. 2002. *Exploring natural language: working with the British component of the International Corpus of English*. John Benjamins Publishing.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Gertjan van Noord, Gosse Bouma, Frank Van Eynde, Daniël de Kok, Jelmer van der Linde, Ineke Schuurman, Erik Tjong Kim Sang, and Vincent Vandeghinste. 2013. Large scale syntactic annotation of written Dutch: Lassy. In Peter Spyns and Jan Odijk, editors, *Essential Speech and Language Technology for Dutch: Results by the STEVIN programme*, Theory and Applications of Natural Language Processing, pages 147–164. Springer, Berlin, Heidelberg.

Roman Ondruška, Jiří Mírovský, and Daniel Průša. 2002. Searching through Prague Dependency Treebank-conception and architecture. In *Proceedings of The First Workshop on Treebanks and Linguistic Theories*, pages 114–122, Sofia, Bulgaria and Tuebingen, Germany. LML, Bulgarian Academy of Sciences and SfS, Tuebingen University.

Petr Pajas. 2009. TrEd. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Petr Pajas and Jan Štěpánek. 2005. A generic XML-based format for structured linguistic annotation and its application to Prague DependencyTreebank 2.0. Technical Report 29, ÚFAL MFF UK, Prague, Czech Republic.

Petr Pajas and Jan Štěpánek. 2009. System for querying syntactically annotated corpora. In *ACL-IJCNLP: Software Demonstrations*, pages 33–36, Suntec, Singapore.

Ulrik Petersen. 2004. Emdros: A text database engine for analyzed or annotated text. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ulrik Petersen. 2005. Evaluating corpus query systems on functionality and speed: TIGERSearch and Emdros. In *International Conference Recent Advances in Natural Language Processing. Proceedings*, pages 387–391. Incoma, Ltd.

Ulrik Petersen. 2006a. Principles, implementation strategies, and evaluation of a corpus query system. In *Finite-State Methods and Natural Language Processing*, pages 215–226, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ulrik Petersen. 2006b. Querying both parallel and treebank corpora: Evaluation of a corpus query system. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Florian Petran, Marcel Bollmann, Stefanie Dipper, and Thomas Klein. 2016. ReM: A reference corpus of Middle High German - corpus compilation, annotation, and access. *Journal for Language Technology and Computational Linguistics*, 31(2):1–15.

Rashmi Prasad, Bonnie Webber, Alan Lee, and Aravind Joshi. 2019. Penn Discourse Treebank Version 3.0 LDC2019T05. Web Download, Philadelphia: Linguistic Data Consortium.

Adam Przepiórkowski. 2004. *The IPI PAN Corpus: Preliminary version*. Institute of Computer Science, Polish Academy of Sciences, Warsaw.

Adam Przepiórkowski, Zygmunt Krynicki, Łukasz Debowski, Marcin Woliński, Daniel Janus, and Piotr Bański. 2004. A search tool for corpora with positional tagsets and ambiguities. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, pages 1235–1238.

Beth Randall. 2008. CorpusSearch 2 users guide.

Philip Resnik and Aaron Elkiss. 2005. The Linguist's Search Engine: An overview. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 33–36, Ann Arbor, Michigan. Association for Computational Linguistics.

Sebastian Riedel. 2008. What's Wrong With My NLP? http://code.google.com/p/whatswrong/.

Douglas L.T. Rohde. 2001. TGrep2 user manual. http://tedlab.mit.edu/ dr/Tgrep2/.

Roland Schäfer. 2015. Processing and querying large web corpora with the COW14 architecture. In *Proceedings of Challenges in the Management of Large Corpora 3 (CMLC-3)*, Lancaster. UCREL, IDS.

Roland Schäfer and Felix Bildhauer. 2012. Building large corpora from the web using a new efficient tool chain. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul. European Language Resources Association (ELRA).

Katrin Schweitzer, Kerstin Eckart, Markus Gärtner, Agnieszka Falenska, Arndt Riester, Ina Rösiger, Antje Schweitzer, Sabrina Stehwien, and Jonas Kuhn. 2018. German radio interviews: The GRAIN release of the SFB732 Silver Standard Collection. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ilona Steiner and Laura Kallmeyer. 2002. VIQTORYA – a visual query tool for syntactically annotated corpora. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Jan Štěpánek and Petr Pajas. 2010. Querying diverse treebanks in a uniform way. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Ann Taylor. 2003. CorpusSearch version 1.1 - reference manual.

Gregor Thiele, Wolfgang Seeker, Markus Gärtner, Anders Björkelund, and Jonas Kuhn. 2014. A graphical interface for automatic error mining in corpora. In *Proceedings of the Demonstrations at the 14th*

Conference of the European Chapter of the Association for Computational Linguistics, pages 57–60, Gothenburg, Sweden. Association for Computational Linguistics.

Sean Wallis and Gerald Nelson. 2000. Exploiting fuzzy tree fragment queries in the investigation of parsed corpora. *Literary and linguistic computing*, 15(3):339–362.

Amir Zeldes, Anke Lüdeling, Julia Ritz, and Christian Chiarcos. 2009. ANNIS: a search tool for multi-layer annotated corpora. In *Proceedings of Corpus Linguistics*.