Consolidating Syntactically Annotated Corpora with LLOD Technology. An Experiment in the Old Saxon Heliand

Christian Chiarcos[a], Janine Siewert[a,b]
[a]Applied Computational Linguistics (ACoLi), University of Augsburg, Germany
[b]Department of Digital Humanities, University of Helsinki, Finland
christian.chiarcos@uni-a.de, janine.siewert@helsinki.fi

Abstract

The humanities are methodologically and technologically diverse, and independent projects often produce complementary but technically incompatible digital editions from the same sources. We show how Linguistic Linked Open Data (LLOD) technology, and in particular, Fintan and CoNLL-RDF can support the post-hoc consolidation of such resources by using SPARQL updates for selection, aggregation and consolidation operations over heterogeneous annotations. This is illustrated for the Old Saxon (Old Low German) *Heliand*, a 9th-century gospel harmony annotated for different syntactic aspects in three independent projects and across different editions and manuscripts. We describe the derivation of a UD-compliant corpus through transformation into corpus-specific CoNLL formats, cross-version alignment, and annotation integration. A central challenge is the consolidation of incomplete and heterogeneous annotations.

Keywords: annotation consolidation, syntax, Fintan, RDF/SPARQL, Old Saxon (Old Low German)

1. Background and Motivation

The increasing availability of digitally annotated linguistic resources has profoundly reshaped research in historical linguistics and philology. At the same time, the humanities remain methodologically and technologically heterogeneous: different theoretical assumptions, annotation traditions and research questions regularly lead to parallel, independent annotation efforts over (often different editions or witnesses of) the same primary texts. While such diversity is scientifically productive, it results in resources that are complementary in scope but technically incompatible, leaving their combined analytical potential largely untapped.

This situation is particularly evident in historical syntax, where constituency-based treebanks inspired by generative grammar coexist with multi-tier annotations in the tradition of interlinear glossed text (Bow et al., 2003, IGT). For the Old Saxon *Heliand*, these approaches differ substantially in design and coverage. Treebanks model hierarchical structure exhaustively, whereas tier-based annotations allow flexible layering, for instance for the study of information-structural phenomena at the syntax–pragmatics interface. Re-annotation is therefore common practice – not due to deficiencies in existing resources, but because reuse across frameworks is technically difficult. As a consequence, high-quality annotations remain siloed.

The *Heliand*, a 9th-century Gospel harmony and the most extensive Old Saxon text, occupies a central position in early West Germanic philology. Its linguistic profile is crucial for reconstructing early syntactic change (Petrova and Solf, 2009; Lühr, 2025). Reflecting its importance, it has been annotated repeatedly in independent projects: the HeliPaD treebank (Walkden, 2016), following Penn-style constituency annotation; the Heliand DDD corpus (Referenzkorpus "Altdeutsch") with tier-based morphosyntactic and clausal annotation; and the Heliand B4 corpus (Linde, 2009), developed for diachronic information-structural research. These corpora differ in granularity. HeliPaD provides full phrase-structure parses; DDD and B4 offer partial, non-recursive annotations (POS, clause boundaries, and in B4 nominal and prepositional phrases). Consolidation therefore requires (i) conversion to CoNLL(-U), (ii) enrichment of partial annotations, (iii) alignment across divergent textual bases, and (iv) merging of alternative syntactic analyses.

We focus on conversion and merging, implemented as graph rewriting operations using SPARQL over on-the-fly transformations between CoNLL sentence blocks and RDF graphs. While earlier work proposed RDF for storage and querying of multi-layer corpora (Burchardt et al., 2008; Mazziotta, 2010), later research demonstrated its suitability for transformation tasks (Fäth et al., 2020; Chiarcos et al., 2021), including integration of heterogeneous syntactic annotations (Chiarcos et al., 2022). In this paper, we extend the application of RDF and Linked Open Data (LLOD) technologies to the combination of conflicting syntax annotations into a coherent CoNLL-U representation. Although RDF and LLOD have been applied previously to harmonizing annotations, the post-hoc consolidation of conflicting legacy annotations for pre-modern corpora remains underexplored and is addressed in this paper for the specific case of the Heliand.

Proceedings of the 10th Workshop on Linked Data in Linguistics (LDL-2026) @ LREC 2026, pages 29–39
12 May 2026.

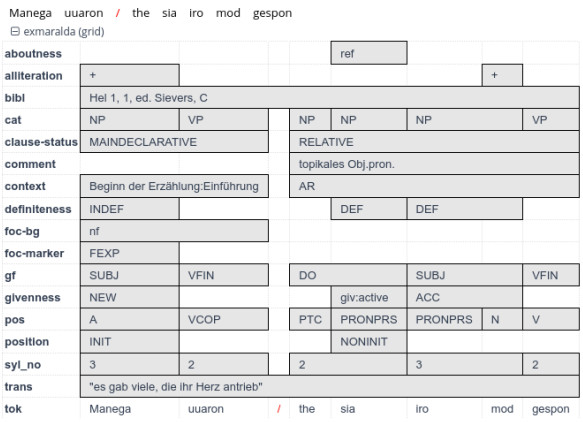

Figure 1: Heliand B4 annotation, first verse, visualized with ANNIS3

2. From Sources to Syntax Annotation. The Story so far

Old Saxon (OS) is attested primarily in two major 9th-century alliterative poems, the *Heliand* and the *Genesis*. Apart from a small number of liturgical texts and tax lists, these constitute the core of the Old Saxon textual record. With roughly 6,000 long verses, the *Heliand* is by far the more extensive text and has therefore become the principal basis for corpus-based research. It is preserved in two major manuscripts, C (London, British Library, Cotton Caligula A. VII) and M (Munich, Bayerische Staatsbibliothek, Cgm 25), as well as in four fragments.

The first syntactically annotated corpus of the *Heliand* was the *Heliand B4 corpus* (Petrova, 2006; Linde, 2009), developed within the Collaborative Research Center 632 “Information Structure”. Its annotation methodology, described in detail by Petrova et al. (2009), relies on the tier-based tool EXMARaLDA (Schmidt and Wörner, 2014). Tokens form the primary timeline; additional layers provide morphosyntactic categories, syntactic functions, information-structural features, and metadata. These layers are attached to spans of tokens in a stand-off architecture. This design allows overlapping and independent hierarchies without enforcing a single dominant structure, and thus accommodates conflicting annotations and discourse-level relations. The data are converted to the PAULA interchange format and integrated into ANNIS for querying and visualization (Petrova et al., 2009). The same multi-tier strategy has been applied to Old Saxon, resulting in a richly annotated but comparatively small corpus: Heliand B4 (Fig. 1) comprises approximately 3,500 tokens, covering less than 10% of the text. Despite its limited size, it remains a valuable complement of the later DDD corpus in providing rich manual annotation, in particular for nominal and prepositional phrases and their grammatical functions.

```
( (IP-MAT
    (CODE <P_7>)
    ...
    (NP-SBJ *exp*)
    (NP-PRD (Q^N^PL Manega-manag)
            (CP-REL *ICH*-1))
    (BEDI^3^PL uuaron-wesan)
    (CODE <C>)
    (CP-REL-1
      (WNP-SBJ-2 0)
      (C the-the)
      (IP-SUB
        (NP-SBJ-RSP-2
          (PRO^A^3^PL sia-he))
        (NP-OB1 (PRO$^N^3^SG iro-his)
                (N^N^SG mod-mod))
        (GE+VBDI^3^SG gespon-spanan)
        (, ,-,)
        ... ))))
```

Figure 2: HeliPaD annotation, first verse, PTB bracketing format

The *Old German Reference Corpus (Referenzkorpus Altdeutsch, DDD)* (Linde and Mittmann, 2013) project aimed to digitize and annotate all extant Old High German and Old Low German (Old Saxon) texts, using a tier-based approach similar to the Heliand B4 corpus, but different tools. For the *Heliand*, DDD is based on the synoptic 55,080-token edition of Behaghel and Taeger (1984), digitized via the TITUS project (Gippert, 2011), enriched with morphological information and lemmatization from Sehrt (1925), and manually revised. Annotation was carried out in ELAN (Brugman and Russel, 2004), another tier-based tool. DDD provides detailed morphosyntactic annotation and shallow syntactic structure in the form of clause chunks and linking relations, but it does not encode full phrase structure trees. Syntactic annotation remains non-recursive and limited to higher-level segmentation. Compared to Heliand B4, DDD covers the complete text but at a shallower syntactic level.

In contrast to both tier-based corpora, the *HeliPaD* (Heliand Parsed Database) (Walkden, 2016) introduces full syntactic annotation for Old Saxon. It follows the conventions of the Penn Historical Corpora (Pintzuk and Taylor, 1997; Taylor, 2003; Booth et al., 2020; Kulick et al., 2022) and encodes explicit constituency structures and grammatical functions (Fig. 2). HeliPaD is based on Sievers (1878), and unlike the synoptic edition underlying DDD, this represents a single manuscript (Ms. C) and thus preserves manuscript-specific orthography and variation. The corpus comprises approximately 46,000 tokens and is smaller than DDD but syntactically more explicit.

Together, these three corpora present a heterogeneous landscape. Each contributes unique in-

formation: HeliPaD offers full constituency structure; DDD provides comprehensive coverage with consistent morphosyntax and clause segmentation; B4 supplies fine-grained manual phrase-level and information-structural annotation. At the same time, they differ in textual basis (single manuscript vs. synoptic edition), tokenization and orthography, annotation depth, tools (EXMARaLDA, ELAN, Penn bracketing), and formats. While conversion of Penn-style bracketing to dependency formats has been addressed in previous work (Johansson and Nugues, 2007; Arnardóttir et al., 2020), tier-based corpora pose a different problem. Their shallow, non-recursive structure cannot simply be converted; it must be enriched with additional syntactic information. In our approach, HeliPaD provides this enrichment both through its direct conversion to UD and through a parser trained on its UD representation and applied to DDD and B4 data.

Two challenges arise: (1) alignment and projection across divergent textual versions of the same work, and (2) consolidation of multiple and incomplete layers of syntax annotation into a consistent representation. The following sections describe how these can be addressed in a unified workflow.

3. Technological foundations

For consolidating the diverse Heliand annotations, we use four primary technical core components: (1) the Flexible Integrated Annotation eNgineering platform (Fäth et al., 2020, Fintan) allows to use SPARQL for transformation tasks, (2) the CoNLL-RDF customization of Fintan (Chiarcos and Fäth, 2017; Chiarcos et al., 2021) for reading and writing CoNLL data, (3) the UDPipe parser (Straka and Straková, 2017) for training CoNLL-U parsers and analysing unseen data, and (4) CoNLL-Merge (Chiarcos and Schenk, 2018) for merging CoNLL files with multiple layers of annotation from different sources. In addition, custom converters transform native corpus formats (PTB bracketing, ELAN, EXMARaLDA) into CoNLL-style TSV representations.

Fintan is a flexible framework for converting, enriching and transforming heterogeneous linguistic annotations by mapping them to RDF graphs and applying SPARQL queries and updates. It is modular and separates loading, transformation and serialization: Loaders convert native formats into RDF graphs that mirror their original structure; transformation modules use SPARQL to perform graph rewriting operations; writers serialize results back into conventional formats, including CoNLL. Because Fintan operates on RDF graphs that represent single sentences, not the complete data set, it can parallelize SPARQL updates, enabling scalable graph rewriting. In doing that, Fintan is agnostic to linguistic data models and vocabularies. This flexibility allows us to harmonize structurally divergent resources within a single graph-based workflow.

For corpus processing, we operate with the CoNLL-RDF vocabulary (Chiarcos et al., 2021), summarized in Fig. 3. CoNLL(-TSV) formats represent sentences as blocks of tab-separated rows, one token per line, optionally preceded by comment metadata. Columns encode annotations such as form, lemma, POS, morphological features, dependency relations or task-specific labels. CoNLL-RDF maps each token to a `nif:Word` node, linked via `nif:nextWord`, with column values represented as literal properties in the `conll:` namespace. Some columns receive dedicated treatment: `HEAD`, for example, is resolved into explicit `conll:HEAD` relations that link tokens (`nif:Word`s) with each other (for syntactic dependencies) or the sentence (for roots). The CoNLL-RDF tree extensions (Chiarcos and Glaser, 2020) extend the basic CoNLL-RDF model with the POWLA vocabulary to represent hierarchical structures such as phrase-structure trees. This extension allows Penn Treebank-style parses to be encoded as additional nodes linked via `powla:hasParent` and `powla:next`. Figure 4 shows a fragment with a token connected to a phrase node, which participates in a larger tree. Any kind of TSV-compatible annotation can be represented in a unified RDF graph, which can be easily traversed and manipulated with SPARQL, even if the data includes multiple layers of styles of syntax annotation, be it phrase structure trees or dependency syntax.

CoNLL-Merge complements this graph-based transformation layer by aligning different witnesses or editions encoded in CoNLL-style formats. It supports token-based alignment, merging and splitting operations, and concatenation of aligned annotations into additional columns. This is crucial for integrating independent annotations of the same text that differ in tokenization or textual basis. Once merged into a common CoNLL representation, annotations can be consolidated through SPARQL-based graph rewriting in CoNLL-RDF.

Overall, the combination of format conversion, alignment (CoNLL-Merge), graph-based normalization and rewriting (Fintan/CoNLL-RDF), and parser-based enrichment (UDPipe) provides a modular and extensible infrastructure for consolidating heterogeneous annotations into a unified CoNLL-U representation. We illustrate this for the Heliand corpora where annotation depth varies considerably: HeliPaD provides full constituency parses; DDD offers clause-level segmentation and morphosyntax; B4 contains non-recursive nominal and prepositional chunks. To compensate for missing syntactic structure, we train UDPipe (Straka and Straková, 2017) on a CoNLL-U conversion of HeliPaD and apply the resulting parser to DDD and

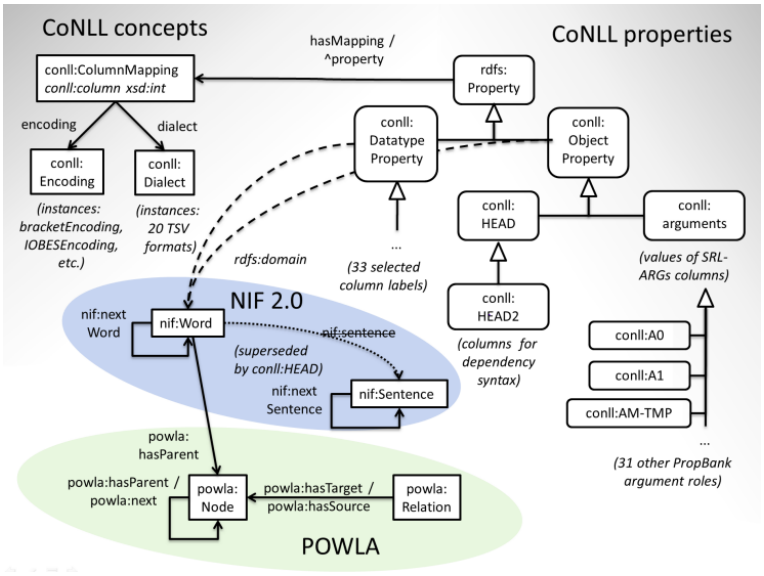

Figure 3: CoNLL-RDF Data Model (Chiarcos et al., 2021)

```
:s1_0    a                nif:Sentence .
:s1_7    a                nif:Word ;
         conll:HEAD       :s1_0 ;
         conll:WORD       "Manega" .
         conll:POS        "Q^N^PL" ;
         conll:LEMMA      "manag" ;
         nif:nextWord     :s1_8 ;
         powla:hasParent  :bPARSE_4 .
:bPARSE_4  a              conll:PARSE ;
         conll:CAT        "NP" ;
         conll:ROLE       "PRD" ;
         powla:hasParent  :bPARSE_2 ;
         powla:next       :bPARSE_5 .
```

Figure 4: CoNLL-RDF fragment for the first word of HeliPaD, with tree extensions

B4 data. The automatically generated dependency structures serve as a baseline that complements projected annotations and fills structural gaps.

4. From heterogeneous sources to a CoNLL-U

Figure 5 illustrates the different conversion, training and merging steps: We convert HeliPaD into CoNLL-U and train UDpipe over it. We convert DDD and B4 to CoNLL representations and align these with both HeliPaD-UD and automatically parsed dependencies to derive an initial CoNLL-U representation. In subsequent processing (Sect. 5), the CoNLL-U editions of all three corpora are merged into full annotations for two major versions of the Heliand, Ms. C (text basis from HeliPaD) and the synoptic BT text (text basis from DDD).

In Fig. 5, corpora without boxes designate datasets in their original format, corpora in boxes indicate CoNLL-U data, corpora in dashed boxes indicate corpus-specific CoNLL (one-word-per-line TSV) formats. Arrows indicate conversion (e.g., from HeliPaD to HeliPaD UD), training (e.g., from HeliPaD UD to the parser), the application of a tool to data (only for the parser) or merging (e.g., from Heliand B4 CoNLL (with converted manual annotations) and Heliand B4 Parsed (with automated annotations) to Heliand B4 UD. For merging, the thickness of an arrow indicates the priority, e.g., manual annotations from the Heliand B4 CoNLL corpus take priority over automated annotations from Heliand B4 Parsed – but only when the former are actually available.

4.1. HeliPaD conversion

For converting HeliPaD from the Penn bracketing format, we apply a straight-forward replication of rule-based approaches such as Arnardóttir et al. (2020) in SPARQL. As for this transformation, we would argue that Fintan and SPARQL allow to easily replicate existing rules, but that it does so in a declarative, less idiosyncratic and more portable manner, because it is the first tool to perform that task that allows to decouple transformation logic (in SPARQL) from the actual format conversion (by Fintan loader and writer modules), whereas earlier systems are characterized by merging conversion and transformation logic and thus, difficult to re-use and adapt. In fact, Arnardóttir et al. (2020) emphasize that their approach is specific to the IcePaHC corpus, and also, that they felt the need to develop it from scratch because existing software for converting the Penn bracketing format (such as the one by Johansson and Nugues (2007) and the Univer-

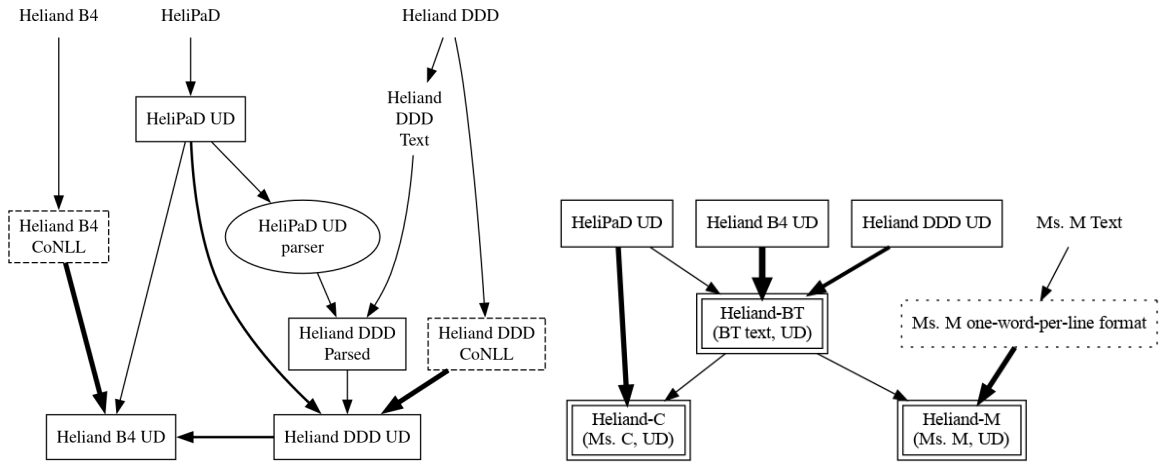

Figure 5: Conversion and consolidation workflows. **left:** conversion of source corpora (Heliand B4, HeliPaD, Heliand DDD) to CoNLL-U format (HeliPaD UD, Heliand DDD UD, Heliand B4 UD). **right:** consolidation of CoNLL-U conversions of source corpora into CoNLL-U editions of different text versions (BT UD: text according to the critical edition by Behaghel-Taeger, with text based on DDD; Ms. C UD: Manuscript C, with text based on HeliPaD; Ms. M UD: Manuscript M, with text based on Schmeller, 1830)

salDependenciesConverter of StanfordNLP[1]) could not be adapted. Beyond that, however, the transformation is otherwise equivalent. We describe details of the SPARQL-based conversion in Chiarcos and Siewert (2026). The focus of this paper is more on innovative aspects of SPARQL, in particular to formulate rules that can operate over *multiple* levels of conflicting syntax annotation.

With HeliPaD converted to CoNLL-U, it is now possible to train the dependency parser of your choice over the Heliand. We conducted our experiments with UDpipe (Straka and Straková, 2017). The trained parser was then used to 'fill in the gaps' of the manual DDD and B4 annotations.

4.2. DDD and B4 conversion

Although they were created with different tools and are distributed in different formats, both the DDD corpus and the Heliand-B4 corpus use a tier-based annotations, where an abstract timeline is annotated with spans, where a span can either represent a token or a sequence of tokens that is then assigned up to one label per layer of annotation (tier). Both DDD and Heliand-B4 provide morphosyntactic annotations, lemmas, clause segments and labels for clause linking, but only B4 also provides a tier for phrases (for NPs, VPs or PPs) and grammatical functions (within the clause). It is to be noted, however, that this is not a full-fledged parse tree as tier-based annotation does not support recursive structures. For the description of the conversion process, we focus on B4, because it provides richer annotations. The general principles and technologies we used are, however, the same for DDD.

We first transform the (DDD and B4) source data into a CoNLL representation. Then, we use CoNLL-Merge to force-align this source data with other ('target') CoNLL annotations which – for tokens that can be aligned – are appended to the original CoNLL columns. For tokens that cannot be aligned, the corresponding number of target columns is filled up with empty annotations. For target tokens that cannot be aligned with source tokens, new rows are added and the columns representing the source data and its annotations are filled up with placeholder symbols (here `*`) by CoNLL-Merge. In our implementation, these are just filtered out. The process can be iterated to align more than two types of CoNLL source source annotations.

For the case of DDD, we merge (a CoNLL representation of) the original DDD annotations with an automated CoNLL-U parse of this data (using a UDpipe parser trained over HeliPaD UD) and with the CoNLL-U conversion of the HeliPaD. Technically, all HeliPaD (ms. C) is included in the (text edition underlying the) DDD corpus. The resulting CoNLL data features a total of 35 columns, and using user-defined labels for all of them, these are mapped to properties in the `conll:` namespace by CoNLL-RDF. Using SPARQL rules as described below, these are then merged by projecting HeliPaD (and, where not available, automatically parsed) dependencies onto DDD data, and using DDD clause labels for inferring dependency labels. The result is then serialized as CoNLL-U.

The B4 corpus was created in EXMARaLDA but has been published in the ANNIS format, so that it can be readily converted to RDF via the SALT

[1] https://nlp.stanford.edu/software/stanford-dependencies.shtml, accessed: 2026-02-15.

component of Fintan (Fäth and Chiarcos, 2022) and serialized into a custom CoNLL format where each tier corresponds to exactly one column. In subsequent processing, this is merged with the resulting CoNLL-U representation of DDD as well as with HeliPaD UD, and a selection (`cut`) of the resulting 58 columns (38 original tiers and 20 CoNLL-U columns) is then further processed by CoNLL-RDF. Although the original corpus does not provide phrasal structures, we could derive a phrase-structural representation from the extent of spans on the respective tiers: We create `powla:Node` entities for every annotation of the respective tier, and `powla:hasParent` relations between overlapping segments according to the following hierarchical organization of tiers: `tok` → `pos` → `cat` (phrase types) → `gf` (grammatical function) → `clause`. The resulting CoNLL-RDF graph is illustrated in Fig. 6. In addition to syntax and glossing, B4 provides annotations for rhyme, bibliographical and context information, as well as annotation for information structure, e.g., aboutness, definiteness, focus-background organization and givenness, excluded from the figure to facilitate readability.

Heliand B4 has been aligned with HeliPaD UD as well as with the CoNLL-U version of Heliand DDD. As its text basis is (except for variance in transliteration) identical to that of DDD, it is not directly parsed automatically, but incorporates major aspects of the automated parses via the DDD alignment.

For inferring consolidated dependencies from the Heliand B4 tree of `powla:Node`s and the DDD- and HeliPaD UD dependencies, three primary transformation steps need to be performed: (1) for every node in the B4 tree, detect the child element that contains the UD head and mark it as `temp:HEAD`, (2-3) for every word that is not a `temp:HEAD`, copy the dependencies from DDD (`conll:DHEAD`), resp. (where not available) HeliPaD-UD (`conll:HHEAD`), and (4-5) collapse the phrase structure to the words that serve as (transitive) `temp:HEAD`s of the respective phrases:

(1) Within every `powla:Node` and every `nif:Sentence`, we identify the syntactic head as the `nif:Word` that has the largest number of dependents in DDD or HeliPaD annotation (as a property path: `(^conll:HHEAD|^conll:DHEAD)* []`). If there are multiple candidate heads, we use the first. In Fig. 6, the `temp:HEAD` edges pointing from `powla:Node`s to the respective head of each phrase are marked in bold.

(2) For siblings in n-ary phrases whose heads (`temp:HEAD*`) are linked in aligned DDD annotation, we copy the DDD dependency and remove the sibling that contains the dependent of the relation.

(3) For siblings in remaining n-ary phrases whose heads are linked in aligned HeliPaD annotation, we copy the HeliPaD dependency and remove the sibling that contains the dependent.

(4) For every binary phrase, we establish a link between the (head of the) non-head child and the (head of the) `temp:HEAD` child and remove the sibling that contains the dependent.

(5) For every remaining n-ary phrase, we link the (heads of) non-head children with the (head of the) `temp:HEAD`.

Note that all these rules exploit the special capabilities that SPARQL property paths provide to navigate within RDF graphs: We use the *transitive closure*, e.g., for the repeated lookup of `temp:HEAD` in (2), marked by `*`, we use the *directional inversion* of `conll:HHEAD` to aggregate over all dependents of a given word in (1), marked by `^`, we use the *disjunction* between different RDF properties in (1), marked by `|`, etc. In code, each of these steps is represented by (at least) one SPARQL Update, appropriately formatted and commented, separated by empty lines and `;` and aggregated into a single `*.sparql` file. Fintan then applies one or more `*.sparql` files in their sequential order to every RDF graph, and, optionally, also to perform loops.

Dependency labels (`conll:EDGE`) are also induced in this process. In (2-3), these are adopted from DDD, resp. HeliPaD along with the transfer of dependencies. For unlabelled dependencies, steps (4-5) provide a mapping from the annotations of the `powla:Node`s that the respective `nif:Word` is `temp:HEAD` of (`(^temp:HEAD)*`). Primarily, this exploits the grammatical function (`gf`) annotation. For words without `gf` annotation, we resort to DDD labels. `UPOS` is adopted from DDD. Figure 7 shows the consolidated result graph.

5. Consolidating Annotations

As the corpora cover different parts from different manuscripts, these need to be regrouped so that we provide individual annotations for the different textual witnesses. Fig. 5 summarizes the annotation consolidation workflow, with source corpora in CoNLL-U marked in boxes, corpora in other CoNLL (one-word-per-line TSV) formats in dotted boxes, and resulting corpora in double boxes. The arrows indicate transformation and merging processes.

Overall, we focus on both major versions of the text for which we possess manual annotations: Heliand-C (Ms. C) primarily based on HeliPaD and Heliand-BT (synoptic text) primarily based on DDD (with additions from B4 and HeliPaD).

Heliand-BT provides the text of the critical edition by Behaghel and Taeger (1984), the most compre-

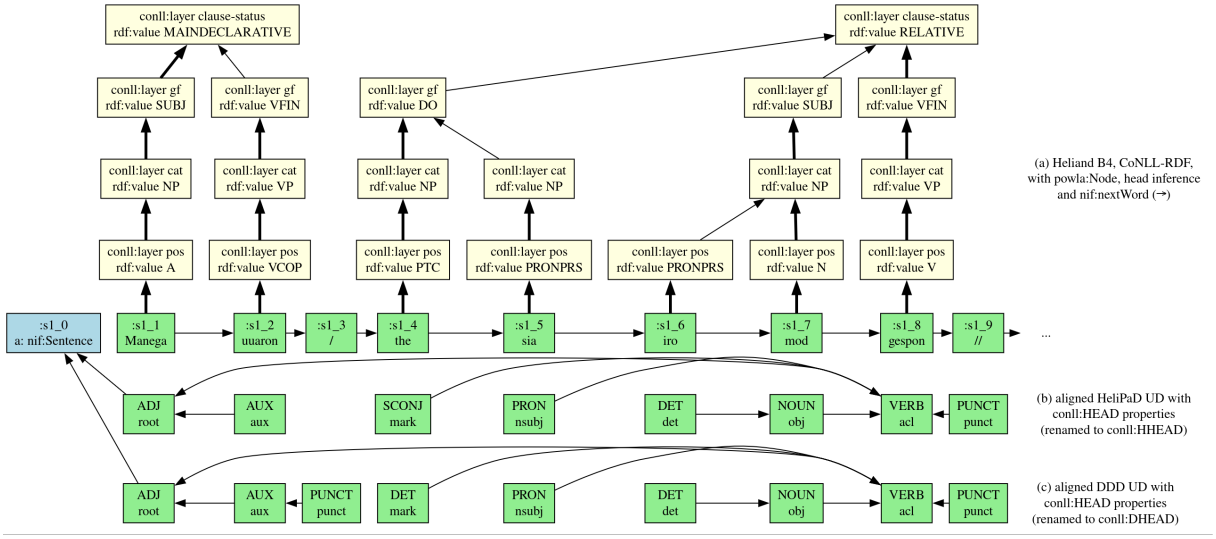

Figure 6: First verse of Heliand B4, compact visualization of a single CoNLL-RDF graph consisting of (a) B4 annotations with inferred `powla:hasParent` (↑, **↑**) and `temp:HEAD` properties (inverse of bold **↑**), (b) merged HeliPaD UD annotations over the same `nif:Word`s, and (c) merged DDD UD annotations.

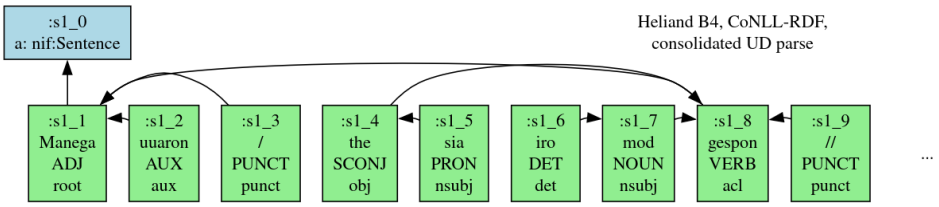

Figure 7: Heliand B4, consolidated CoNLL-RDF graph as constructed from Fig. 6 with SPARQL Updates, with `conll:HEAD` properties (←, →, ↑); `nif:nextWord` not shown.

hensive text version of the Heliand, but, in comparison to the source manuscripts, amended at several occasions. It forms the basis of both Heliand DDD and Heliand B4, but it should also contain all textual material covered by Ms. C (i.e., HeliPaD), as well as Ms. M (not annotated, so far), the most extensive witnesses. Building a consolidated and internally consistent Heliand UD corpus over all three corpora thus required to first align all subcorpora with the BT text (as provided by the DDD corpus) and to aggregate their information, and then to project (parts of) their information into the annotated corpora, resp., Ms. M.

As our CoNLL-U edition of the DDD corpus already builds on an alignment with HeliPaD, the DDD CoNLL-U corpus was taken as a basis and aligned with the B4 CoNLL-U corpus. The input to CoNLL-RDF is thus a 20-column format that provides DDD CoNLL-U, followed by B4 annotations (where available) for all DDD words, and using DDD sentence splits.

With SPARQL Update, the merged dependency annotations are copied from B4 (because this provides richer manual annotations), or from DDD if no B4 annotations are available. In case of cycles, we remove all B4 annotations for the current sentence and resort to DDD. The values for `UPOS`, `XPOS` (i.e., original `POS` of HeliPaD and `pos` and `morph` annotations of DDD concatenated) and `LEMMA` are taken from DDD. For debugging purposes, the `MISC` column is used for tracking where which piece of annotation came from. In release data, this information is subsequently removed. The result is serialized in CoNLL-U. Figure 8 illustrates an excerpt of the merged corpus.

Effectively, the resulting parse is 99% (54735/55080 tokens) identical with DDD (because of the limited size of B4 Heliand, because of using DDD as fallback where alignment fails, and because DDD parses are mostly confirmed by B4). This does not mean that they are error-free, because the B4 UD annotation is partially derived from DDD, and because both source corpora provide *partial* manual annotations for syntax only, and new errors may have been introduced both by the alignment with HeliPaD and the automated parsing (unless superseded by the partial annotations provided by DDD or B4), whereas morphosyntax is (mostly) based on original DDD annotations.

Heliand-C is primarily based on the previously con-

```
1 *      ,      PUNCT CODE|_                                _ 2 punct _ DDD
2 Manega manag  ADJ   Q^N^PL|DIS.MASC_PL_NOM_ST            _ 0 root  _ B4=DDD
3 uuâron wesan  AUX   BEDI^3^PL|VVFIN.IND_PAST_PL_3        _ 2 aux   _ B4=DDD
4 ,      ,      PUNCT ,|$,                                 _ 2 punct _ DDD
5 the    de     DET   C|DDSREL.MASC_PL_ACC                 _ 9 mark  _ DDD
6 sia    he     PRON  PRO^PL^A^3|PPER.MASC_PL_ACC_3        _ 5 nsubj _ B4
7 iro    he     DET   PRO$^SG^3^N|DPOS.MASC_PL_GEN_3       _ 8 det   _ DDD
8 môd    mod    NOUN  N^N^SG|NA.SG_NOM                     _ 9 nsubj _ B4
9 gespôn spanan VERB  GE+VBDI^SG^3|VVFIN.IND_PAST_SG_3     _ 2 acl   _ B4
```

Figure 8: Merged Heliand-BT annotations derived from DDD and B4 (and, indirectly, HeliPaD), with provenance information in the `MISC` column

verted HeliPaD corpus, aligned with CoNLL-Merge with the Heliand-BT corpus and enriched in `XPOS` and for dependency labels for clausal juncture: The HeliPaD dependencies were left untouched, but for cases in which the same dependency relation received a different label in Heliand-BT, we resort to the Heliand-BT label. If Heliand-BT `XPOS` annotations were compatible (i.e., starting with) HeliPaD `XPOS` annotations, the Heliand-BT `XPOS` was used.

For **Heliand-M**, a plain text edition of Ms. M (Schmeller, 1830) was annotated experimentally both by the HeliPaD parser and by alignment with Heliand-BT. However, we observed a large number of alignment errors: At many occasions, the scribe decided to concatenate morphological words that Ms. C and BT would consider to be independent words, at others, a word is split. So, we read *inatorht lico* in Ms. M for *ina torhtlîco* '(that reminded) him (of) shining (times)' in BT (Ms. C: *ina torohtlico*). As a result, the alignment identifies *inatorht* as pronoun and direct object (correct for *ina*, only), and *lico* as adverb. A more correct analysis that stays true to the text and that is consistent with UD should introduce a multi-word annotation for *inatorht*, and create a `flat` link between the sub-token *torht* and *lico*. However, these kind of corrections would require manual oversight, and unless this can be provided, the Heliand-M build scripts and its textual source file are provided, but the data will not be released.

6. Results and Perspectives

In this paper, we described the end-to-end consolidation of heterogeneous syntactic annotations for the Old Saxon *Heliand* across multiple corpora, annotation styles, and textual witnesses. The workflow comprises (i) the conversion of several corpora with different syntactic coverage and formalism into CoNLL-style formats and, for the fully parsed HeliPaD corpus, into CoNLL-U, (ii) the training of a UD parser on the resulting HeliPaD-UD data, (iii) the enrichment of partially annotated corpora (Heliand B4 and Heliand DDD) by complementing manual annotations with automatically generated dependency parses, (iv) the merging of all annotations into a master representation based on the most comprehensive consolidated text (Behaghel/Taeger, BT), and (v) the alignment and projection of annotations from this master edition onto different textual witnesses (notably manuscripts C and M).

These steps yield three main contributions. First, we provide the first UD-compliant dataset for Old Saxon. Second, we demonstrate the application of CoNLL-Merge to the alignment of divergent witnesses and editions of the same source text while preserving their respective annotations, even where these are incomplete or structurally incompatible. Third, we show how CoNLL-RDF, Fintan and SPARQL can be used for rule-based merging and consolidation of multiple layers of syntactic annotation, including conflicting and partial analyses, within a unified graph-based framework.

In the broader landscape of historical Germanic corpora, several syntactically annotated resources have been created in recent years. Within the Universal Dependencies ecosystem, however, only Old Icelandic, Gothic, Old English and Middle High German are currently covered. Old English UD treebanks are largely based on conversions of Penn Treebank-style corpora. From a purely technical perspective, converting another Penn-style treebank such as HeliPaD to UD is therefore not fundamentally novel, and could be achieved with dedicated conversion scripts. Likewise, training a UD parser on the converted corpus follows established methodology. The more innovative aspect lies in the consolidation of multiple overlapping annotation layers from independent projects, combining HeliPaD, DDD, B4 and parser-based annotations. Other than graph technologies as used here, we are not aware of an existing solution that could have been used for this purpose, because existing tools typically specialize either in dependency or in constituency representations and offer limited support for handling both simultaneously. Moreover, independently created historical corpora usually differ in transliteration, tokenization and editorial principles, which makes their integration difficult and often requires substantial manual effort.

Our workflow addresses these obstacles by combining token-level alignment with graph-based representation and rewriting. CoNLL-Merge supports automated alignment across divergent tokenizations and editions and produces joint CoNLL representations with parallel annotation columns. CoNLL-RDF and Fintan convert these tabular structures into RDF graphs that can host multiple syntactic layers in parallel. Distinct annotation layers are represented through user-defined column labels and properties, enabling their separation and joint processing. SPARQL `SELECT` queries support cross-layer search and aggregation, while SPARQL Update rules enable rule-based consolidation, repair and restructuring of annotations, including the correction of alignment errors and the controlled prioritization of manual over automatic analyses. Because Fintan processes sentences as independent RDF graphs, this can be executed efficiently and in parallel with minimal memory requirements.

Related work has shown that CoNLL-RDF and Fintan can support complex annotation engineering tasks, including the merging of complementary syntactic and semantic layers into theory-specific representations, rule-based enrichment and rewriting, and joint querying across heterogeneous annotation styles. The present study extends this line of work to the consolidation of annotations that differ not only in scheme and depth but also in their underlying primary data. Our resources cover manuscripts and editions with partial textual overlap, normalized versus manuscript-faithful spelling, and divergent principles of transliteration, punctuation and word segmentation. We show that even under these conditions, large-scale consolidation is feasible using RDF-based representations and SPARQL-driven transformation.

So far, evaluation focused on technological feasibility rather than linguistic quality, as the contribution of this paper is primarily methodological and infrastructural: we establish a reproducible, extensible workflow for integrating heterogeneous legacy annotations into a coherent UD-style corpus. We adhere to manual annotations wherever available, but the exact rules employed require additional evaluation, or, adjustment on the basis of manually annotated dependency data. For Old Saxon, such data is currently prepared by the authors.

Three core technologies are central to this workflow: CoNLL-Merge for cross-version alignment and column-wise integration of annotations, CoNLL-RDF/Fintan for conversion between CoNLL(-U) and RDF and for multi-layer graph representation, and UDPipe for training a baseline parser that fills structural gaps in partially annotated corpora. Together, these components form a modular toolkit for post-hoc annotation consolidation.

From a linguistic perspective, the resulting resources open perspectives beyond the *Heliand*. For the closely related Old High German and Old Low Franconian (Old Dutch), no dependency corpora are known to exist, and only partial syntax annotations are available from DDD. As far as historical German is concerned, existing corpus initiatives and annotation experiments primarily address younger stages of the language (Sapp et al., 2024; Dipper et al., 2024; Haiber, 2024). Directly applying a the HeliPaD parser (or a novel Old Saxon parser) to these is difficult due to orthographic and dialectal variation, but where consistent lemmatization, part-of-speech tags and agreement features are available (as in DDD), it is straightforward to construct a representation that abstracts away from surface spelling. Training parsers over Old Saxon text normalized accordingly might offer a direction toward broader parsed resources. Even if automatically generated, such corpora would substantially improve empirical coverage for the study of historical West Germanic syntax.

We publish code and data described here as the Consolidated Old Saxon (ConOS) corpus `https://github.com/nds-spraakverarbeiden/ConOS` under different open source licenses. The prospective goal is to provide a UD-compliant edition of the Old Saxon data of the DDD and HeliPaD corpora, which includes the Heliand, the Old Saxon Genesis and a number of smaller fragments. At the moment, we plan to provide automatically processed data only, as created with the workflows described above. As for licensing data, we are bound to the licenses of the respective source corpora, for aggregated data, this means to follow the most restrictive source license. Fortunately, DDD, B4 and HeliPaD licenses are compatible with each other, so that this is legally possible.

In order to evaluate not only the viability of the conversion, but also the linguistic quality of the converted and consolidated annotations, we created a small training set of about 2,000 tokens based on DDD text and morphosyntax, and released it as part of the Universal Dependencies under `https://github.com/UniversalDependencies/UD_Old_Saxon-ConOS`. The UD ConOS corpus currently consists of the first fit (chapter) of Heliand (806 tokens, based on B4 text and DDD morphosyntax) and the second fragment of the Old Saxon genesis (1,145 tokens, based on the DDD corpus). Initial results on the evaluation of the consolidated annotations against the Heliand part and on linguistic aspects of the UD annotation of Old Saxon have been reported in Chiarcos and Siewert (2026). The Genesis part has been prepared for a future evaluation of automated parsing and its consolidation with DDD morphosyntax on unseen text.

7. Bibliographical References

Þórunn Arnardóttir, Hinrik Hafsteinsson, Einar Freyr Sigurðsson, Kristín Bjarnadóttir, Anton Karl Ingason, Hildur Jónsdóttir, and Steinþór Steingrímsson. 2020. A Universal Dependencies conversion pipeline for a Penn-format constituency treebank. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW-2020)*, pages 16–25, Barcelona, Spain (online).

Otto Behaghel and Burkhard Taeger. 1984. *Heliand und Genesis*, 9 edition. Max Niemeyer Verlag, Tübingen.

Hannah Booth, Anne Breitbarth, Aaron Ecay, and Melissa Farasyn. 2020. A Penn-style Treebank of Middle Low German. In *Proceedings of the Twelfth Language Resources and Evaluation Conference (LREC-2020)*, pages 766–775, Marseille, France (online).

Cathy Bow, Baden Hughes, and Steven Bird. 2003. Towards a general model of interlinear text. In *Proceedings of E-MELD Workshop 2003: Digitizing and annotating texts and field recordings*, pages 11–13, East Lansing, Michigan.

Hennie Brugman and Albert Russel. 2004. Annotating multi-media / multi-modal resources with ELAN. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-2004)*, pages 2065–2068, Lisbon, Portugal.

Aljoscha Burchardt, Sebastian Padó, Dennis Spohr, Anette Frank, and Ulrich Heid. 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: Volume-I*.

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

Christian Chiarcos, Christian Fäth, and Maxim Ionov. 2022. Querying a dozen corpora and a thousand years with Fintan. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC-2022)*, pages 4011–4021, Marseille, France.

Christian Chiarcos and Luis Glaser. 2020. A Tree Extension for CoNLL-RDF. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC-2020)*, pages 7161–7169, Marseille, France (online).

Christian Chiarcos, Maxim Ionov, Luis Glaser, and Christian Fäth. 2021. An Ontology for CoNLL-RDF: Formal Data Structures for TSV Formats in Language Technology. In *3rd Conference on Language, Data and Knowledge (LDK 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

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

Christian Chiarcos and Janine Siewert. 2026. Towards a Universal Dependency corpus for Old Saxon (Old Low German). In *Ninth Workshop on Universal Dependencies (UDW-2026)*, Palma de Mallorca, Spain. Co-located with LREC 2026.

Stefanie Dipper, Cora Haiber, Anna Maria Schröter, Alexandra Wiemann, and Maike Brinkschulte. 2024. Universal Dependencies: Extensions for modern and historical German. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17101–17111.

Christian Fäth and Christian Chiarcos. 2022. Spicy salmon: Converting between 50+ annotation formats with Fintan, Pepper, Salt and POWLA. In *Proceedings of the 8th Workshop on Linked Data in Linguistics (LDL-2022), held in conjunction with LREC-2022*, pages 61–68, Marseille, France.

Christian Fäth, Christian Chiarcos, Björn Ebbrecht, and Maxim Ionov. 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 (online).

Jost Gippert. 2011. The TITUS project. 25 years of corpus building in ancient languages. In *Perspektiven einer corpusbasierten historischen Linguistik und Philologie*. Internationale Tagung des Akademienvorhabens Altägyptisches Wörterbuch.

Cora Haiber. 2024. A Crosslingual Approach to Dependency Parsing for Middle High German. In *Proceedings of the 20th Conference on Natural Language Processing (KONVENS 2024)*, pages 23–31, Vienna, Austria.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for English. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA*

2007), pages 105–112, Tartu, Estonia. University of Tartu, Estonia.

Seth Kulick, Neville Ryant, and Beatrice Santorini. 2022. Penn-Helsinki parsed corpus of early Modern English: First parsing results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 578–593, Seattle, United States. Association for Computational Linguistics.

Sonja Linde. 2009. Aspects of word order and information structure in Old Saxon. In Roland Hinterhölzl and Svetlana Petrova, editors, *Information structure and language change: New approaches to word order variation in Germanic*, pages 367–389. Walter de Gruyter.

Sonja Linde and Roland Mittmann. 2013. Old German Reference Corpus: Digitizing the knowledge of the 19th century. In Paul Bennett, Martin Durrell, Silke Scheible, and Richard J Whitt, editors, *New Methods in Historical Corpora*, pages 235–246. Narr Francke Attempto Verlag, Tübingen.

Rosemarie Lühr. 2025. Reflexivität im Altsächsischen. In Norbert Kössinger, editor, *Altsächsisch: Beiträge zur altniederdeutschen Sprache, Literatur und Kultur*. Walter de Gruyter.

Nicolas Mazziotta. 2010. Building the syntactic reference corpus of Medieval French using NotaBene RDF annotation tool. In *Proceedings of the Fourth Linguistic Annotation Workshop (LAW IV)*, pages 142–146, Uppsala, Sweden.

Svetlana Petrova. 2006. A discourse-based approach to verb placement in Early West-Germanic. *ISIS| Working Papers of the SFB 632| 5 (2006)*, page 153.

Svetlana Petrova and Michael Solf. 2009. On the methods of information-structural analysis in historical texts: A case study on Old High German. In Roland Hinterhölzl and Svetlana Petrova, editors, *Information structure and language change: New approaches to word order variation in Germanic*, pages 121–203. Walter de Gruyter.

Svetlana Petrova, Michael Solf, Julia Ritz, Christian Chiarcos, and Amir Zeldes. 2009. Building and using a richly annotated interlinear diachronic corpus: The case of Old High German Tatian. In *Traitement Automatique des Langues, Volume 50, Numéro 2: Langues anciennes [Ancient Languages]*, pages 47–71.

Susan Pintzuk and Ann Taylor. 1997. Annotating the Helsinki Corpus: The Brooklyn-Geneva-Amsterdam-Helsinki Parsed Corpus of Old English and the Penn-Helsinki Parsed Corpus of Middle English. In *Tracing the Trail of Time: Proceedings from the Second Diachronic Corpora Workshop, New College, University of Toronto, Toronto, May 1995*, pages 91 – 104. Brill, Leiden, Niederlande.

Christopher D. Sapp, Elliott Evans, Rex Sprouse, and Daniel Dakota. 2024. Introducing a Parsed Corpus of Historical High German. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9224–9233, Torino, Italia. ELRA and ICCL.

Johann Andreas Schmeller. 1830. *Heliand: oder die altsächsische Evangelienharmonie*. Cotta.

Thomas Schmidt and Kai Wörner. 2014. EXMARaLDA. In *The Oxford handbook of corpus phonology*.

Edward Henry Sehrt. 1925. *Vollständiges Wörterbuch zum Heliand und zur altsächsischen Genesis*. Vandenhoeck & Ruprecht.

Eduard Sievers. 1878. *Heliand*. Buchhandlung des Waisenhauses, Halle.

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 shared task: Multilingual parsing from raw text to universal dependencies*, pages 88–99.

Ann Taylor. 2003. The York—Toronto—Helsinki parsed corpus of Old English Prose. In *Creating and digitizing language corpora: Volume 2: Diachronic Databases*, pages 196–227. Springer.

George Walkden. 2016. The HeliPaD: a parsed corpus of Old Saxon. *International Journal of Corpus Linguistics*, 21(4):559–571.