Automatic Detection of Language and Annotation Model Information in CoNLL Corpora

Frank Abromeit
Applied Computational Linguistics Lab, Goethe University Frankfurt, Germany
abromeit@em.uni-frankfurt.de

Christian Chiarcos
Applied Computational Linguistics Lab, Goethe University Frankfurt, Germany
http://acoli.informatik.uni-frankfurt.de/
chiarcos@informatik.uni-frankfurt.de

Abstract

We introduce *AnnoHub*, an on-going effort to automatically complement existing language resources with metadata about the languages they cover and the annotation schemes (tagsets) that they apply, to provide a web interface for their curation and evaluation by means of domain experts, and to publish them as a RDF dataset and as part of the (Linguistic) Linked Open Data (LLOD) cloud. In this paper, we focus on tabular formats with tab-separated values (TSV), a de-facto standard for annotated corpora as popularized as part of the CoNLL Shared Tasks. By extension, other formats for which a converter to CoNLL and/or TSV formats does exist, can be processed analoguously. We describe our implementation and its evaluation against a sample of 93 corpora from the Universal Dependencies, v.2.3.

2012 ACM Subject Classification Information systems → Structure and multilingual text search

Keywords and phrases LLOD, CoNLL, OLiA

Digital Object Identifier 10.4230/OASIcs.LDK.2019.23

Category Short Paper

Supplement Material https://annohub.linguistik.de

Funding The research described in this paper was conducted in the context of the Specialized Information Service Linguistics, funded by German Research Foundation (DFG/LIS, 2017-2019). The contributions of the second author were conducted with additional support from the Horizon 2020 Research and Innovation Action "Pret-a-LLOD. Ready-to-use Multilingual Linked Language Data for Knowledge Services across Sectors" (H2020-ICT-2018-2, 2019-2021).

1 Introduction

The *lin|gu|is|tik.de* portal is a virtual library which provides a rich, manually curated bibliography for linguists, coupled with inter-library search, library catalogues, indices of electronic resources, as well as various services supporting research in the language sciences [2].[1] Since 2015, we have been extending this service with respect to indexing and search over language resources, initially for language resources data provided as part of the (Linguistic) Linked Open Data (LLOD) cloud [3],[2] i.e., using RDF as a data format, HTTP URIs for identifying elements of linguistic analysis, and open licenses for data publication.

This functionality is currently being extended for indexing language resources in other popular formats. Such data is available in greater numbers than RDF-native language

[1] https://www.linguistik.de/
[2] http://linguistic-lod.org/

2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 23; pp. 23:1–23:9

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

resources, however, much of its information is implicit: In particular, RDF features explicit markers for the language of a particular string (language tags), and also URIs to identify grammatical features and linguistic annotations across different data sets, e.g., using vocabularies such as the Ontologies of Linguistic Annotation (OLiA),[3] the General Ontology of Linguistic Description (GOLD),[4] or the lexinfo model for grammatical features in lexical resources.[5] In conventional formats, such information is often missing, and if not provided as part of the formal metadata, it needs to be inferred from the data itself. In this paper, we describe a method for the automatic detection of language and annotation metadata from popular one-word-per-line (OWPL) formats, where rows correspond to individual words, and columns correspond to annotations of a particular type each. Because of their popularity, we specifically focus on CoNLL and related TSV formats as commonly used in corpus linguistics, lexicography and natural language processing.

AnnoHub is a web application that provides services to analyze language resources like corpora in RDF, CoNLL and XML formats with respect to the used annotation schemes and present languages, and to curate and publish such data. The AnnoHub web application is specifically designed to facilitate the workflow of librarians and domain experts involved in creating bibliographical records for language resources and scientific publications, it is thus internally available, only, at the moment. The resulting RDF meta data and the underlying technology stack will be published under an open license with the end of the project. Our implementation builds on – and complements – existing open source software on mapping CoNLL data to RDF [1, CoNLL-RDF].[6]

2 Automated language detection

The language detection was implemented with the *Optimaize* Java library[7] which provides n-gram-based language classification. Natively, it supports the detection of 71 languages. In order to extend the detection to other languages the library provides a tool to build new languages profiles with a text sample from a specific language. We build 444 additional language profiles from a set of about 1.500 machine-readable Bible texts created as part of earlier research.[8]

Given the large number of language models and for reasons of scalability, we perform language detection on a random sample of only 15 sentences from each CoNLL TSV file. As we aim for a generic implementation, and the position of WORD and LEMMA columns varies across different CoNLL dialects, we test *every* column for all languages (and all annotation models, see below). The language profile with the highest probability score for the majority of the 15 sentences was then selected. In some cases, increasing the set of test sentences might improve results, but for reasons of scalability, this was not tested. For detailed results we refer to section 5.

[3] `http://purl.org/olia`
[4] `http://linguistics-ontology.org/`
[5] `https://lexinfo.net/`
[6] `https://github.com/acoli-repo/conll-rdf`
[7] `https://github.com/optimaize/language-detector`
[8] Selected results of this conversion and edition project have been described by Chiarcos et al. [5], although restricted to a subset of Germanic languages. For reasons of copyright, and due to the lack of a fair use principle in German legislation, we were not able to disseminate the data. Instead, we provide build scripts for several major Bible aggregation portals in our Github repository (`https://github.com/acoli-repo/acoli-corpora/tree/master/biblical`), covering about 50% of the internally available data.

2.1 Evaluation

We evaluate our implementation on a sample of 93 corpora from the Universal Dependencies (UD) collection, v.2.3 [6].[9] In the CoNLL-U format that these corpora follow, WORD and LEMMA columns are second, resp. third column, and a summary for the language detection test over these is presented in Tab. 1. Overall, we achieved 84% accuracy (including non-detection of languages for cases where no text was provided, e.g., for ESL data).

Table 1 Result summary for language detection.

	Result	Comment
Match	78/93 (83.88%)	correct language or no language (if not present)
Partial match	3/93 (3.22%)	language correctly recognized for one of the max. 2 text columns
Weak match	2/93 (2.15%)	language found among the top 4 but not top match
Fail	2/93 (2.15%)	language was not correctly identified
No profile	8/93 (8.6%)	no language profile for the language available

Reasons for mis-classification among known languages are the proximity of certain language varieties, e.g., different varieties of Norwegian (nno/nob), the close relationship among historically closely languages (and orthographies) such as Russian (rus) and Bulgarian (bul), or Serbian (srp) and Croatian (hrv), or the relative proximity of different historical stages of the same language in the case of Ancient Greek (grc) and Modern Greek (ell). Another source of errors is that the language models are trained on texts, but that the LEMMA column contains uninflected forms only. Thus, the LEMMA column is more likely to be incorrect than the WORD column. Such errors need attention and should be (and can be) manually corrected by the user. If manual selection among the top matches for a column is allowed (and correctly applied), the accuracy can be increased by 5% to up to 89%, with unrecoverable errors going back to mis-classification (*Fail*, 2.15%) and missing language profiles (8.6%).

3 Automated detection of annotation models

Our approach to detect and disambiguate annotation models builds on the Ontologies of Linguistic Annotation [4, OLiA].[10] The OLiA ontologies provide a formalized, machine-readable view on linguistic annotations for more than 75 different language varieties, they cover morphology, morphosyntax, phrase structure syntax, dependency syntax, aspects of semantics, and recent extensions to discourse, information structure and anaphora, all of these are linked with an overarching reference terminology module. OLiA includes several multi-lingual or cross-linguistically applicable annotation models such as the Universal Dependencies (77 languages), EAGLES (11 European languages), Multext-East (16 Eastern European and Near Eastern languages).

An OLiA annotation model for a given annotation scheme (tagset) provides a formalization in terms of an ontology that defines tags (grammatical features) as instances of ontological concepts. An example for such a definition is given in Fig. 1 for the part-of-speech tag ADJ for an adjective in Morphisto, an annotation model for inflectional morphology in German[7].

[9] `https://universaldependencies.org`, `https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2895`
[10] `http://purl.org/olia`

```
@prefix system: <http://purl.org/olia/system.owl#> .
@prefix : <http://purl.org/olia/morphisto.owl#> .

:ADJ system:hasTagContaining "|ADJ"^^xsd:string ;
     system:hasTagStartingWith "ADJ"^^xsd:string ;
     a :SyntacticAdjective ;
     rdfs:comment "\"proper\" adjectives"^^xsd:string .
```

Figure 1 Definition for the part-of-speech tag ADJ in `http://purl.org/olia/morphisto.owl`.

The definition in Fig. 1 declares `http://purl.org/olia/morphisto.owl#ADJ` as an instance of the class `http://purl.org/olia/morphisto.owl#SyntacticAdjective` and assigns the annotation string "ADJ" to it. An individual does not have to correspond to a particular string (`system:hasTag`), but it can also be defined by a partial match (`system:hasTagContaining`, `system:hasTagStartingWith`, `system:hasTagEndingWith`) or a Perl-style regular expression (`system:hasTagMatching`).[11] For every annotation model, an OLiA linking model defines relationships between classes/properties in the respective annotation model and the OLiA reference model. In that way a connection between the occurrence of a annotation in a corpus and the OLiA reference model which specifies a common terminology that different annotation schemes can refer to can be established. This enables for example a SPARQL search that looks for realizations of adjectives in a RDF resource independently from the annotation scheme used in that resource – as long as an OLiA annotation model for that scheme exists.

3.1 Implementation

In a first step we build a graph database (model graph) from all OLiA annotation models. The *model graph* is a simplified version of the OLiA RDF graphs. It mainly serves 3 purposes:

- Store classes, attributes and relations of all OLiA annotation models
- Store results - annotations in CoNLL resources which could be linked to OLiA
- Enable annotation scheme detection via database queries

The *model graph* contains only 3 types of vertices: CLASS vertices are equivalent to RDF class definitions. TAG vertices contain the string value that is attached to an RDF class/individual (see Fig. 1) via a RDF property like (`hasTag, hasTagStartingWith, hasTagEndingWith or hasTagContaining`). Finally HIT vertices contain the annotation string that is found in a CoNLL file.

The following algorithm describes the steps to determine a best fitting annotation scheme for a given CoNLL column. Before the algorithm can start the set of annotations from that column has to be extracted. A annotation can be a single token (e.g. ADJ) but can also be of the form of a sequence of syntactical or morphological features, e.g. SG-IND-NOM. The input of the algorithm is then the serialization of the individual tokens in such expressions. It should be noted that the upper bound for the input size of the algorithm is the number of different annotations that are found in a CoNLL column.[12] Steps 3 and 8 in the algorithm are query operations on the model graph. Since the TAG vertices are directly connected to CLASS vertices via an edge, the query operation in step 8 is very cheap. The query in step

[11] As an example for a regular expression, consider ^AJ...1.* for `http://purl.org/olia/eagles.owl#NominativeCase`.

[12] Test files commonly contained up to 100 different annotation types per column.

Algorithm 1 Annotation model detection.

1: Extract tokens_i={Annotations from CoNLL file column i}
2: **For t** in tokens_i :
3: **Try** to match **t** with the string/regex of a TAG vertex in the model graph
4: **If** (**t** matches TAG_j) **then**
5: **1.** Insert a new vertex HIT_t into the model graph
6: **2.** Insert an edge from HIT_t to TAG_j
7: **For h** in h_i= {HIT vertices that were created from tokens_i} :
8: Compute z_i = {CLASS vertices that are connected to h via a path in the model graph}
9: **For z** in z_i :
10: **If z** belongs to annotation model X then $\text{count\_AM}_X = \text{count\_AM}_X + 1$
11: **Output** $\max(\text{AM}_X)$

3 is most expensive when an OLiA annotation model defines an annotation in terms of a partial match or (worse) a regular expression. Finally, in step 10 the found CLASS vertices are summed up with respect to the OLiA annotation model they belong to. In Fig. 2, the models SUC, BROWN, MAMBA, GENIA, QTAG and PENN would receive (+1) in step 10 of the algorithm.

```
HIT : DT matches CLASS : http://purl.org/olia/suc.owl#dt
HIT : DT matches CLASS : http://purl.org/olia/brown.owl#DT
HIT : DT matches CLASS : http://purl.org/olia/mamba-syntax.owl#determiner
HIT : DT matches CLASS : http://purl.org/olia/genia.owl#DT
HIT : DT matches ClASS : http://purl.org/olia/qtag.owl#DT
HIT : DT matches CLASS : http://purl.org/olia/penn.owl#DT
```

Figure 2 Different choices to match the tag DT for determiner.

3.2 Evaluation

Table 2 summarizes evaluation results for annotation model detection for four annotations columns (C-4, C-5, C-6, C-8) in 93 UD corpora. Again, the same test data was used. For Universal Dependencies corpora, we focus on parts of speech and dependency labels, i.e., CoNLL-U columns UPOS (UD-style parts of speech, column C-4), XPOS (native parts of speech, column C-5), FEATS (UD dependency labels, column C-6) and DEP (UD dependency labels, column C-8).

Throughout all datasets, UPOS, FEATS and DEP are correctly detected, for evaluating annotation model accuracy, we thus focus on C-5 (XPOS). A challenging aspect is that OLiA does not support all native tagsets for the 77 UD languages. As a measure to estimate the quality of a predicted annotation model for a CoNLL column *c* we introduce the *Coverage* measure which is defined by :

$$\text{Coverage}(\text{model}_X) = \frac{\#\text{ annotations in c found in OLiA annotation model X}}{\#\text{ annotations in c found in any OLiA annotation model}}$$

The Coverage measure ignores annotations that were not recognized in any OLiA annotation model.[13] As tags tend to re-occur in different tagsets, we applied a restrictive filtering to models with a Coverage of more than 95%, so that we achieved a precision of 81.25%.

[13] Note that this allows to identify missing annotations (tags) in OLiA annotation models. Unmatched annotations are displayed in the application user interface.

Table 2 Overview of model column detection (baseline 374 possible model columns).

Criterion/Comment	Columns with a predicted model
Model detected with Coverage > 95%	200/374 (53.4%)
Model detected with Coverage > 80%	314/374 (84%)
Model detected with Coverage ≤ 80%	32/374 (8.5%)
No annotation model was detected	28/374 (7.5%)
A text column was regarded as a model column	3

In cases where no annotation model could be detected, an appropriate OLiA annotation model was missing.[14] Since annotations are highly ambiguous (e.g. same tag can be used in multiple annotation models) other properties of these models need to be incorporated in the detection process. One possibility is to include language information into OLiA annotation models because many annotation models were specifically designed for certain languages. At the moment, such information is not provided by OLiA in order to support adaptations of existing annotation models to linguistically or culturally related language varieties.

4 Editor for manual curation and verification

Aside from the detection routines described above, the AnnoHub infrastructure provides a web front-end that features an editor for the interactive curation and verification of language and annotation model predictions. Its functionality is illustrated here with respect to annotation model detection. For language detection, analoguous views are provided. The model editor can be used to review the computed best fitting models for a CoNLL resource and to correct errors by selecting a different model manually. The editor window (Fig. 3) gives an overview of possible annotation models for a specific column in a CoNLL file. On top of the list the model with the best fitting for all tags in a column is listed. Further results include the following values : a) coverage of occurring tag types in %, b) the number of different found tags, c) the total number of instances for all tags in b), d) the number of tags types that are matched exclusively by this model, e) the total number of instances for all tags in d).

As an example consider the result for column 4 for the CoNLL file en_ewt-ud-train.conllu (Fig. 3). At the top of edit window the PENN annotation model is shown as the selected model for that column. Detailed results for each candidate model can be displayed by expanding a row in the table. In the example the results for the EMILLE annotation model are displayed. In the first column (Found tag/Class) the only part-of-speech tag CC that could be matched in the corpus is listed. In the second column a URL shows the ontology class were a definition for the part-of-speech tag CC can be found and the third column shows the number of found instances for that tag (74). Finally the entry ZERO MATCH displays those annotations that could not be found in the OLiA annotation model for EMILLE together with their count (31) in the last column.

[14] This includes cases where the XPOS column provided POS tags concatenated with other annotations, e.g., for grammatical features. With an OLiA annotation model expecting POS tags to occur in isolation, rule-based preprocessing of XPOS annotations is necessary to produce a match. This has not been attempted, so far.

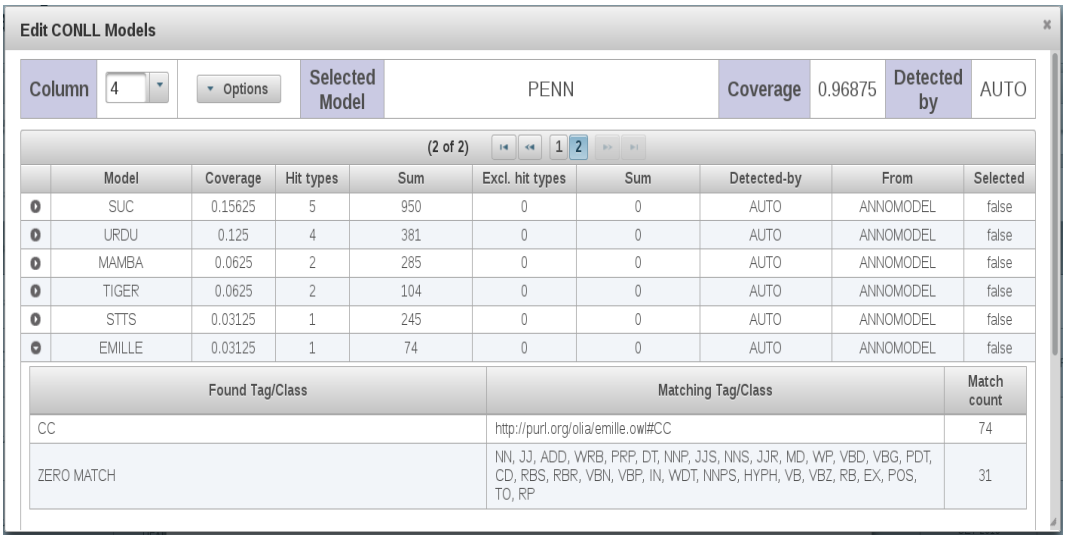

Figure 3 Edit details for the CoNLL file en_ewt-ud-train.conllu.

5 Results

Detailed results are shown in Tab. 3: The first three table columns comprise the results for language detection and the last four columns show the predicted annotation models (were C-4, C-5, C-6 and C-8 refer to the respective columns of a CoNLL file). For table column C-5 a restrictive filtering was applied to the detection results. It only shows those annotation models that could provide a coverage of more than 95%. The UD columns have coverage scores of 100% (C-4, C-6, UPOS and DEPS), resp., 83% – 100% (C-8, FEATS).

As an example consider the file ar_padt. In column 2 (WORD), the language was detected for which the corpus was also marked in the metadata (ara, Macro-Arabic [all varieties]), in column 3 (LEMMA), a different variety was predicted (arb, Standard Arabic), counting here as an error. However, as the first tag was correct, this counts as a partial match.[15] In column 4, the UD part-of-speech, in column 6 the UD features and in column 8 UD dependency labels were detected. Column 5 did not produce an annotation model with coverage greater than 95%. Columns 9 and following were generally excluded.

6 Summary and Outlook

We presented a method to analyze and to curate language resources with respect to their annotation schemes and languages. This functionality is provided as a component to faciliate for metadata indexing and search functionalities in an information system tailored for applications in the language sciences, where it will be applied to provide search beyond bibliographical references to relevant language resources. We specifically described the treatment of TSV formats as frequently used for corpora and provided an evaluation against the Universal Dependencies corpora. We are currently in the process of extending the described methods to CoNLL and TSV formats beyond the Universal Dependencies. In

[15] Of course, this is most likely not an error, as Standard Arabic is a variety of Arabic. But we ground our evaluation in the available metadata.

Table 3 Detailed prediction results for 93 UD corpora.

	Languages			Annotation Models			
corpus	**ISO 639-3**	**predicted**	**comment**	**C-4**	**C-5**	**C-6**	**C-8**
ar_nyuad	—	—	✓(no text)	UD	†	UD	UD
ar_padt	ara	ara,arb	partial match	UD	†	UD	UD
bxr_bdt	bxr	khk	no profile	UD	—	UD	UD
ca_ancora	cat	cat	✓	UD	UD	UD	UD
cop_scriptorium	cop	—	fail	UD	†	UD	UD
cu_proiel	chu	bul	no profile	UD	†	UD	UD
de_gsd	deu	deu	✓	UD	STTS	UD	UD
el_gdt	ell	ell	✓	UD	UD	UD	UD
en_esl	—	—	✓(no text)	UD	PENN	—	UD
en_ewt	eng	eng	✓	UD	PENN	UD	UD
en_gum	eng	eng	✓	UD	PENN	UD	UD
es_ancora	spa	spa	✓	UD	UD	UD	UD
fr_ftb	—	—	✓(no text)	UD	—	UD	UD
fro_srcmf	fro	fra	no profile	UD	†	UD	UD
fr_spoken	fra	fra	✓	UD	—	—	UD
gl_ctg	glg	glg	✓	UD	†	—	UD
grc_perseus	grc	grc,ell	partial match	UD	†	UD	UD
he_htb	heb	heb	✓	UD	UD	UD	UD
hi_hdtb	hin	hin	✓	UD	ANCORRA	UD	UD
hsb_ufal	hsb	pol	no profile	UD	—	UD	UD
ja_bccwj	—	—	✓(no text)	UD	—	—	UD
kk_ktb	kaz	bel	no profile	UD	†	UD	UD
ko_gsd	kor	kor	✓	UD	†	—	UD
ko_kaist	kor	kor	✓	UD	†	—	UD
no_nynorsklia	nor	nno,nob	weak match	UD	†	UD	UD
no_nynorsk	nor	nno	weak match	UD	—	UD	UD
ro_rrt	ron	ron	✓	UD	MULT	UD	UD
ru_gsd	rus	rus,bul	partial match	UD	PENN	UD	UD
sk_snk	slk	slk	✓	UD	MULT	UD	UD
sl_ssj	slv	slv	✓	UD	MULT	UD	UD
sl_sst	slv	slv	✓	UD	MULT	UD	UD
sme_giella	sme	prf	no profile	UD	†	UD	UD
sr_set	srp	hrv	fail	UD	—	UD	UD
swl_sslc	swl	ude	no profile	UD	†	—	UD
te_mtg	tel	tel	✓	UD	UD	UD	UD
ug_udt	uig	pes	no profile	UD	†	UD	UD
uk_iu	ukr	ukr	✓	UD	MULT	UD	UD
zh_gsd	zho	zho	✓	UD	PENN	UD	UD
55 other[1)]	correct		✓	UD		UD	UD

1) 55 corpora for 39 languages, afr, bel, bul, ces, dan, eng, est, eus, fas, fin, fra, gle, glg, got, grc, hrv, hun, hye, ind, ita, jpn, kmr, lat, lit, lav, mar, mlt, nld, nor, pol, por, ron, rus, spa, swe, tam, tur, urd, vie († marks a model coverage < 96%, — marks no language or model info in corpus)

addition, other corpus and dictionary formats will be supported, most noteably various XML formats. A generic XML converter/indexer is currently under development. The code of our detectors and the generated RDF metadata from this resources will be published in early 2020 under an open license[16].

[16] `https://annohub.linguistik.de`

References

1 Christian Chiarcos and Christian Fäth. CoNLL-RDF: Linked corpora done in an NLP-friendly way. In Jorge Gracia, Francis Bond, John P. McCrae, Paul Buitelaar, Christian Chiarcos, and Sebastian Hellmann, editors, *Language, Data, and Knowledge*, pages 74–88, Cham, Switzerland, 2017. Springer.

2 Christian Chiarcos, Christian Fäth, Heike Renner-Westermann, Frank Abromeit, and Vanya Dimitrova. Lin|gu|is|tik: Building the Linguist's Pathway to Bibliographies, Libraries, Language Resources and Linked Open Data. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, 2016. European Language Resources Association (ELRA).

3 Christian Chiarcos, Sebastian Nordhoff, and Sebastian Hellmann. *Linked Data in Linguistics*. Springer, 2012.

4 Christian Chiarcos and Maria Sukhareva. OLiA – Ontologies of Linguistic Annotation. *Semantic Web Journal,518:379–386*, 2015.

5 Christian Chiarcos, Maria Sukhareva, Roland Mittmann, Timothy Price, Gaye Detmold, and Jan Chobotsky. New Technologies for Old Germanic. Resources and Research on Parallel Bibles in Older Continental Western Germanic. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 22–31. Association for Computational Linguistics, 2014.

6 Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, 2016. European Language Resources Association (ELRA).

7 Andrea Zielinski and Christian Simon. Morphisto – An Open Source Morphological Analyzer for German. In *Proceedings of the 2009 Conference on Finite-State Methods and Natural Language Processing: Post-proceedings of the 7th International Workshop FSMNLP 2008*, pages 224–231, Amsterdam, The Netherlands, 2009. IOS Press.