Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 4011–4021
Marseille, 20-25 June 2022

Querying a Dozen Corpora and a Thousand Years with Fintan

Christian Chiarcos[+*], Christian Fäth[+], Maxim Ionov[+*]
[+] Applied Computational Linguistics (ACoLi), Goethe University Frankfurt, Germany
[*] Institute for Digital Humanities (IDH), University of Cologne, Germany
{chiarcos|faeth|ionov}@cs.uni-frankfurt.de

Abstract

Large-scale diachronic corpus studies covering longer time periods are difficult if more than one corpus are to be consulted and, as a result, different formats and annotation schemas need to be processed and queried in a uniform, comparable and replicable manner. We describes the application of the Flexible Integrated Transformation and Annotation eNgineering (Fintan) platform for studying word order in German using syntactically annotated corpora that represent its entire written history. Focusing on nominal dative and accusative arguments, this study hints at two major phases in the development of scrambling in modern German. Against more recent assumptions, it supports the traditional view that word order flexibility decreased over time, but it also indicates that this was a relatively sharp transition in Early New High German.
The successful case study demonstrates the potential of Fintan and the underlying LLOD technology for historical linguistics, linguistic typology and corpus linguistics. The technological contribution of this paper is to demonstrate the applicability of Fintan for *querying* across heterogeneously annotated corpora, as previously, it had only been applied for *transformation* tasks. With its focus on quantitative analysis, Fintan is a natural complement for existing multi-layer technologies that focus on query and exploration.

Keywords: interoperability, corpus querying, German, word order, Linguistic Linked Open Data (LLOD) technology

1. Introduction

Large-scale quantitative diachronic corpus studies covering long time periods are relatively difficult if multiple corpora need to be queried, since, typically, each of these comes with distinct characteristics in format, data model, and annotation schema. Multi-layer corpus technology can solve the problem (if the corpora are transformed into a common representation), but it requires the user to run queries manually. In studies with a large number of parameters over a large number of corpora, this quickly becomes infeasible.

We illustrate this problem with a study of diachronic word order in German, addressing ordering preferences for nominal accusative and dative arguments in postverbal position, as this has been a matter of longstanding interest in NLP (Rambow, 1993; Strube and Hahn, 1999; Wunsch, 2006) and linguistics (Speyer, 2007; Abraham, 2007; Vinckel-Roisin, 2011; Molnár and Vinckel-Roisin, 2019). In modern German, both orders are acceptable, but in corpora we find a preference of dative arguments to precede accusative arguments. In the literature, there is some debate about whether German word order was more restrictive in the past (Speyer, 2011; Speyer, 2013) or whether the flexible word order is part of its common West Germanic heritage (the traditional assumption), but until recently, it had been impossible to explore this with quantitative, corpus-linguistic methods, as syntactically annotated data was only available for selected periods of time. In December 2021, the long-awaited Referenzkorpus Frühneuhochdeutsch (Wegera et al., 2021, ReF) has been released, it provides a syntactically annotated sub-corpus of 600.000 words for the period from 1350-1650 and thus closes the most pressing gap for the study of historical syntax in German. Together with the Middle High German Treebank (Chiarcos et al., 2018), which provides a layer of automated syntax annotation (Chiarcos et al., 2018b) over the Referenzkorpus Mittelhochdeutsch (Petran et al., 2016, ReM) for the 11th to 14th centuries (Klein et al., 2016), the GerManC corpus of syntactic dependencies (Durrell et al., 2012) (Scheible et al., 2011) (17th to 18th c.) and a number of smaller-scale corpora, it has thus become possible to study diachronic developments of syntactic phenomena in German through the entire course of the past 1000 years. However, the large number and diverse nature of the corpora available posits a number of technical challenges, as the different corpora follow different traditions in linguistic annotation, representing all major paradigms of syntactic annotation, i.e., dependency syntax (GerManC), phrase structure syntax of different flavours (ReF, ReM) and span-based annotation (corpora of Old High German, and a number of smaller-scale corpora created with annotation tools focusing on tiers rather than phrases).

We address the challenge for the conjoint, comparable and reproducible evaluation of a potentially vast number of parameter combinations over such heterogeneous data by converting the original data to graphs and then transforming these graphs to conform to a coherent data model. To this representation, then, queries are applied and matches are retrieved. For this purpose, we describes the application of Fintan, the Flexible Integrated Transformation and Annotation eNingeering platform (Fäth et al., 2020) to perform queries over a large number of corpora for historical German and re-

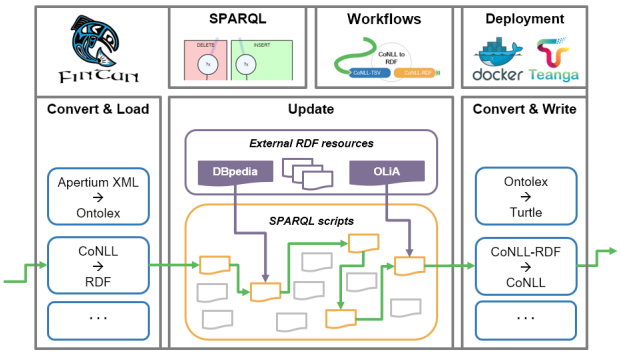

Figure 1: Fintan overview

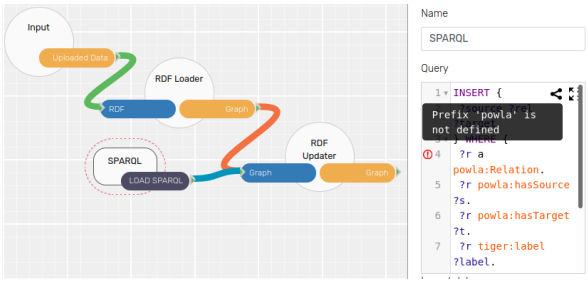

Figure 2: Editing workflows with the Fintan Frontend

lated language varieties, in diverse formats and coming from different theoretical backgrounds. The technological contribution of this paper is to demonstrate the applicability of Fintan for corpus-linguistic purposes, and in particular, for *querying* across heterogeneously annotated corpora, as previously, it had only been applied for *transformation* tasks in NLP and linking workflows. With Fintan, it is possible to apply these transformation and query operations directly on the original source data, and its parallelized and streamable architecture guarantees efficient performance. With its focus on quantitative analysis, Fintan is a natural complement for existing multi-layer technologies that focus on query and exploration, but which are more tailored towards visualization and exploration of query results rather than for quantitative analysis.

2. The Fintan Platform

The Fintan platform,[1] as shown in Fig. 1, alleviates the creation of complex transformation workflows for various input and output formats by wrapping existing converter frameworks and combining them with a stream-based graph processing engine. This encompasses:

- An interoperable pool of workflow *components* including: (a) external converter tools, (b) stream-based graph processing for RDF (c) serializer tools and data writers
- A graphical Frontend for developing SPARQL[2] updates and transformation workflows
- A means of deploying specific converter pipelines as integrated Dockercontainers

Internally, Fintan parses various source data formats into a series of RDF graphs to apply graph transformation on the resulting data stream. This transformation is parallelized and thus highly **scalable**. Esp. with existing open source frameworks for RDF processing, such as Apache Jena[3], query execution tends to increase in complexity regarding both memory consumption and execution time when processing larger datasets. Data segmentation is one feasible way to counteract this and is largely applicable to linguistic resources. For a corpus normally each segment represents one sentence at a time, optionally also a window of preceding and following sentences. For dictionaries, segments could be lexical entries or translation sets.

An earlier version of the technology underlying Fintan provided coverage for tabular formats such as the popular CoNLL format family (Chiarcos and Fäth, 2017). In Fintan, this is generalized to arbitrary formats for corpora and machine-readable dictionaries using a **modular** approach to data conversion: types of data transformation are divided into distinct workflow *components* which can be instanced and configured to perform specific transformation tasks. *Transformer* and *Loader* components hereby prepare and load the data for segmented processing while *Updaters* execute the graph transformation and *Writers* are being provided for serializing the resulting datasets in different RDF serializations, various corpus formats, graphical visualizations or result tables. Most components can be configured by scripts. While for the core components updates and queries are implemented in SPARQL, a W3C-standardized query language for RDF, Transformers, Loaders and Writers may additionally wrap an existing converter suite such as Saxon[4], e.g. for executing XSLT scripts.

The modular approach also alleviates **reusable** pipeline design since generic components supporting a multitude of input and output formats can be instanciated to granular transformation steps defined by scripts: e.g. if both, an instance of an XSL transformer, as well as an instance of a CSV transformer could convert a dictionary to OntoLex-Lemon[5], a subsequent change of annotation models could be performed by the same instance of an RDF updater.

Modularity is also reflected in the program architecture. The *Fintan Core API* is implemented in Java as the primary execution layer. It only provides a minimal amount of dependencies for stream-based graph and RDF processing and can be imported into existing

[1] https://github.com/Pret-a-LLOD/Fintan
[2] https://www.w3.org/TR/sparql11-query/
[3] https://jena.apache.org/
[4] https://www.nuget.org/packages/Saxon-HE
[5] OntoLex-Lemon is a widely used community standard for publishing lexical resources: https://www.w3.org/2016/05/ontolex/

projects as a Maven[6] artifact. In addition, it is designed to host and run custom-made components, thus making the whole system **extensible**. The full *Fintan Backend*, wraps the Core API together with other compliant tranformation components, e.g. for TBX terminologies, CoNLL corpora or generic XML. It can directly be executed as a stand-alone command line tool or used with the *Graphical Frontend* to build and export pipelines as dockerized *services* which can be deployed to the Teanga[7] platform, thus enabling RDF-based NLP modules to directly feed on generic resource types, further increasing scope and applicability for long-term use by a wider audience.

3. Data and Data Modelling

In order to perform querying across the heterogeneous collection of corpora addressed in our study, the existing data structures need to be harmonized into coherent representations. For this purpose, we employ RDF graphs, following the insight that a directed (acyclic) graph can represent literally any kind of linguistic annotation (Bird and Liberman, 2001), an assumption also underlying the relevant ISO standards (ISO, 2012; ISO, 2014). However, we do normalize the *content* of the annotations against a standard vocabulary, but only the structures, i.e., phrases, hierarchies and dependencies with labelled or unlabelled edges. As a result, the data remains faithful to its original content, but equivalent data structures can be queried analoguously. As the different corpora are represented in RDF, we can use the RDF standard query language SPARQL for retrieval, with adjustments for the labels and tags used in the respective corpora.

3.1. Corpora

For this study, we consult *all* syntactically annotated corpora of historical German that we are aware of and that comprise at least 5.000 tokens for at least one 50-year period, see Tab. 1. In addition to material on historical and modern High German, we also consulted syntactically annotated corpora from Old Saxon and Old English, as these are closely related to Old High German (for which we have less than 10 examples of nominal dative and accusative arguments in our data).
In terms of structure and annotation paradigm, the corpora considered here fall into three major groups:

dependency annotation in different CoNLL-TSV formats;

phrase structure annotations as developed in the traditions of the Penn Treebank (Taylor et al., 2003, PTB), as well as those of the German NEGRA/TIGER (Brants et al., 2003; Brants et al., 2004) and TüBa-D/Z (Telljohann et al., 2004) corpora;

span-based annotations as produced by tools like Exmaralda (Schmidt and Wörner, 2014) and ELAN (Sloetjes, 2017).

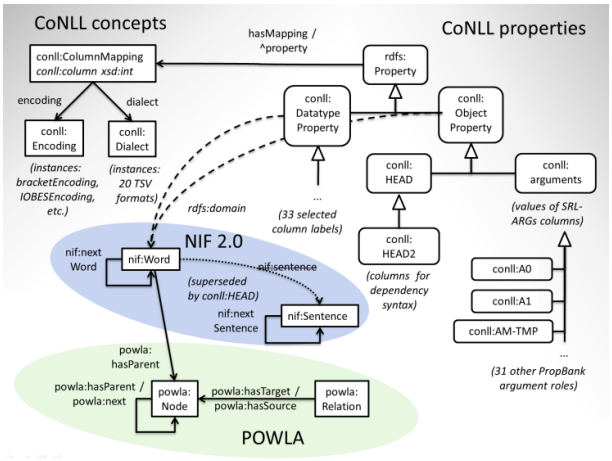

Figure 3: CoNLL-RDF vocabulary

3.2. Modelling Syntax in RDF

Fintan is designed (albeit not limited) to use RDF as data model for its internal information exchange, but it is not limited to any specific data model or schema. In particular, individual loaders may produce RDF representations that mimick the structure of the original data rather than adhere to a consistent model. In order to develop broadly applicable queries, their data structures need to be aligned. This transformation is implemented here by SPARQL update operations over the original sentence graphs.
It is important to note that Fintan stays fully agnostic about data modelling, but that any RDF vocabulary can be used. For processing corpus data, we normalize the different data sources into a consistent representation in accordance with the CoNLL-RDF vocabulary (Chiarcos et al., 2021) as summarized in Fig. 3. CoNLL-RDF provides data structures for linguistic annotation normally provided in tabular 'CoNLL' formats as used in long-standing series of Shared Tasks (CoNLL, since 1999),[8] and subsequent community efforts to create cross-linguistically comparable annotations (McCarthy et al., 2020; Akbik et al., 2015; Kyjánek et al., 2020)), but also in corpus linguistics (Kilgarriff et al., 2014; Hardie, 2012). CoNLL-RDF was originally conceived as part of a set of APIs for processing such data and for enabling round-tripping between RDF and CoNLL formats that allows for seamlessly integrating conventional NLP pipelines and Linked Data technologies (Chiarcos and Schenk, 2018). It consists of four main

[6] https://maven.apache.org/

[7] Teanga is a workflow management software for integrated, dockerized NLP and Linked Data services developed in the Prêt-à-LLOD project initially presented by Ziad et al. (2018): https://github.com/Pret-a-LLOD/teanga

[8] https://www.conll.org/

corpus	tokens	language/ period	genre	annotation type	annotation e.labels	format	renaming/ restructuring
YCOE (Taylor et al., 2003)	1.5 m	Old English	balanced	phrase	yes	PTB	yes/no
ISWOC (Eckhoff et al., 2018)	28 k	Old English	balanced	dep	yes	CoNLL	no
HeliPaD (Walkden, 2016)	50 k	Old Saxon	poetry	phrase	yes	PTB	yes/no
TCodex (Petrova et al., 2011)	6 k	9th c.	religious	span	no	Exmaralda	yes
ReM (Klein et al., 2016; Chiarcos et al., 2018b)	2.2 m	11th - 14th c.	balanced	phrase	no	CoNLL	yes/no
ReF (Wegera et al., 2021)	600 k	14th - 17th c.	balanced	phrase	yes	TIGER	no
ENHG (Light, 2012)	200 k	16th c.	religious	phrase	yes	PTB	yes/no
Mercurius (Demske, 2003)	200 k	16th - 17th c.	news	phrase	yes	TIGER	no
Fuerstinnenkorrespondenz (Lühr et al., 2014)	260 k	16th - 18th c.	letters	span	no	Exmaralda	yes
GerManC (Scheible et al., 2011)	770 k	17th - 18th c.	balanced	dep	yes	CoNLL	no
UD-LIT (De Marneffe et al., 2021)	40 k	18th c.	poetry	dep	yes	CoNLL-U	no
TüBa-D/Z (Telljohann et al., 2004)	1.4 m	modern	news	phrase	yes	TIGER	no
UD-HDT (De Marneffe et al., 2021)	3.8 m	modern	news	dep	yes	CoNLL-U	no
UD-GSD (De Marneffe et al., 2021)	300 k	modern	news/web	dep	yes	CoNLL-U	no

Table 1: Corpora consulted and their features

components associated with the namespaces `nif:`, `powla:`, `conll:`, and `tiger:`.

`nif:` core data structures The core of CoNLL-RDF is constituted by a minimal subset of the NLP Interchange Format (Hellmann et al., 2013, NIF 2.0), i.e., `nif:Word` and `nif:Sentence`, which are sequentially arranged by means of `nif:nextWord` and `nif:nextSentence`, respectively.

`powla:` phrases and relations Linguistic annotations for non-consecutive segments (e.g., zero morphemes or discontinuous phrases) are annotated using a minimal subset of POWLA (Chiarcos, 2012), an OWL2/DL implementation of an early version of the Linguistic Annotation Framework (Ide and Suderman, 2014), i.e., `powla:Node` (for markables, e.g., phrases), `powla:hasParent` (for structural relations between nodes in a parse tree), `powla:next` (to connect sibling nodes in a parse tree), and `powla:Relation` (for labelled edges connecting two nodes with `powla:hasTarget` and `powla:hasSource`).

`conll:` token-level annotations For every column label in a CoNLL format, a property of the same name is created that contains the annotation as string value. Some columns receive special treatment, e.g. `HEAD` column and the associated `onto:HEAD` property, as this is used to *either* represent dependency relations, or connect every word with the sentence (in the absence of dependency annotations). The list of CoNLL datatype properties is otherwise extensible.

`tiger:` phrase and edge labels For phrase and edge labels, we adopt the naming conventions of TIGER-XML (Mengel and Lezius, 2000) as this is is the native format of several corpora in our sample. For the annotation of phrase categories, we use `tiger:cat`, for edge labels, we use `tiger:label`. Without a formal namespace for TIGER-XML not any web-accessible XML schema, we resort to the URL of the page that currently provides reference information about the vocabulary.[9] The difference between edges and secondary edges is not encoded as an explicit data structure. Instead, a `powla:Relation` represents an edge if it is accompanied by a `powla:hasParent` property, a secondary edge otherwise. Note that token-level annotations are encoded with analoguous CoNLL properties, to which conventional TIGER-XML attributes like `@word`, `@id`, `@pos`, `@lemma` and `@morph` are mapped.

In comparison to the state of the art in the RDF-based processing of syntax annotations, the TIGER vocabulary is an innovation. It was extended here to all phenomena of phrase structure syntax not covered by POWLA, CoNLL-RDF or NIF, i.e., as a generic namespace for resource-specific attributes and labels, regardless of whether the original resource was provided as TIGER XML or another phrase structure format. However, using a URI designating the original TIGER XML format to define the namespace is a provisional solution, only. In the future, all relevant namespaces should be unified and aligned into a single, coherent data model.[10]

[9] `https://www.ims.uni-stuttgart.de/documents/ressourcen/werkzeuge/tigersearch/doc/html/TigerXML.html#`

[10] For initial steps towards this goal, see on-going discussions on harmonizing web standards for linguistic annotations at the W3C Community Group Linked Data for Language Technology, LD4LT, `https://www.w3.org/community/ld4lt/`. We expect our vocabulary to serve as input to these discussions.

4. Data Transformation

4.1. Fintan Workflows

We use the Fintan frontend and its integrated SPARQL editor with highlighting and syntax check to design the workflows described here, see Fig. 2 for an example. In Fintan workflows, modules exchange information in either serialized formats (an RDF serialization or otherwise) or as an object stream of RDF graph objects. The Frontend visualizes the different data streams in different colors, so that format mismatches can be avoided.

As our corpus queries are applied in batch, they are run against the backend using scripts, in this case aggregated into a general `Makefile` that defines different goals (output files) for different corpora. The main processing steps are

Makefile preprocessing retrieval of data and installation of dependencies

Fintan loaders conversion of source data to RDF (sometimes with preprocessing in the `Makefile`)

Fintan updater: SPARQL transformation of raw RDF graphs into a consistent representation in accordance with the tree extension

Fintan Formatter: SPARQL querying of resulting data and export as TSV table

Quantitative evaluation quantitative evaluation is performed using external tools

Conversion to RDF is handled by format-specific loaders, further transformations via the CoNLL-RDF Updater, and querying and generation of result table by the CoNLL-RDF Formatter, all integrated in the Fintan Backend. In addition, the `Makefile` retrieves the relevant corpora and installs the dependencies (Fintan/CoNLL-RDF, CoNLL-Merge, LLODifier). Note that some corpora cannot be freely distributed, so that local building is required. Also, providing a build script that operates from the source formats instead of readily transformed data allows to take future updates into account.

Overall, we provide transformation and retrieval scripts for 8 corpora of historical German, 3 corpora of modern German, two corpora of Old English and one corpus of Old Saxon as summarized in Tab. 1.[11]

[11]From the eventual analysis, we excluded the Early New High German Fuerstinnenkorrespondenz (Lühr et al., 2014) because its annotations were found to be incomplete (only 35% [92.000 of 260.000 tokens] annotated with clause structure). More importantly, it deals with a highly specific genre (mostly letters containing or responding to requests for financial support) that is not directly comparable with the predominantly narrative texts from the other corpora.

4.2. RDF Conversion and Consolidation

For our corpora, existing loaders for phrase structure annotations in TIGER-XML (ReM, ReF; a designated loader for XML data) and for dependency annotation in CoNLL (GerManC; the CoNLL-RDF loader) directly produce RDF data compliant with this data model. This data requires no further preprocessing.

Other corpora require considerable restructuring. This includes corpora developed in the Penn Treebank (PTB) tradition (Early New High German corpus by Caitlin Light, Old Saxon HeliPaD), as well as the loader for span-based annotations which combines a converter from the Exmaralda format (Chiarcos and Schenk, 2018) to CoNLL. Both types of corpora are first transformed to a CoNLL representation of parse trees and then processed by the CoNLL-RDF loader.

Some corpora require additional preprocessing steps after RDF conversion. This conversion is provided by SPARQL Update scripts that perform simple replacement operations to conform to the overall data model. In particular, span-based annotations require substantial restructuring *after* RDF conversion. These represent phrases as a flat annotation with separate tiers for phrase spans, grammatical function spans and clause spans that each link each span with offsets in the text, but not with other tiers, so that nesting into a hierarchical structure needs to be asserted with SPARQL Update:

```
DELETE { ?w powla:hasParent ?gf, ?cl. }
INSERT {
 ?phrase powla:hasParent ?gf.
 ?gf     powla:hasParent ?cl.
} WHERE { # for all spans over a word
 ?w powla:hasParent ?phrase, ?gf, ?cl.
 ?phrase a conll:CAT.    # define the
 ?gf     a conll:GF.     # respective
 ?cl     a conll:CLAUSE. # tier
};
```

The resulting tree structure is further processed and queried analogously to phrase structure syntax.

Frequently, preprocessing addresses naming conventions, e.g., phrase structure annotations converted with the CoNLL-RDF loader follow the naming conventions of the CoNLL tree extensions (Chiarcos and Glaser, 2020) rather than the TIGER vocabulary: While it creates the correct CoNLL and POWLA data structures, its properties `rdf:value` and `rdfs:label` that are replaced by `tiger:label` and `tiger:cat`.

The resulting representation conforms to the vocabulary delineated above, and with SPARQL, it is now possible to query all corpora in a uniform way.

5. Querying

Within Fintan, SPARQL queries can be applied after loading, and, optionally, transforming a resource, e.g., as parameters of Fintan 'formatters' that create tabular data. When editing workflows with the Fintan frontend,

the addition of a query parameter will open a SPARQL editor window that features syntax highlighting, validation. (Likewise, it can also be used for creating and editing update scripts at transformation modules.)
The eventual queries follow a uniform structure, although with adjustments for differences in annotation scheme (different labels; explicit encoding of grammatical roles in edge labels or nodes or implicitly via case morphology) and theoretical framework (i.e., as phrase structure, phrase structures with labelled edges or dependency syntax with labelled edges).

5.1. Optimization for Labelled Edges

For corpora with labelled edges, we apply an additional update in order to facilitate querying of transitive edges for phrase structure annotations with edge labels. Thanks to the preceding harmonization, the same update can be applied to all corpora: Labelled edges (`powla:Relations` with `tiger:label`) are precompiled into object properties, so, that for the value of `tiger:label`, say `OA`, we create the corresponding property, in this case `tiger:OA` between the original source and target. In SPARQL, this replacement is implemented with the following update:

```
INSERT {  ?source ?rel ?target }
WHERE {
 ?r a powla:Relation.
 ?r powla:hasSource ?s.
 ?r powla:hasTarget ?t.
 ?r tiger:label ?label.
 BIND(URI(concat("http://...#",?label))
      AS ?rel) }
```

The concatenation operation forms a new URI (relation) by concatenating the URL of the TIGER namespace with the edge label. These shortcuts are particularly useful for complex queries as, now, complex property paths can be defined over labelled edges. In an example, it is now possible, for example, to perform transitive search:

```
SELECT ?src ?selfOrHd
WHERE { ?src tiger:HD* ?selfOrHd }
```

5.2. SELECT Queries

Even though a common data model is applied over the different corpora, differences in structure and annotation schemas require corpus-specific adaptations, so that each corpus requires (sometimes slightly) adjusted queries. At the same time, some of the queries are relatively complex, e.g., where the syntactic head is not explicitly annotated, we need to filter out non-matches. SPARQL provides the necessary operations for conjunction (default), disjunction (`UNION`), negation (`MINUS`), optional matches (`OPTIONAL`), a broad band-width of filters (`contains`, `regex`, `strends`, etc.), an explicit mechanism for handling the scope of these operators ({...}). Aside from directly addressing individual RDF statements (relations, triples), SPARQL also features property paths, which are a compact way of querying over complex series of relations:

```
?a nif:nextWord ?b.            # next word
?a ^nif:nextWord ?b.           # last word
?a nif:nextWord+ ?b.    # a following word
?a nif:nextWord* ?b.  # following or same
?a (^nif:nextWord)+ ?b.        # preceding
?a nif:nextWord/powla:hasParent ?b.
                     # parent of next word
```

Unlike most corpus query languages we are aware of, SPARQL requires an explicit `SELECT` statement to declare which variables are to be returned as results and which only serve an auxiliary function. This is a key requirement for quantitative studies, as it allows to deduplicate results. In SPARQL, deduplication is not an automated process, but must be triggered by an explicit (`SELECT DISTINCT`) statement. A downside in comparison to most corpus query languages is that deduplication is performed over the complete result set, so that for queries with a long runtime, no preliminary results can be returned. In consequence, we do not use SPARQL to query a large-scale datadump and to perform the quantitative analysis directly, but instead, we use it for generating result tables over which such statistical evaluations can then be performed with dedicated software. The generation of result tables is analoguous to the streamable update mechanism we used for transformations, i.e., each sentence (plus, optionally, its context) is queried in isolation and results are then concatenated. When processing large-scale corpora, this allows us to retrieve and inspect results early on, even before the entire corpus has been processed.

5.3. Corpus Querying with SPARQL

As shown above, querying can involve multiple SPARQL scripts for the corpora studied here: In addition to an obligatory `SELECT` query that returns a table of tab-separated values with results, we can provide optional update (`INSERT`/`DELETE`) operations for annotation preprocessing. The execution of SPARQL scripts is off-the-shelf RDF technology, so this can be executed on any triple store and in any programming language. However, with Fintan we can stream-process multiple sentences of very large corpora in parallel.
As Fintan extends SPARQL with the capability to perform loops and iterations, it is effectively a Turing-complete programming language (Hogan et al., 2020), so it exceeds conventional corpus query languages both in expressivity and in complexity. For quantitative evaluation, however, it brings a number of important improvements over corpus query languages. For comparison, one may think of ANNIS QL as implemented in ANNIS3 (Krause and Zeldes, 2016). As a directed acyclic graph, the data model of ANNIS is conceptually comparable to that of RDF, so, in principle, it supports equivalent means for graph traversal and matching. ANNIS does not have an equivalent of property

paths which can combine different properties in the graph by means of logical operators, but it supports a transitive closure over a pre-defined set of relations. More important, however, is that ANNIS QL does not distinguish result and auxiliary variables, so that every binding of an auxiliary variable will also produce a new result. This is probably less essential for corpus querying, but it massively skews quantitative analyses. Furthermore, ANNIS QL can only return direct matches from a corpus. SPARQL on the other hand can flexibly bind values to new variables, and that these values can be created by applying a broad number of transformations, including string and mathematical operations. In this way, return values of a `SELECT` query are not limited to corpus matches, but they can also be information derived from matches, e.g., counts or aggregates.

Fintan does not require any specific extensions to run SPARQL SELECT queries, as these have been part of the SPARQL protocol that we already support for updating RDF graphs. These queries can also be run with any RDF data base, but a specific benefit of using Fintan is that updates and queries can be applied to the original corpus data, and if this is not provided as RDF data, the native format can be converted on-the-fly. This is relatively performant due to parallelization and stream processing, it does quickly produce initial results that can be used for debugging and tuning queries and transformations.

One disadvantage of corpus querying with Fintan is that it does currently not provide graphical visualizations of query results and that its query language is somewhat more complex than conventional corpus query languages such as ANNIS QL. However, both aspects are already being addressed: On the one hand, we have recently integrated the Salt'n Pepper converter suite (Druskat et al., 2016) into Fintan (Fäth and Chiarcos, 2022), so that it can consume and produce ANNIS-compliant data, e.g., in order to benefit from existing visualizations of query results via ANNIS. On the other hand, we have developed a query engine over CoNLL-RDF data that precompiles a conventional corpus query language (CQP) into SPARQL (Ionov et al., 2020). Both solutions, however, are not readily applicable to the syntactic annotations considered here. The Salt'n'Pepper integration is a prototype only, and the CQP query language does not support querying recursive data structures as required for syntax.

6. Diachronic Word Order in German

For querying the corpora listed above, we provide five SPARQL updates (for preprocessing) and 11 SPARQL SELECT queries. While preprocessing scripts can be re-used for related formats, normally each corpus requires a slightly adjusted query to account for different schemas and modelling decisions. In this regard, it does not differ from other generic corpus query languages such as ANNIS QL (Krause and Zeldes, 2016). In a corpus management system, these queries normally have to be entered by hand, but Fintan provides a way to create extensive long batches of query (and transformation) scripts over a large set of corpora, so that different parameters can be explored in a non-interactive way.

We use the Fintan frontend to design workflows and to edit SPARQL updates and queries. These workflows are executed by the Fintan backend. However, as a large number of workflows is to be run, workflow execution is wrapped into a Makefile that retrieves the source data, optionally performs some minor preprocessing tasks and then calls the Fintan backend for generating result tables.

The result tables are uniform TSV files with 9 columns:
sentence URI link to the sentence in the corpus
verb form (or lemma) of the verbal predicate
context URI match identifier (for de-duplication)
argument order 'DAT>ACC' or 'ACC>DAT'
ACC determiner, its part of speech
ACC part of speech of head word
DAT determiner/ DAT POS
clause type: main/dependent

The aggregated results, limited to nominal arguments (proper nouns and common nouns, excluding pronouns) are summarized in Fig. 4 and Tab. 2. The full result tables can be reproduced using the Makefile provided in our GitHub repository.[12] Aside from absolute frequencies, Tab. 2 provides the significance for the transitions between individual time periods (the p numbers in the last row), using windows of 100 years. Except for the transition between 1300 and 1400 (the onset of Early New High German), no significant differences can be found (G^2, with $p \geq 0.05$ n.s., $p < 0.05$ (*), $p < 0.01$ *, $p < 0.001$ **, $p < 0.0001$ ***). However, for times after the 14th c., this may also reflect data sparsity issues. In fact, we observe an increase of word order flexibility around 1500, which may correspond to Speyer's original findings (even though he located their peak to be somewhat later).[13]

7. Discussion

We describe the application of Fintan, the Flexible Integrated Transformation and Annotation eNgineering platform to the study of scrambling in historical German and over more than a dozen syntactically annotated corpora that represent its entire written history. For a broad range of source formats, Fintan provides on-the-fly conversion to graphs, the refinement of these

[12] `https://github.com/acoli-repo/germhist/tree/master/analyses/scrambling`.

[13] In fact, this 'peak' in word order flexibility during the Early New High German period is the only period after 1400 that shows a significantly different (larger) number of ACC>DAT than the modern distribution. This is quite the oppositve of Speyer's view who assumed that scrambling evolved during Early Modern High German from a restrictive DAT>ACC order to the degree of relative freedom observed in modern data.

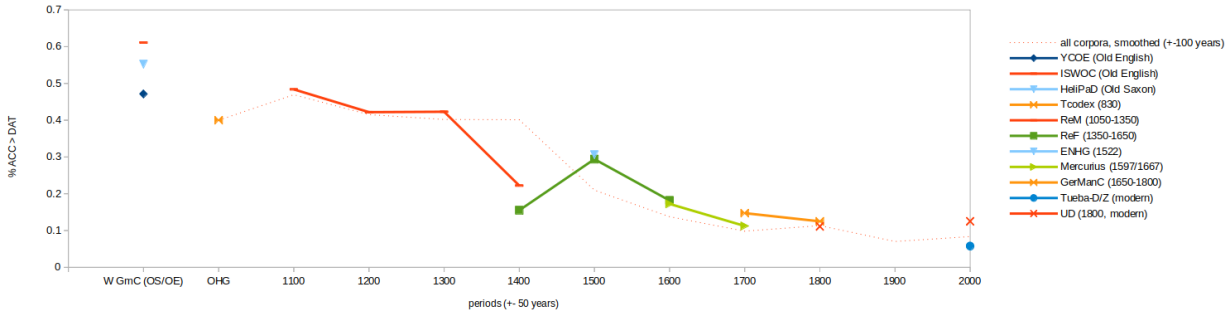

Figure 4: Word order of post-verbal nominals. Accusative and dative arguments in historical German

	ACC>DAT	DAT>ACC	p
OHG,OE,OS	633	694	
1100±50	75	80	n.s.
1200±50	281	385	n.s.
1300±50	481	656	n.s.
1400±50	21	91	***
1500±50	35	82	n.s.
1600±50	11	51	n.s.
1700±50	23	81	n.s.
1800±50	7	50	n.s.
2000±50	26	311	n.s.

Table 2: Aggregate counts for individual periods

graphs towards a uniform internal representation, and eventual query and the retrieval of the result set as a table for subsequent statistical evaluation.

The benefits of SPARQL have been emphasized before, both for querying (Burchardt et al., 2008; Chiarcos, 2012; Ionov et al., 2020) and for resource creation (Mazziotta, 2010; Chiarcos et al., 2018a; Chiarcos and Fäth, 2019),[14] Fintan provides an environment in which such workflows can be developed and replicated more easily. The Fintan frontend provides a graphical user interface for workflow design and management, as well as an interactive editor for SPARQL. The resulting workflows were, however, not directly executed by the Fintan backend, but manually compiled into a Makefile that binds them together with data retrieval, so that they can be more easily replicated.

The closest piece of related research is in multilayer corpus technologies. ANNIS (Krause and Zeldes, 2016) lacks aggregate functions and does not allow to distinguish return values from filter conditions. A notable example is the retrieval of accusative NPs: For the statistical evaluation, we need to bind the NP exactly once, but if we need to inspect the nominal head to determine the morphological case, the head is not explicitly marked in the annotation (as in TIGER and PTB schemes and in span-based annotations), and multiple candidates for the head exist (e.g., accusative nouns and pronouns in an NP), an ANNIS QL query will return one variable binding for every head candidate, and those duplicates need to be filtered out manually. In other words, ANNIS QL is not a suitable query language for statistical evaluation of phrase structure corpora. Nevertheless, it is possible to use ANNIS to produce result lists, and then to process these via a script. This is basically the same approach as taken here, but a major difference is that our technologies allow to perform and replicate the entire search and retrieval in a compact script and that manual search and export are not necessary.

This makes our approach comparable with generic converter pipelines,[15] as provided, for example, with the converter framework Pepper, also developed in conjunction with ANNIS (Druskat et al., 2016). A difference is that the integration of an new dataset into ANNIS is a considerable effort, as not only the data has to be converted, but also, it requires the user to set up an ANNIS instance and to configure its visualization components (as ANNIS provides different, domain-specific visualizations over its internal data model). Running Fintan transformation and querying workflows, however, only requires Docker or a command-line interface in a Unix-style environment.

Using this technology for studying ordering preferences of post-verbal nominal dative and accusative arguments in historical German, our results point to the existence two major phases in the development of scrambling in modern German. Against more recent assumptions (Speyer, 2011), this supports the traditional view that word order flexibility decreased over time, but it also indicates that this was not a gradual process, but a relatively sharp transition that occurred in the 14th to 15th centuries. From this time onward on no significant differences from modern word order can be observed.

[14] Note that this includes the Middle High German Treebank (Chiarcos et al., 2018), annotated with a a rule-based parser natively written in SPARQL.

[15] We focus on converter and transformation pipelines rather than NLP workflow management systems, which also include converter capabilities for their respective internal data model, but which are primarily designed for technical applications rather than to facilitate corpus querying. In particular, to the best of our knowledge, none of these support a scripting environment that provides the user with a declarative language for transforming annotations. However, Fintan could be integrated into such workflows, and, indeed, this has been implemented for the Teanga NLP workflow management system (Ziad et al., 2018).

8. Acknowledgements

The research described in this paper has been conducted by the Applied Computational Linguistics (ACoLi) lab at Goethe University Frankfurt, partially in the context of the BMBF Early Career Research Group 'Linked Open Dictionaries (LiODi)', and partially in the context of the Horizon 2020 Research and Innovation Action '527', Grant Agreement number 825182. Early steps of the work described in this paper have been funded by the Center for Digital Research in the Humanities and Social and Educational Sciences (CEDIFOR) in a pilot project on Quantitative and Qualitative Aspects of Historical German Linguistics (QuantQual@CEDIFOR, Feb - Sep 2017).

We would like to thank three anonymous reviewers for feedback and additions.

9. Bibliographical References

Abraham, W. (2007). Discourse binding: DP and pronouns in German, Dutch, and English. In *Nominal Determination*, pages 21–47. John Benjamins.

Akbik, A., Chiticariu, L., Danilevsky, M., Li, Y., Vaithyanathan, S., and Zhu, H. (2015). Generating high quality proposition banks for multilingual semantic role labeling. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, pages 397–407, Beijing, China.

Bird, S. and Liberman, M. (2001). A formal framework for linguistic annotation. *Speech communication*, 33(1-2):23–60.

Brants, T., Skut, W., and Uszkoreit, H. (2003). Syntactic annotation of a German newspaper corpus. In *Treebanks*, pages 73–87. Springer, Dordrecht.

Brants, S., Dipper, S., Eisenberg, P., Hansen-Schirra, S., König, E., Lezius, W., and Uszkoreit, H. (2004). TIGER: Linguistic interpretation of a German corpus. *Research on language and computation*, 2(4):597–620.

Burchardt, A., Padó, S., Spohr, D., Frank, A., and Heid, U. (2008). Formalising multi-layer corpora in OWL/DL – Lexicon modelling, querying and consistency control. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP-2008)*, Hyderabad, India.

Chiarcos, C. and Fäth, C. (2017). CoNLL-RDF: Linked corpora done in an NLP-friendly way. In *Proceedings of the First International Conference on Language, Data and Knowledge (LDK-2017)*, pages 74–88, Galway, Ireland. Springer.

Chiarcos, C. and Fäth, C. (2019). Graph-based annotation engineering: Towards a gold corpus for Role and Reference Grammar. In *Proceedings of the 2nd Conference on Language, Data and Knowledge (LDK 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Chiarcos, C. and Glaser, L. (2020). A tree extension for CoNLL-RDF. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC-2020)*, pages 7161–7169, Marseille, France.

Chiarcos, C. and Schenk, N. (2018). The ACoLi CoNLL libraries: Beyond tab-separated values. In *In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Chiarcos, C., Khait, I., Pagé-Perron, É., Schenk, N., Fäth, C., Steuer, J., Mcgrath, W., Wang, J., et al. (2018a). Annotating a low-resource language with llod technology: Sumerian morphology and syntax. *Information*, 9(11):290.

Chiarcos, C., Kosmehl, B., Fäth, C., and Sukhareva, M. (2018b). Analyzing Middle High German syntax with RDF and SPARQL. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Chiarcos, C., Ionov, M., Glaser, L., and Fäth, C. (2021). An ontology for CoNLL-RDF: Formal data structures for TSV formats in language technology. In *Procedings of the 3rd Conference on Language, Data and Knowledge (LDK 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Chiarcos, C. (2012). POWLA: Modeling linguistic corpora in OWL/DL. In *proceedings of the Extended Semantic Web Conference (ESWC-2012)*, pages 225–239, Heraklion, Greece. Springer.

Druskat, S., Gast, V., Krause, T., and Zipser, F. (2016). corpus-tools. org: An interoperable generic software tool set for multi-layer linguistic corpora. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4492–4499.

Fäth, C., Chiarcos, C., Ebbrecht, B., and Ionov, M. (2020). Fintan - Flexible, integrated transformation and annotation engineering. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC-2020)*, pages 7212–7221, Marseille, France.

Fäth, C. and Chiarcos, C. (2022). Spicy salmon: Converting between 50+ annotation formats with Fintan, Pepper, Salt and POWLA. unpublished ms.

Hardie, A. (2012). CQPweb. Combining power, flexibility and usability in a corpus analysis tool. *International journal of corpus linguistics*, 17(3):380–409.

Hellmann, S., Lehmann, J., Auer, S., and Brümmer, M. (2013). Integrating NLP using linked data. In *Proceedings of the International Semantic Web Conference (ISWC-2013)*, pages 98–113, Berlin, Heidelberg. Springer.

Hogan, A., Reutter, J. L., and Soto, A. (2020). In-database graph analytics with recursive SPARQL. In *Proceedings of the International Semantic Web Conference (ISWC-2020)*, pages 511–528. Springer.

Ide, N. and Suderman, K. (2014). The Linguistic An-

notation Framework: A standard for annotation interchange and merging. *Language Resources and Evaluation*, 48(3):395–418.

Ionov, M., Stein, F., Sehgal, S., and Chiarcos, C. (2020). cqp4rdf: Towards a suite for RDF-based corpus linguistics. In *Proceedings of the European Semantic Web Conference (ESWC-2020)*, pages 115–121. Springer.

ISO. (2012). Iso 24612:2012 language resource management — Linguistic annotation framework (LAF). Technical report, International Organization for Standardization.

ISO. (2014). Iso 24615-1:2014 language resource management — Syntactic annotation framework (SynAF) — Part 1: Syntactic model. Technical report, International Organization for Standardization.

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlỳ, P., and Suchomel, V. (2014). The Sketch Engine: Ten years on. *Lexicography*, 1(1):7–36.

Krause, T. and Zeldes, A. (2016). Annis3: A new architecture for generic corpus query and visualization. *Digital Scholarship in the Humanities*, 31(1):118–139.

Kyjánek, L., Zabokrtskỳ, Z., Sevcíková, M., and Vidra, J. (2020). Universal derivations 1.0, a growing collection of harmonised word-formation resources. *Prague Bull. Math. Linguistics*, 115:5–30.

Mazziotta, N. (2010). Building the syntactic reference corpus of medieval French using Notabene RDF annotation tool. In *Proceedings of the Fourth Linguistic Annotation Workshop (LAW-2010)*, pages 142–146, Uppsala, Sweden.

McCarthy, A. D., Kirov, C., Grella, M., Nidhi, A., Xia, P., Gorman, K., Vylomova, E., Mielke, S. J., Nicolai, G., Silfverberg, M., et al. (2020). Unimorph 3.0: Universal morphology. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3922–3931.

Mengel, A. and Lezius, W. (2000). An XML-based representation format for syntactically annotated corpora. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2020)*, Athens, Greece.

Molnár, V. and Vinckel-Roisin, H. (2019). Discourse topic vs. sentence topic exploiting the right periphery of german verb-second sentences. In *Architecture of Topic*, pages 293–334. De Gruyter Mouton.

Petran, F., Bollmann, M., Dipper, S., and Klein, T. (2016). ReM: A reference corpus of Middle High German. Corpus compilation, annotation, and access. *J. Lang. Technol. Comput. Linguistics*, 31(2):1–15.

Rambow, O. (1993). Pragmatic aspects of scrambling and topicalization in german: A centering approach. In *IRCS Workshop on Centering in Discourse*.

Schmidt, T. and Wörner, K. (2014). EXMARaLDA. In Gjert Kristoffersen Jacques Durand, Ulrike Gut, editor, *The Oxford handbook of corpus phonology*. Oxford University Press.

Sloetjes, H. (2017). ELAN. In Gjert Kristoffersen Jacques Durand, Ulrike Gut, editor, *The Oxford handbook of corpus phonology*. Oxford University Press.

Speyer, A. (2007). Die Bedeutung der Centering Theory für Fragen der Vorfeldbesetzung im Deutschen. *Zeitschrift für Sprachwissenschaft*, 26(1):83–115.

Speyer, A. (2011). Die Freiheit der Mittelfeldabfolge im Deutschen. Ein modernes Phänomen. *Beiträge zur Geschichte der deutschen Sprache und Literatur (PBB)*, 133(1):14–31.

Speyer, A. (2013). Performative Mündlichkeitsnähe als Faktor für die Objektstellung im Mittel-und Frühneuhochdeutschen. *Beiträge zur Geschichte der deutschen Sprache und Literatur*, 135(3):342–377.

Strube, M. and Hahn, U. (1999). Functional Centering – Grounding referential coherence on information structure. *Computational linguistics*, 25(3):309–344.

Taylor, A., Marcus, M., and Santorini, B. (2003). The Penn Treebank: An overview. In *Treebanks: Building and Using Parsed Corpora*, chapter 20. Springer.

Vinckel-Roisin, H. (2011). Wortstellungsvariation und Salienz von Diskursreferenten. *Zeitschrift für germanistische Linguistik*, 39(3):377–404.

Wunsch, H. (2006). Anaphora resolution – What helps in German. In *Pre-proceedings of the International Conference on Linguistic Evidence, Tübingen, Germany*, pages 2–4.

Ziad, H., McCrae, J. P., and Buitelaar, P. (2018). Teanga: A linked data based platform for natural language processing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

10. Language Resource References

Chiarcos, C. and Plate, R. and Sukhareva, M. and Kosmehl, B. (2018). *Middle High German Treebank / Baumbank Mittelhochdeutsch, Version 0.1*. Applied Computational Linguistics (ACoLi) Lab, Goethe University Frankfurt am Main, `https://github.com/acoli-repo/germhist/edit/master/ReM/full_corpus`, ISLRN 557-678-923-277-5.

De Marneffe, M., Manning, C., Nivre, J., and Zeman, D. (2021). Universal dependencies. *Computational linguistics*, 47(2):255–308, ISLRN 586–682–285–530–1.

Demske, U. (2003). *Mercurius Treebank*. Universität Potsdam. `https://www.uni-potsdam.de/en/guvdds/projects/complproj/mercuriustreebank-1`.

Durrell, M. and Bennett, P. and Scheible, S. and Whitt, R. (2012). *The GerManC Corpus*. Manchester: School of Languages, Linguistics and Cultures.

Eckhoff, H., Bech, K., Bouma, G., Eide, K., Haug, D., Haugen, O. E., and Jøhndal, M. (2018). The PROIEL treebank family: A standard for early attestations of Indo-European languages. *Language Resources and Evaluation*, 52(1):29–65. http://hdl.handle.net/11495/DB0A--50D0--AD5A--1.

Klein, T. and Wegera, K. and Dipper, S. and Wich-Reif, C. (2016). *Referenzkorpus Mittelhochdeutsch (1050–1350), Version 1.0*. https://www.linguistics.ruhr-uni-bochum.de/rem/, ISLRN 332-536-136-099-5.

Light, C. (2012). The information structure of subject extraposition in Early New High German. *University of Pennsylvania Working Papers in Linguistics*, 18(1):20.

Lühr, R. and Faßhauer, V. and Prutscher, D. and Seidel, H. (2014). *Fuerstinnenkorrespondenz 1.1, Version 1.1*. Universität Jena, DFG. https://doi.org/10.34644/laudatio-dev-ZCSVC3MB7CArCQ9CVedt.

Petrova, S. and Donhauser, K. and Odebrecht, C. (2011). *Tatian Corpus of Deviating Examples 2.1, Version 2.1*. Humboldt-Universität zu Berlin. https://doi.org/10.34644/laudatio-dev-aiTdCXMB7CArCQ9CzEPy.

Scheible, S., Whitt, R., Durrell, M., and Bennett, P. (2011). A gold standard corpus of Early Modern German. In *Proceedings of the 5th Linguistic Annotation Workshop (LAW-2011)*, pages 124–128. https://www.sketchengine.eu/germanc--corpus/.

Taylor, A. and Warner, A. and Pintzuk, S. and Beths, S. and others. (2003). *The York-Toronto-Helsinki Parsed Corpus of Old English Prose (YCOE)*. Oxford Text Archive Core Collection, University of Oxford. https://www-users.york.ac.uk/~lang22/YCOE/YcoeHome.htm.

Telljohann, H., Hinrichs, E., Kübler, S., and Kübler, R. (2004). The TüBa-D/Z treebank. http://www.linse.uni-due.de/linguistische-korpora-und-datensammlungen/tuebinger-baumbank-des-deutschen-zeitungskorpus-tueba.html.

Walkden, G. (2016). The HeliPaD: A parsed corpus of Old Saxon. *International Journal of Corpus Linguistics*, 21(4):559–571. https://dx.doi.org/10.1075/ijcl.21.4.05wal.

Wegera, K. and Solms, H. and Demske, U. and Dipper, S. (2021). *Referenzkorpus Frühneuhochdeutsch (1350–1650), Version 1.0*. https://www.linguistics.ruhr-uni-bochum.de/ref/, ISLRN 918-968-828-554-7.