Annotating a low-resource language with LLOD technology: Sumerian morphology and syntax

Christian Chiarcos, Ilya Khait, Émilie Pagé-Perron, Niko Schenk, Jayanth, Christian Fäth, Julius Steuer, William Mcgrath, Jinyan Wang

Angaben zur Veröffentlichung / Publication details:

Chiarcos, Christian, Ilya Khait, Émilie Pagé-Perron, Niko Schenk, Jayanth, Christian Fäth, Julius Steuer, William Mcgrath, and Jinyan Wang. 2018. "Annotating a low-resource language with LLOD technology: Sumerian morphology and syntax." *Information* 9 (11): 290. https://doi.org/10.3390/info9110290.

information

MDPI

Article

Annotating a Low-Resource Language with LLOD Technology: Sumerian Morphology and Syntax †

Christian Chiarcos [1], Ilya Khait [1], Émilie Pagé-Perron [2,3,*], Niko Schenk [1], Jayanth [2], Christian Fäth [1], Julius Steuer [1], William Mcgrath [3] and Jinyan Wang [3]

[1] Department of Informatik und Mathematik, Goethe University Frankfurt, D-60325 Frankfurt, Germany; chiarcos@informatik.uni-frankfurt.de (C.C.); khait@informatik.uni-frankfurt.de (I.K.); schenk@informatik.uni-frankfurt.de (N.S.); faeth@informatik.uni-frankfurt.de (C.F.); steuer@informatik.uni-frankfurt.de (J.S.)

[2] Department of Near Eastern Languages and Cultures, University of California Los Angeles, Los Angeles, CA 90095, USA; jayanthj@ucla.edu

[3] Department of Near and Middle Eastern Civilizations, University of Toronto, Toronto, ON M5S 1C1, Canada; bill.mcgrath@mail.utoronto.ca (W.M.); jinyan.wang@mail.utoronto.ca (J.W.)

* Correspondence: epp@ucla.edu; Tel.: +1-416-939-8173

† This paper is an extended version of our paper published in Christian Chiarcos, Ilya Khait, Émilie Pagé-Perron, Niko Schenk, Jayanth, and Lucas Reckling. "Annotating Sumerian: A LLOD-enhanced Workflow for Cuneiform Corpora" Proceedings of the 6th Workshop on Linked Data in Linguistics (LDL-2018): Towards Linguistic Data Science, Miyazaki, Japan, May 2018.

Received: 17 September 2018; Accepted: 25 October 2018 ; Published: 19 November 2018

Abstract: This paper describes work on the morphological and syntactic annotation of Sumerian cuneiform as a model for low resource languages in general. Cuneiform texts are invaluable sources for the study of history, languages, economy, and cultures of Ancient Mesopotamia and its surrounding regions. Assyriology, the discipline dedicated to their study, has vast research potential, but lacks the modern means for computational processing and analysis. Our project, Machine Translation and Automated Analysis of Cuneiform Languages, aims to fill this gap by bringing together corpus data, lexical data, linguistic annotations and object metadata. The project's main goal is to build a pipeline for machine translation and annotation of Sumerian Ur III administrative texts. The rich and structured data is then to be made accessible in the form of (Linguistic) Linked Open Data (LLOD), which should open them to a larger research community. Our contribution is two-fold: in terms of language technology, our work represents the first attempt to develop an integrative infrastructure for the annotation of morphology and syntax on the basis of RDF technologies and LLOD resources. With respect to Assyriology, we work towards producing the first syntactically annotated corpus of Sumerian.

Keywords: linked open data; linguistic linked open data; morphology; syntax; parsing; RDF; SPARQL; low-resource languages; Sumerian; Cuneiform

1. Introduction

The Sumerian language, an agglutinative isolate, is the earliest known language recorded in writing. It was spoken in the third millennium BC in southern Iraq, and continued to be written until the late first millennium BC. This language was written in cuneiform, a logo-syllabic script with around one thousand signs in its inventory, formed by impressing a sharpened reed stylus into fresh clay.

Assyriologists make a text available for research by first copying and transcribing it from the inscribed artifact. The results of this labor-intensive task are usually published on paper. A dozen projects which make various cuneiform corpora available on-line have emerged, building on digital transcriptions created as early as the 1960s. Unfortunately, these initiatives rarely use shared conventions, and the tool-set available

to process these data is limited, thus vast numbers of transliterated and digitized ancient cuneiform texts remain only superficially exploited.

Here, we employ Linguistic Linked Open Data (LLOD) technology to improve interoperability and resource integration for machine translation and linguistic annotation of Sumerian.

1.1. Linked Open Data for Sumerian

Linked Open Data (LOD) defines principles and formalisms for the publication of data on the web with the goal of facilitating its accessibility, transparency, and re-usability. Most importantly, the application of LOD formalisms to philological resources within the field of Assyriology promises to establish interoperability and exchange between distributed resources that currently persist in isolated data silos—or that provide human-readable access only, with no machine-readable content. In addition to that, Chiarcos et al. [1] also mentions federation, ecosystem, expressivity, and semantics by reference. Converting data to an RDF representation is an essential step to opening up the possibility of linking with other resources and integrating content from different portals. Further, using shared vocabularies allows us to publish structured descriptions of content elements in a transparent and well-defined fashion. Ontologies play a crucial role in this regard, as they define shared data models and concepts.

We have experimented linking our data and metadata with external dictionaries, metadata repositories, and museums in earlier research [2]. Here, we demonstrate that RDF technologies are also a suitable means for rule-based annotation transformation in, and annotation of low-resource languages. While this can also be accomplished with graph databases in general, we particularly benefit from W3C standardization, as this provides us with a rich technological ecosystem comprising various database implementations, APIs, and—most importantly—a standardized query language for the flexible querying and manipulation of our data [3], SPARQL. An important feature of SPARQL is that it allows us to freely port our code between different programming languages and database back-ends. Another significant aspect is that SPARQL 1.1 introduced the concept of property paths which permit the expression of iterated and alternative transition sequences in RDF graphs in a compact and generic fashion. Finally, SPARQL-based annotation provides the opportunity to consult LLOD resources during the transformation, e.g., dictionaries or terminology repositories, and, furthermore, the creation and manipulation of linguistic resources with native RDF technology motivates the publication, exchange and consumption of linguistic annotations on the web as Linguistic Linked Open Data.

Previously, Ref. [4] developed an ontology for representing Sumerian morphology, Ref. [5] designed an ontology-backed relation extraction system for Sumerian administrative texts, and [6] developed and applied the mORSuL ontology for the study of narrative structures in Sumerian. These experiments only attained the status of pilot experiments and case studies, yet they show the potential of, and interest in Sumerian corpus data being published in accordance with Semantic Web principles. Neither of these projects, however, has published Linked Data so far. However, Linked Data is being used in relation to Sumerian cuneiform for metadata and lexical data [2], so that the nucleus of a Sumerian Linked Open Data sub-cloud already exists, which may be extended with corpus data as a result of our activities.

1.2. The MTAAC Project

The "Machine Translation and Automated Analysis of Cuneiform Languages" (MTAAC) project (https://cdli-gh.github.io/mtaac) aims to develop state-of-the-art computational linguistics tools for cuneiform languages, using internationally recognized standards to share the resulting data with the widest possible audience [7]. This is made possible through a collaboration between the Cuneiform Digital Library Initiative (CDLI) (https://cdli.ucla.edu) and specialists in Assyriology, computer science and computational linguistics at the Goethe University Frankfurt, Germany, the University of California Los Angeles (UCLA) and the University of Toronto, Canada. The project develops a methodology and an NLP pipeline for Sumerian, with the goal to process, annotate and translate Sumerian texts, and to enable information extraction on this basis.

In order to facilitate the re-usability of these data, as well as to encourage reproducibility, we use linked data and open vocabularies, thereby contributing to interoperability with other portals addressing linguistically or historically related languages. (Including other cuneiform languages (ORACC, oracc.museum.upenn.edu), Syriac (http://syriaca.org), or Hebrew (http://tinyurl.com/guwe8kr).) Another aim in the application of LOD is to set new standards for digital cuneiform studies and to contribute to resolving data integration challenges, both in Assyriology and in related linguistic research. In our LOD edition for Sumerian language and object data, we build on CoNLL-RDF (Section 3.2) for corpus data, lemon/OntoLex for lexical data [8], CIDOC/CRM for object metadata [9], lexvo for language identification [10], Pleiades for geographical information [11], and OLiA for linguistic annotations [12]. Bringing these disparate strands of Assyriologically relevant resources together breaks new ground in the field of Assyriology, and in the digital humanities in general.

One objective of our project is to complement the range of cuneiform corpora with morphological, syntactic and semantic annotations for an extensive but currently under-translated genre, namely the administrative texts, especially those written in the Neo-Sumerian language of the Ur III period (2100–2000 BC). As our corpus comprises almost 70,000 texts, we provide manual annotations only for a core corpus. These data are then used to train NLP tools for the automated annotation and translation of the full corpus. As for manual annotation, this is supported by automated pre-annotation routines using RDF technology and LLOD resources. The present paper concerns this particular aspect of the pipeline.

2. Corpus Data

The MTAAC project works toward annotating 69,070 transliterated administrative and legal texts from the Ur III period, including 1966 that are already supplied with parallel English translations. This material is a subset of entries from the Cuneiform Digital Library Initiative (CDLI).

CDLI is a major Assyriological on-line project that aims to provide information on cuneiform inscriptions and the artifacts bearing them which are kept in museums and collections around the world. At the moment, the CDLI catalog contains entries for about 334,000 objects. Data made available by the CDLI include images, meta-data, transliterations, transcriptions, translations, bibliography, and soon also the annotations produced though the MTAAC project. As the basic format for storing unannotated textual data, CDLI uses C-ATF (Canonical ASCII Transliteration Format, see Figure 1): Numbers at the beginnings of lines with transliteration correspond to lines on the tablet; the data include structure tags, translation, and comments which adds to the content of each textual entry.

```
&P414545 = YOS 15, 173
#atf: lang sux
@tablet
@obverse
1. 9(disz) gu4-gesz
#tr.en: 9 plow-oxen,
2. 1(disz) ab2-mah2
#tr.en: 1 mature cow,
3. ki da-ge-ta
#tr.en: from Dage;
4. gu4 nig2-gur11 iszib {d}szul-gi-ra
#tr.en: the oxen are the property of the
#tr.en: incantation priest of Šulgi;
```

Figure 1. Ur III administrative text (C-ATF). Text by [13], CDLI entry by Robert K. Englund. https://cdli.ucla.edu/P414545.

The morphologically and syntactically annotated corpus of Ur III data developed by MTAAC is complemented (and partially builds on) earlier efforts in the linguistic annotation of Sumerian—albeit addressing different periods, genres and phenomena—, namely, the Electronic Text Corpus of Sumerian Literature (ETCSL) [14] and the Electronic Text Corpus of Sumerian Royal Inscriptions (ETCSRI) (http://oracc.museum.upenn.edu/etcsri/), which provide morphosyntactic annotations

only. To the best of our knowledge, that is also the state-of-the-art in other branches of Assyriology, where representative morphosyntactic annotations (glosses) have been assembled, for example, within the Open Richly Annotated Cuneiform Corpus (ORACC) (http://oracc.museum.upenn.edu) portal. Further (unannotated) Sumerian textual data is available from other projects, such as the Database of Neo-Sumerian Texts for Ur III administrative documents. (http://bdts.filol.csic.es/).

At the moment, the annotation of Sumerian with syntactic relations is limited to experimental pilot studies and there is no syntactically annotated corpus of Sumerian currently available. (The only existing cuneiform corpus with manual annotation of syntax is the Annotated Corpus of Hittite Clauses [15], however, this addresses another language. Experiments on the automated syntactic annotation of Sumerian cuneiform have been described by [5,16], but both focused on extracting automatically annotated fragments rather than on providing a coherently annotated corpus.)

A notable contribution in this direction, however, has been the Penn Parsed Corpus of Sumerian (PPCS), (See http://oracc.museum.upenn.edu/doc/help/languages/sumerian/syntax/index.html, official website (currently offline) archived under https://web.archive.org/web/20040906191032/http://psd.museum.upenn.edu:80/ppcs/). a pilot experiment in annotating Sumerian syntax. Although this project did not develop significant quantities of annotations—and the approach to syntactic parsing adopted here is radically different from the phrase structure grammar underlying this annotation effort—we benefited from their annotation guidelines and the examples that were analyzed in this context.

3. Technical Setup

3.1. CoNLL Format

Due to the specifics of our data, we have to extend existing representation formalisms. As the ATF format(s) does not allow to add another layer to retain, for example, annotation of syntax, we supplement it with a CoNLL format, a common community standard in NLP which provides a table of tab-separated values (TSV) for various annotations of one word per line. CoNLL formats have been used for many kinds of annotation, e.g., CoNLL-U for syntax in the context of the Universal Dependencies [17], UD, and they are thus well supported by annotation tools. (UD had been employed in relation to work on other low-resource dead languages such as e.g., Ancient Greek and Latin [18] or Coptic [19]. Further examples include Sanskrit, Gothic, Old Church Slavonic, Old French, and Akkadian (planned), see http://universaldependencies.org/.) Another advantage of CoNLL is its extensibility, genericity, and simplicity, allowing us to transform data from CDLI, ETSCRI and ETCSL (ATF, XHTML, JSON, XML/TEI) into CoNLL.

We introduce the CDLI-CoNLL format as a TSV format with seven columns: `ID`, `FORM`, `SEGM`, `XPOSTAG`, `HEAD`, `DEPREL`, `MISC`. In comparison to the widely used CoNLL-U format, (http://universaldependencies.org/format.html). CDLI-CoNLL is both more compact and more informative, but tailored to a specific use:

`ID` Unique identifier composed of side (o/r), line number, and token number.
`FORM` Transliteration of the token in C-ATF format.
`SEGM` Dash-separated morphological segmentation. Affix standardization follows ETCSRI. (http://oracc.museum.upenn.edu/etcsri/glossing/).
`XPOSTAG` Part-of-speech and morpheme glosses, sequentially aligned with `SEGM`. Tags mostly follow ETSCRI.
`HEAD` Head of the current token in dependency syntax, i.e., either its ID or 0 (for root).
`DEPREL` Dependency label of the relation in HEAD, following CoNLL-U specifications.
`MISC` Comments; other content.

We provide a conversion from CDLI-CoNLL to CoNLL-U, (https://github.com/cdli-gh/CDLI-CoNLL-to-CoNLLU-Converter.) see Figure 2 for a comparison.

```
#ID     FORM                    SEGM                XPOSTAG
o.0.4   szukuppak{ki}-ga-ke4    Szuruppag[1]-ak-e   SN.GEN.ERG

#ID     FORM                    LEMMA               UPOSTAG XPOSTAG FEATS
4       szukuppak{ki}-ga-ke4    Szuruppag[1]        PRON    SN      Number=Sing|Case=Gen.Erg|Animacy=Nhum
```

Figure 2. CDLI-CoNLL annotation compared to CoNLL-U, excluding syntax annotation (which is identical).

The CDLI-CoNLL `SEGM` column cannot be adequately expressed in CoNLL-U, but represents the basis for extracting the `LEMMA`. The original `XPOSTAG` includes information about the part-of-speech as well as morpheme glosses. The CoNLL-U `XPOSTAG` is restricted to part-of-speech; morpheme glosses are mapped to CoNLL-U `FEATS`. In the process, we lose the level of detail as well as information about the original morpheme order. Moreover, CoNLL-U conventions allow us to preserve only parts of the morphological information in CoNLL-U: the last word of a Sumerian noun phrase aggregates all case morphology (its own as well as that of its head), a phenomenon known as Suffixaufnahme. In this case (Figure 2), the place name *Shuruppak* is a genitive attribute of an ergative argument. It is thus inflected for *both* genitive (*-ak*) and ergative (*-e*). In CoNLL-U, multiple case marking is not foreseen, so that here, language-specific aggregate features for multiple cases is introduced. (This solution is problematic in that long chains of case markers can arise, and it is no longer possible to generalize over the resulting multitude of case features. Case combinatorics in the ETCSRI corpus yield 47 case chains resulting from only 15 case labels.) In addition, the SN tag marks the word as a site name, and we infer non-human animacy.

CoNLL-U requires a non-trivial mapping from `XPOSTAG` annotations to tags, features, and dependency labels according to the Universal Dependencies (UD) schema. (http://universaldependencies.org/u/dep). We adopt a Linked Open Data approach for this purpose: We provide and consult an OWL representation of the CDLI annotation scheme and its linking with UD POS, feature and dependency labels as part of the Ontologiexs of Linguistic Annotation [12], OLiA: Using SPARQL update, these ontologies are loaded, their hierarchical structure traversed by property paths, and the corresponding tags replaced.

We argue that the clear separation of (SPARQL) code and (OWL) data of different provenance (CDLI annotation model, UD annotation models, linking between both) facilitates the transparency, reproducibility, and reversibility of our mapping in comparison to direct replacement rules. (Mapping to morphological features is mediated by the Unimorph ontologies http://purl.org/olia/owl/experimental/unimorph/, which are linked via `skos:broader` (etc.) statements with the concepts in the CDLI annotation scheme, and inherit their UD interpretation from OLiA).

In addition to a CoNLL-U conversion, CDLI-CoNLL can also be converted to the Brat Standoff format for further syntactic annotation, visualization (Figure 3), or applying other tools geared to processing data in this format.

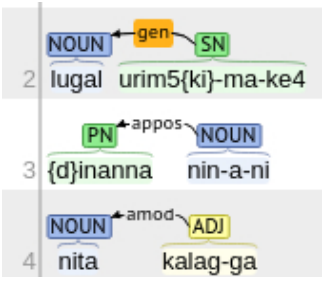

Figure 3. Brat annotation example.

3.2. CoNLL-RDF

CoNLL-RDF [20] provides a generic rendering of CoNLL data structures in RDF as well as a convenient and human-readable representation that structurally resembles CoNLL TSV but can be

directly processed as RDF/Turtle. Crucially, it is comparably easy to read and parse as CoNLL: it provides the direct means to string-based manipulations that CoNLL is praised for, but in addition it allows us to seamlessly integrate LOD resources and use graph transformation to process, manage, and manipulate CoNLL data with off-the-shelf technologies [21].

CoNLL-RDF APIs provide a means to convert from and to CoNLL on a sentence-by-sentence basis. This allows us to easily reorganize, add, or drop CoNLL columns, but also to apply sequences of SPARQL updates to every individual sentence. CoNLL-RDF supports iterations over SPARQL updates, as well as the consultation of external (LOD) resources during processing. In this way, sentence graphs can be flexibly transformed, and subsequently serialized as CoNLL-RDF, CoNLL-TSV or in other formats. As sentences are processed individually, even large-scale corpora can be processed on small workstations.

In the context of our corpus annotation workflow, CoNLL-RDF is primarily used as an internal format for transformations between CDLI-CoNLL and CoNLL-U, and for parsing and for pre-annotation with SPARQL [3], cf. Sections 4.2 and 5.1, but it can also serve as a release format, thus for the publication of annotated corpora as linked data.

In application to a specific CoNLL file, every word receives a URI (from a user-provided basename and the `ID` column), every column is represented as a property (in the `conll:` namespace, using a user-provided label), and its annotation is represented as a value. The column `HEAD` receives special handling and yields pointers to (the URI of) another word. Words and sentences are defined with the NIF vocabulary. (http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core). For CDLI-CoNLL, we thus produce the properties `conll:WORD` (for `FORM`), `conll:SEGM`, `conll:XPOSTAG`, etc. One advantage of CoNLL-RDF is that it allows us to handle annotations independently from the specifics of the format (e.g., the order of columns). As such, syntax annotations in CoNLL-U and CDLI-CoNLL can be processed with the same workflow, even though `HEAD` is the 5th column in CDLI-CoNLL but the 7th in CoNLL-U. This transition only requires the user to provide the appropriate column (i.e., property) names.

3.3. Annotation Workflow

The annotation workflow is shown in Figure 4. As explained in Section 2, the raw data entering the pipeline comprise unannotated textual data in the ATF format. ATF data will be validated, converted to CDLI-CoNLL, and fed into morphological pre-annotation (Section 4.1). A human annotator verifies and corrects the annotations and fills in the lines left incomplete. The resulting file is validated again, and then stored in the database.

Morphologically annotated CDLI-CoNLL data are subject to the syntax pre-annotation (Section 5.1), the resulting data are serialized as CoNLL-U, and converted to the Brat format. The human annotator can then finalize the syntactic annotation of the text using the CDLI Brat server interface. The completed Brat annotation is converted back to CoNLL-U, and the resulting file is fused with the original CDLI-CoNLL file using CoNLL-Merge [21]. (CoNLL-Merge is designed for the robust integration of conflicting CoNLL annotations of the same source file. It performs a word-level diff on the `FORM` column. Beyond merely identifying mismatches, it also provides heuristic but robust merging strategies in case a mismatch occurred, e.g., if a word has been split, two words have been merged, or deletions or additions occurred).

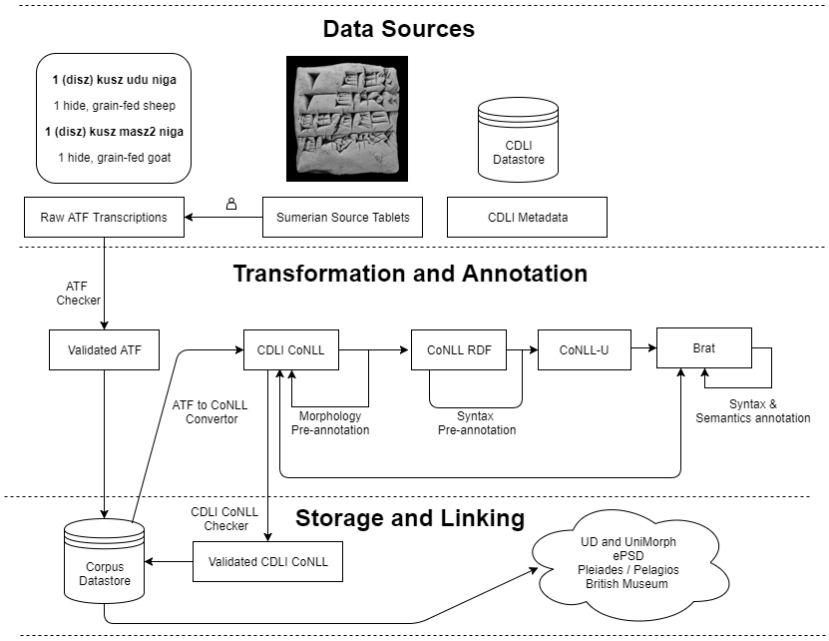

Figure 4. Corpus annotation pipeline: From ATF to RDF. Illustrated for https://cdli.ucla.edu/P322539, an Ur III cuneiform text from Garshana, Mesopotamia [22], no. 851; picture reproduced here with the kind permission of David I. Owen.

Only the ATF and CDLI-CoNLL versions of the data are kept in the datastore as we can easily convert the CDLI-CoNLL format to CoNLL-U and CoNLL-RDF formats, according to need. While both will be important publication formats to facilitate usability and re-usability of our data, they will only be generated on demand. We are, however, exploring options to offer CoNLL-RDF as a dynamic view on the internal (relational) database via technologies such as R2RML [23].

4. Annotating Morphology

The only publicly available system that performs an automated morphological and morphosyntactic annotation of Sumerian is represented by the ORACC lemmatizer [24], a lookup-based system that uses a dictionary of previously annotated words to suggest a possible analysis. The ORACC lemmatizer is, however, firmly integrated into the ORACC infrastructure and cannot be run as an independent application. For the MTAAC workflow, we thus provide an independent implementation of this procedure.

4.1. Dictionary-Based Pre-Annotation

As part of the pipeline, we provide a dictionary-based pre-annotator to improve the speed and internal consistency of manual annotation. Using a frequency dictionary of previous annotations of the current word, it provides the most frequent annotation associated with the form. For example, a text could contain the form *ensi2* ("ruler"), without attached morphemes. Possible analyses include `N` (noun, no case, e.g., because it is followed by a nominal modifier that carries the case information) or `N.ABS` (noun, absolutive case [without morphological marking]). These and other variant analyses of the form encountered so far are stored in a frequency dictionary.

When the pre-annotation tool encounters the form *ensi2* while pre-annotating a text, it will add the most frequent analysis in the appropriate `SEGM` and `XPOSTAG` fields. The other choices are appended in subsequent columns so the human editor can easily copy and paste another option into the appropriate fields, if required. The additional columns are removed by the "formatter" function of the tool. The pre-annotation tool adds new entries to the dictionary on demand.

4.2. Rule-Based Pre-Annotation with SPARQL

Dictionary-based pre-annotation leads to a speed-up in manual annotation, but it does not generalize beyond forms previously encountered in the annotation process. To facilitate the annotation of previously unseen forms, we developed an experimental prototype for the rule-based annotation of Sumerian morphology and morphosyntax. Earlier work in this direction includes Tablan et al. [25], and we would like to thank Hamish Cunningham, Valentin Tablan and Angus Roberts for sharing code and data with us. Unfortunately, the transcription and morphological annotation principles follow ETCSL conventions rather than the CDLI/ETSCRI conventions adopted in MTAAC, so that it could not be directly applied to our data.

A novel feature of our implementation is that it is based on CoNLL-RDF and SPARQL rather than strings and transducers, and a prospective advantage of doing so is that other forms of annotation and dictionary data can be more easily integrated during or after morphological annotation. Furthermore, RDF's graph structure allows us to build multiple interconnected syntactic or morphological trees without having to modify the underlying CoNLL format.

Taking the `conll:FORM` *szuruppak{ki}-ga-ke4* (Figure 2) as an example, we define this to be a 'trunk', i.e., a morphologically unanalyzed word form in the morph namespace:

```
<INSERT> {
    [] a morph:TRUNK;          # blank node of type morph:TRUNK
       rdfs:label ?string;     # with the simplified string szuruppak{ki}gake
       morph:SOURCE ?word.     # linked with the original CoNLL word
} WHERE {
  ?word conll:FORM ?form.      # original string szuruppak{ki}-ga-ke4
  BIND(replace(?form,'[0-9\-]','') as ?string)
}
```

The trunk is assigned the simplified transliteration as label, with sign separators, determinatives and numerical indices dropped (here, *szuruppakgake*). The property `morph:SOURCE` links the trunk with the word from which it originates. Depending on the original parts of speech, every initial trunk is then assigned a POS tag, e.g., `morph:TPOS "N"` for nominals. In the absence of POS information, we create one trunk for each possible POS. Every trunk thus represents a different morphological interpretation.

The initial trunks represent the basis for subsequent parsing rules. The following SPARQL Update rule, for example, separates the ergative morph(eme)-*e* from a nominal (`morph:TPOS "N"`) trunk and creates a subtrunk that contains the unanalyzed parts of the original string:

```
INSERT {
 [] a morph:TRUNK; morph:TPOS "N";
    rdfs:label ?substring; # here: szuruppak{ki}gak
    morph:SOURCE ?trunk;
    morph:NEXT [
      a morph:MORPH; rdfs:label "-e"; conll:SUFFIX "ERG"
    ]
} WHERE {
  ?trunk a morph:TRUNK; morph:TPOS "N"; # a nominal trunk
  rdfs:label ?string.                   # here: szuruppak{ki}gake
  FILTER(strends(?string,'e'))
  BIND(replace(?string,'e$','') as ?substring)
}
```

The order between the unanalyzed subtrunk and the morph(eme) is represented by `morph:NEXT`. The `morph:SOURCE` relation connects a trunk with the trunk from which it has been derived. SPARQL Update rules are applied in a sequential order, reflecting the agglutinative nature of Sumerian morphology, and can modify *any* trunk, including trunks which already have another morphological analysis. Different rules can produce multiple trunks pointing to the same `morph:SOURCE` object; they thus represent alternative analyses. In this way, trunks form a tree structure that contains all possible analyses. From the final tree, all generated analyses are aggregated by means of `GROUP_CONCAT` and stored in `conll:MORPH2`. From the possible analyses, an annotator may then choose one possibility.

4.3. Application and Evaluation

In the current annotation workflow, we employ dictionary-based pre-annotation only. In an evaluation performed on the ETSCRI corpus (Table 1), using a corpus of 1000 tokens as the training set and 2000 tokens as the test set, the dictionary-based pre-annotator produced correct predictions in 48.0% of the cases, made no prediction for 50.4%, and incorrect predictions for 1.7%. These rates can be increased to an accuracy above 70% (for 5000+ annotated words), but improvements beyond that slow down, also because of the rising number of incorrect predictions. This confirms the inherent limitations expected for dictionary-based pre-annotation, and has been one motivation for developing a rule-based component.

Table 1. Evaluating dictionary-based morphological pre-annotation on ETSCRI.

Training Set (Tokens)	Predictions (% of 2000 Tokens)		
	Correct	None	Incorrect
1000	48.0	50.4	1.7
2000	63.9	33.3	2.8
5000	71.9	19.7	8.5
10,000	77.7	16.9	5.5
15,000	81.7	12.1	6.3

The rule-based pre-annotation is at an experimental stage only and has been implemented with a focus on nominal morphology. Preliminary evaluation results on 10 sample texts from the ETSCRI corpus show that 47.3% of the generated analyses contain the correct annotation. With rule-based annotation, we can thus expect to reduce the number of no-predictions by at least 50%. However, the rule-based annotation provides no disambiguation at the moment, thus it over-generates massively. For the future, we can expect the ePSD2 dictionary to become available in a LLOD edition (pers. comm. Steve Tinney). As it provides frequency information, this can be used to assess rule probability during SPARQL transformation.

The coverage gaps of rule-based annotation are mostly due to the under-specified nature of cuneiform orthography, where phonemes or entire morphemes may just be omitted in writing. This can be compensated by further extending the rule inventory and permitting rules with empty morphs (for non-written morphemes).

A limitation shared by dictionary-based and rule-based approaches for morphological pre-annotation is their limited awareness of context. Since a word can have different meanings, identifying the right one requires an awareness of the context. The same problem occurs when dealing with forms where case markers were not written; they must be inferred based on the analysis of the whole sentence, or in the case of the Ur III administrative texts, the order of words, since it is often stereotyped. To counteract those limitations, the human annotator analyzes the text and corrects and refines the generated annotations.

5. Annotating Syntax

At the time of writing, no syntactically annotated corpora of Sumerian are in existence. Pilot experiments on rule-based parsing have been described by Jaworski [26], on rule-based parsing and Tinney [27], on manual annotation, but they are limited in coverage and no annotated data have been released.

5.1. RDF-Based Pre-Annotation

In our CoNLL-RDF-based pre-annotation pipeline, we adopt Shift-Reduce terminology [28], 100–104. However, we model SHIFT and REDUCE as RDF properties that result from parsing operations, rather than these parsing operations themselves. The sequential order of tokens or partial parses is no longer maintained by ‘stack’ or ‘queue’ data structures but by explicit SHIFT relations

which are inserted for every `nif:nextWord` property in the graph. The initial 'queue' of partial parses thus reflects the word order of a sentence.

Each further parsing step then applies language-specific rules in a bottom-up fashion (instead of left-to-right as in classical Shift-Reduce parsing). Rules remove corresponding parses from the 'queue' by deleting their SHIFT relations and replacing them by REDUCE relations with the respective head of the parse. The head is then connected to the parses' SHIFT-precedent, or successor, thus restoring the sequence of the SHIFT 'queue'. With any remaining SHIFT relations of the reduced elements being transferred to the (partial) parse, the sequence of SHIFTs takes over the functions of the traditional 'queue' and the traditional 'stack' at the same time, but elements are processed regardless of their sequential order; instead, the order of parsing rules plays a decisive role in the parsing process.

Our parser uses CoNLL-RDF update to execute and iterate SPARQL updates rules in a pre-defined order until no further transformations occur, i.e., because a single root for the sentence has been established. In the end, the remaining SHIFT transitions are removed. The REDUCE relations now connect elements with their respective head and are therefore replaced by `conll:HEAD` properties.

For a moderate-scale rule set, SPARQL updates are convenient to write and manage. The resulting parser is simple, deterministic and non-lexicalized, and thus not sufficiently precise for automated annotation. Yet, it is sufficient to produce baseline parse trees for subsequent manual correction. With just a handful of rules, it can thus be used for effective *pre*-annotation: (Abbreviations follow Universal Dependencies; SHIFT and REDUCE relations are designated by whitespace (left) and arrow (right) respectively.

1. Reduce adjective to preceding noun with adjectival modifier relation:
   $\mathrm{NOUN}_0\ \mathrm{ADJ}_{\mathrm{CASE}} \Rightarrow \mathrm{NOUN}_{\mathrm{CASE}} \xleftarrow{\mathrm{amod}} \mathrm{ADJ}$
   e.g., nita $\xleftarrow{\mathrm{amod}}$ kalag-ga "strong male".

2. Reduce noun in the genitive to preceding noun with genitive modifier relation:
   $\mathrm{NOUN}\ \mathrm{NOUN}_{\mathrm{GEN}} \Rightarrow \mathrm{NOUN} \xleftarrow{\mathrm{GEN}} \mathrm{NOUN}$
   e.g., lugal $\xleftarrow{\mathrm{GEN}}$ $\mathrm{urim_5}^{\mathrm{ki}}$-ma "king of Ur".

3. Reduce noun with case marker to preceding noun with no case marker with appositional modifier relation:
   $\mathrm{NOUN}_0\ \mathrm{NOUN}_{\mathrm{CASE}} \Rightarrow \mathrm{NOUN}_{\mathrm{CASE}} \xleftarrow{\mathrm{appos}} \mathrm{NOUN}$
   e.g., $^{\mathrm{d}}\mathrm{inana}_{\mathrm{DAT}}$ $\xleftarrow{\mathrm{appos}}$ nin-a-ni "to Inanna, his lady".

4. Reduce noun to preceding noun with case relation:
   $\mathrm{NOUN}_0\ \mathrm{NOUN}_{\mathrm{CASE1+CASE2}} \Rightarrow \mathrm{NOUN}_{\mathrm{CASE1}} \xleftarrow{\mathrm{CASE2}} \mathrm{NOUN}$
   e.g., $\mathrm{lugal}_{\mathrm{ERG}}$ $\xleftarrow{\mathrm{GEN}}$ $\mathrm{urim_5}^{\mathrm{ki}}$-ma-$\mathrm{ke_4}$ "king of Ur".

5. Reduce noun to preceding numeral with numeral modifier relation:
   $\mathrm{NUM}_0\ \mathrm{NOUN}_{(\mathrm{CASE})} \Rightarrow \mathrm{NUM}_{(\mathrm{CASE})} \xleftarrow{\mathrm{nummod}} \mathrm{NOUN}$
   e.g., 3(u) $\xleftarrow{\mathrm{nummod}}$ $\mathrm{sila_3}$ "thirty sila (measuring unit)"

6. Reduce noun in case to following verb with absolutive relation:
   $\mathrm{NOUN}_{\mathrm{ABS}}\ \mathrm{VERB} \Rightarrow \mathrm{NOUN} \xrightarrow{\mathrm{ABS}} \mathrm{VERB}$
   e.g., numun-na-ni $\xrightarrow{\mathrm{ABS}}$ $\mathrm{he_2}$-eb-til-le-ne
   "may they end his lineage".

7. Reduce noun in case to following verb with case relation:
   $\mathrm{NOUN}_{\mathrm{CASE}}\ \mathrm{VERB} \Rightarrow \mathrm{NOUN} \xrightarrow{\mathrm{CASE}} \mathrm{VERB}$

The case features employed as dependency labels are subsequently replaced by nsubj, obj, obl and nmod, and thus they are sufficient for the structure of clauses. In addition to morphology-driven rules, administrative texts require special handling for a number of frequent patterns:

8. Reduce a sequence of numerals to the first:
   $\mathrm{NU}\ \mathrm{NU} \Rightarrow \mathrm{NU} \xleftarrow{\mathrm{nummod}} \mathrm{NU}$

9. Render mathematical operators as prepositions: (Note that rule 9 extends beyond the Shift-Reduce framework by considering non-adjacent elements.)
   $\mathrm{NU}\ \mathit{minus}\ \mathrm{NU} \Rightarrow \mathrm{NU} \xleftarrow{\mathrm{nummod}} (\mathit{minus} \xrightarrow{\mathrm{case}} \mathrm{NU})$

10. A numeral interval after time unit (day, month, or year) is analyzed like its numeral modifier — *year* NU $\Rightarrow$ *year* $\xleftarrow{\text{nummod}}$ NU

11. Reduce a numeral to its unit of measurement: — NU N $\Rightarrow$ NU $\xrightarrow{\text{nummod}}$ N

Finally, a generic fall-back rule applies that considers unattached post-nominal elements as appositions:

12. Reduce post-nominal elements to the nominal: — N X $\Rightarrow$ N $\xleftarrow{\text{appos}}$ X

Then, case labels are mapped to UD dependencies, and in a final processing step, REDUCE relations are converted to `conll:HEAD` references between individual words; the root node(s) receive the URI of the local sentence as `conll:HEAD`. When exported to CoNLL TSV, the URIs are replaced by the word IDs (or 0 for the sentence).

All graph-rewriting rules are implemented in SPARQL Update, (The full code is available from https://github.com/cdli-gh/mtaac_work/tree/master/parse.) as illustrated in Figure 5. An example of the output of the syntactic pre-annotation for a Sumerian royal inscription is provided below in Figure 6.

```
DELETE {
# update SHIFT `queue'/`stack'
    ?noun conll:SHIFT ?adj.
    ?adj conll:SHIFT ?next.
} INSERT {
# add dependency
    ?adj conll:REDUCE ?noun.
    ?adj conll:EDGE "amod".
    ?noun conll:CASE ?case.
# update SHIFT `queue'/`stack'
    ?noun conll:SHIFT ?next.
} WHERE {
# a noun or proper noun
    ?noun conll:POS
    ?pos FILTER(strends(?pos,"N")).
# that is uninflected for case
    MINUS { ?noun conll:CASE [] }
# precedes
    ?noun conll:SHIFT ?adj.
# a nominalized verb
    ?adj conll:NOM [] .
# case and context (SHIFT) update)
    OPTIONAL { ?adj conll:SHIFT ?next }
    OPTIONAL { ?adj conll:CASE ?case. }
};
```

Figure 5. Adjective attachment rule in SPARQL Update, corresponding to NOUN_0 $\text{ADJ}_{\text{CASE}} \Rightarrow \text{NOUN}_{\text{CASE}} \xleftarrow{\text{amod}}$ ADJ.

We estimate that this method can be efficiently used for pre-annotation of dependency syntax; however, one cannot fully rely on its unsupervised result: mistakes and ambiguities are expected and these have to be resolved manually.

```
s1_1 . . / DAT-H---- an          BASE an GW 1 ID 1
                                 MORPH2 N1=NAME POS DN HEAD 10
s1_2 . . . \ appos-- lugal       BASE lugal GW king ID 2
                                 MORPH2 N1=STEM POS N HEAD
s1_3 . . . . \ GEN-- dijir-re-ne BASE dijir GW deity ID 3
                                 MORPH2 N1=STEM.N4=PL.N5=GEN POS N HEAD
s1_4 . . . \ appos-- lugal-a-ni  BASE lugal GW king ID 4
                                 MORPH2 N1=STEM.N3=3-SG-H-POSS.N5=DAT-H POS N HEAD 1
s1_5 . . / ERG------ ur-{d}namma BASE ur-{d}namma GW 1 ID 5
                                 MORPH2 N1=NAME POS RN HEAD 10
s1_6 . . . \ appos-- lugal       BASE lugal GW king ID 6
                                 MORPH2 N1=STEM POS N HEAD
s1_7 . . . . \ GEN-- urim5{ki}-ma-ke4 BASE urim5{ki} GW 1 ID 7
                                 MORPH2 N1=NAME.N5=GEN.N5=ERG POS SN HEAD 6
s1_8 . . / ABS------ kiri6       BASE kiri6 GW orchard ID 8
                                 MORPH2 N1=STEM POS N HEAD 10
s1_9 . . . \ amod--- mah         BASE mah GW great ID 9
                                 MORPH2 NV2=mah.N5=ABS POS V/i HEAD 8
s1_10 . \ mu-na-gub              BASE gub GW stand ID 10
                                 MORPH2 V4=VEN.V6=3-SG-H.V7=DAT.V11=3-SG-H-A.V12=gub.V14=3-SG-P POS V/i HEAD
s1_11 . . / ABS------ barag      BASE barag GW dais ID 11
                                 MORPH2 N1=STEM.N5=ABS POS N HEAD 14
s1_12 . . / L2-NH---- ki         BASE ki GW place ID 12
                                 MORPH2 N1=STEM POS N HEAD 14
s1_13 . . . \ amod--- sikil-la   BASE sikil GW pure ID 13
                                 MORPH2 NV2=STEM.N5=L2-NH POS V/i HEAD 12
s1_14 . \ mu-na-du3              BASE du3 GW build ID 14
                                 MORPH2 V4=VEN.V6=3-SG-H.V7=DAT.V11=3-SG-H-A.V12=STEM.V14=3-SG-P POS V/t HEAD 0
```

Figure 6. Syntactic pre-annotation of the Ur-Namma 5 inscription (http://oracc.iaas.upenn.edu/etcsri/Q000937/html).

5.2. Application and Evaluation

Manual annotation of the syntax is greatly simplified with the application of the pre-annotation tool. Using Brat, a human annotator must first verify that annotations generated by the pre-annotation tool are correct. When an annotation is faulty, the annotator removes the annotation and creates the appropriate one instead. Navigating the Brat interface is made easy as we modified the GUI to necessitate fewer clicks for each task. Finally, missing relationships must be added. Figure 3 shows a screenshot of three examples of relationships between words. Clicking on one term and then another one opens up a panel for choosing the nature of the relationship and creates it upon confirmation; selecting a word or a relationship and pressing <DEL> removes the annotation.

5.3. Limits of Syntactic Pre-Annotation

Our implementation is not a fully-featured parser, but a simple deterministic and greedy algorithm to assist manual annotation. Yet, for a sample of 25 tablets with 442 words from ETSCRI, we found that 75.3% tokens (333/442) had correct HEAD assignment (unlabelled attachment score); out of these, 88.8% (296/333) carried the correct UD label.

For certain complex cases, however, syntactic pre-annotation analysis does systematically fail:

1. Nominal clause. Clauses that do not contain an independent verbal form might not be parsed correctly in some cases

   $urdu_2$ lu_2-še lugal-zu-u_3
   slave man=that=ABS master=your=ABS
   'Slave! Is that man your master?' [29], 716, no. 7

2. Word order. Sumerian normally has an SOV word order, with the verb at the final position. However, exceptional right-dislocated clauses are known. Clause boundaries will not be correctly recognized in such cases.

   i_3-ĝu_{10} i_3-gu_7-e d nisaba-ke_4
   fat=my=ABS VP-eat -3SG.A:IPFV Nisaba =ERG
   'She will eat my cream, Nisaba.' [29], 300, no. 27

3. Enclitic copula. The Sumerian copula *me* can be both independent and enclitic. In the latter case, the analysis of the token in the context of other words is ambiguous, as it contains both nominal and verbal annotation:

   še dub-sar-ne-kam
   barley scribe =PL =GEN=ABS=be:3N.S
   'This is barley of the scribes.'

   nagar-me-e$š_2$
   carpenter=ABS=be -3PL.S
   'They are carpenters.' [29], 681-2, nos. 24 and 27

4. Enclitic possessive pronouns. To facilitate subsequent dependency parsing, enclitic possessives are analyzed in terms of their *morphosyntactic* characteristics, not on grounds of their *semantics*: In their function, enclitic possessives are referential and this could be explicitly expressed with links between possessor and possessum within UD using the language-specific but popular `nmod:poss` relation. However, such links cannot be easily integrated into UD-compliant syntactic annotation as it may easily lead to non-projective trees (i.e., crossing edges):

   sipa-de$_3$-ne / gu$_2$-ne-ne-a / e-ne-ĝar
   shepherd=PL =DAT neck=their =LOC VP-3PL.OO-3SG.A-place-3N.S/DO
   'He placed this (as a burden) on the shepherds, on their necks.' [29], 686, no. 21a

   In this example, the locative argument syntactically depends on the verb; at the same time, the enclitic possessive (glossed as 'their') refers to the preceding argument. Therefore, these semantic relations are to be captured in a subsequent processing step akin to anaphor resolution in other languages.

It is to be noted that the bulk of these grammatical elements occurs very rarely in Ur III administrative texts and royal inscriptions.

6. Beyond Syntax

In the preparation of future applications in prosopographical studies and information extraction, the scope of our projects extends beyond mere annotation.

6.1. Annotating Semantics

In a Google Summer of Code project advised by CDLI and the MTAAC project, Bakhtiyar Syed conducted initial experiments on creating semantic role annotations for Sumerian. Aside from Hayes [30], who used semantic roles as a didactic device in his 'Manual of Sumerian grammar and texts', we are not aware of any previous application of semantic roles to Sumerian. We employed English translations of Sumerian texts (taken from CDLI and ETCSL) to annotate these with existing semantic role-labelling systems for English [31], to align these translations with (normalized) Sumerian transliterations, and to project these annotations onto Sumerian. As a result, we obtained a corpus of 44,326 Sumerian tokens with 8017 (verbal and nominal) predicates and 7301 arguments, represented in a CoNLL format. These projected annotations are under evaluation in the preparation of experiments towards automated semantic role labeling. The outcome of such a system can be integrated into the existing MTAAC workflow, e.g., as a factor in syntactic pre-annotation, as CoNLL-RDF supports the CoNLL-specific representation formalisms for semantic roles.

6.2. Machine Translation

The goal of the MTAAC project is to facilitate machine translation of and information extraction from Sumerian texts. Whereas this paper focuses on linguistic annotation as a necessary prerequisite for information extraction, annotations can also be used to facilitate machine translation. So far, we have established statistical [32] and neural machine translation [33] baselines operating on normalized plain text. On our data, both kinds of systems suffer from sparsity issues, partly arising from the morphological richness of Sumerian, partly reflecting the challenges of the writing system. It should be noted, however, that the comparatively regular structure of administrative texts provides a suitable basis for classical transfer-based machine translation, so that syntactic (and semantic) parses can directly feature in the machine translation process.

7. Summary

This paper describes work on the morphological and syntactic annotation of Sumerian cuneiform as a model for low resource languages in general. Our contribution is two-fold: in terms of language technology, our work represents the first attempt to develop an integrative infrastructure for the annotation of morphology and syntax on the basis of RDF technologies and LLOD resources. With respect to Assyriology, we work towards producing the first syntactically annotated corpus of Sumerian.

The workflow that brings ATF raw textual data to publication as Linked Open Data, and the pipeline for text annotation—in particular the annotation of morphology and syntax—described in this paper, offers a roadmap for further development in the processing and analysis of ancient cuneiform languages. Improving and automating the annotation process for Sumerian sources is foundational for future work on cuneiform corpora, while the generation of annotations using a semi-automated annotation process for Sumerian syntax is generally unprecedented and innovative. We find the implementation of new standards for Assyriology as a digital discipline hardly meaningful without compatibility with existing LLOD standards on the one hand, and their adaptation to the particular languages and the material under scrutiny on the other, hence the choice of the CoNLL formats, RDF, UD, and the CIDOC-CRM. Building the machine translation pipeline for Sumerian, the ultimate goal of the MTAAC project, is greatly dependent on this work.

In all, these are crucial steps towards LLOD editions of Sumerian and other cuneiform languages. We hope that our work will help to provide Assyriogists and researchers from other fields with new and open annotated textual datasets, and a reusable infrastructure that can contribute significantly to the study of ancient languages and cultures.

The codes for format conversion, validation, dictionary-based pre-annotation and syntactic pre-annotation that we are designing for this pipeline are available in repositories kept under the CDLI organization page on Github, https://github.com/cdli-gh.

Author Contributions: C.C. conceived and designed the experiments; E.P., J.W., W.M., J. and I.K. prepared the data; C.C., C.F., I.K., N.S., E.P., and J. contributed analysis tools; C.C., É.P.-P. and J.W. performed the experiments in Section 4.1; J.S. performed the experiments in Section 4.2; C.C., I.K. and É.P.-P. performed the experiments in Section 5; N.S. and I.K. performed and supervised the experiments in Section 6; É.P.-P. and I.K. designed and supervised the manual annotation; É.P.-P., I.K., J.W. and W.M. analyzed the data; C.C., É.P.-P., I.K., C.F., N.S. and J. wrote the paper.

Funding: This research was funded by the Deutsche Forschungsgemeinschaft grant number HJ-253601-17, the Social Sciences and Humanities Research Council of Canada grant number HJ-253601-17, and the National Endowment for the Humanities grant number HJ-253601-17, through the T-AP Digging into Data Challenge (https://diggingintodata.org/.) The project is also supposed by the Humanities Division at the University of California Los Angeles. The research of Christian Chiarcos has been partially supported by the German Federal Ministry of Education and Research (BMBF) in the context of the Early Career Research Group 'Linked Open Dictionaries (LiODi)'.

Acknowledgments: Our appreciation goes to Heather D. Baker and Robert K. Englund for their insights and suggestions. We thank our student assistants at the Goethe University Frankfurt, Mohamed Boudan and Florian Stein, for their contribution to the code and to the analysis of the data. Bakhtiyar Syed, our GSoC student at IIIT Hyderabad, who worked on SRL projection for our materials, deserves special thanks. Thanks go to Maria Sukhareva and Lucas Reckling, who contributed to MTAAC at an early stage, and Graduate Student Researchers at UCLA who contributed to the manual annotation pipeline: Prashant Rajput, Shraddha Manchekar, and Anoosha Sagar. We also thank Gábor Zólyomi for sharing his data under a creative common license and giving us permission to release derivative work under the Public Domain, and Steve Tinney for his collaboration and efforts towards open and linked open data through ORACC.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chiarcos, C.; McCrae, J.; Cimiano, P.; Fellbaum, C. Towards Open Data for linguistics: Linguistic Linked Data. In *New Trends of Research in Ontologies and Lexical Resources: Ideas, Projects, Systems*; Oltramari, A., Vossen, P., Qin, L., Hovy, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 7–25. doi:10.1007/978-3-642-31782-8_2.
2. Chiarcos, C.; Pagé-Perron, É.; Khait, I.; Schenk, N.; Reckling, L. Towards a Linked Open Data Edition of Sumerian Corpora. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018; European Language Resources Association (ELRA): Miyazaki, Japan, 2018.
3. Buil Aranda, C.; Corby, O.; Das, S.; Feigenbaum, L.; Gearon, P.; Glimm, B.; Harris, S.; Hawke, S.; Herman, I.; Humfrey, N.; et al. SPARQL 1.1 Overview. 2013. Available online: https://www.w3.org/TR/sparql11-overview/ (accessed on 5 June 2016).
4. Alivernini, S.; D'Agostino, F.; Romano, M.; Severini, L. Ur_Namma, an OWL Ontology of a Sumerian Grammar. 2006. Available online: http://www.epistematica.com/2012/05/ur_namma-an-owl-ontology-of-a-sumerian-grammar/ (accessed on 5 June 2016).
5. Jaworski, W. Ontology-Based Knowledge Discovery from Documents in Natural Language. Ph.D. Thesis, Uniwersytet Warszawski, Warszawa, Poland, 2008.
6. Nurmikko-Fuller, T. Telling Ancient tales to Modern Machines: Ontological Representation of Sumerian Literary Narratives. Ph.D. Thesis, University of Southampton, Southampton, UK, 2015.
7. Pagé-Perron, É.; Sukhareva, M.; Khait, I.; Chiarcos, C. Machine Translation and Automated Analysis of the Sumerian Language. In *LaTeCH-CLfL Workshop, Association for Computational Linguistics (ACL) Anthology*; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 10–16. doi:10.18653/v1/W17-2202.
8. Cimiano, P.; McCrae, J.; Buitelaar, P. Lexicon Model for Ontologies. Available online: https://www.w3.org/2016/05/ontolex/ (accessed on 5 June 2016).
9. Crofts, N.; Doerr, M.; Gill, T.; Stead, S.; Stiff, M. Definition of the CIDOC Conceptual Reference Model; Version 5.0.4. 2011. Available online: http://old.cidoc-crm.org/docs/cidoc_crm_version_5.0.4.pdf (accessed on 5 June 2016).
10. de Melo, G. Lexvo.org: Language-Related Information for the Linguistic Linked Data Cloud. *Semant. Web J.* **2015**, *6*, 393–400. [CrossRef]
11. Elliott, T.; Gillies, S. Pleiades: The un-GIS for ancient geography. *J. Geogr. Inf. Sci.* **2008**, *22*, 1091–1108.
12. Chiarcos, C.; Sukhareva, M. OLiA—Ontologies of Linguistic Annotation. *Semant. Web* **2015**, *6*, 379–386. [CrossRef]
13. Goetze, A. *Cuneiform Texts from Various Collections*; Yale Oriental Series, Babylonian Texts; Yale University Press: New Haven, CT, USA, 2009.
14. Black, J.A.; Cunningham, G.; Ebeling, G.; Flückiger-Hawker, J.; Robson, E.; Taylor, J.; Zólyomi, G. The Electronic Text Corpus of Sumerian Literature. 1998–2006. Available online: http://etcsl.orinst.ox.ac.uk (accessed on 22 February 2015).
15. Molina, M. Syntactic annotation for a Hittite corpus: Problems and principles. In Proceedings of the Workshop on Computational Linguistics and Language Science (CLLS 2016), Moscow, Russia, 26 April 2016.
16. Smith, E. Query-Based Annotation and the Sumerian Verbal Prefixes. Ph.D. Thesis, University of Toronto, Toronto, ON, Canada, 2010.
17. Nivre, J.; Agić, Ž.; Ahrenberg, L.; Aranzabe, M.J.; Asahara, M.; Atutxa, A.; Ballesteros, M.; Bauer, J.; Bengoetxea, K.; Berzak, Y.; et al. Universal Dependencies 1.4. 2016. Available online: http://hdl.handle.net/11234/1-1827 (accessed on 5 June 2016).
18. Bamman, D.; Crane, G.R. The Ancient Greek and Latin Dependency Treebanks. In *Language Technology for Cultural Heritage*; Springer: New York, NY, USA, 2011; pp. 79–98.
19. Zeldes, A.; Schroeder, C.T. An NLP Pipeline for Coptic. In Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, Berlin, Germany, 11 August 2016; pp. 146–155. [CrossRef]
20. Chiarcos, C.; Fäth, C. CoNLL-RDF: Linked corpora done in an NLP-friendly way. In *International Conference on Language, Data and Knowledge*; Springer: New York, NY, USA, 2017; pp. 74–88.

21. Chiarcos, C.; Schenk, N. The ACoLi CoNLL Libraries: Beyond tab-separated values. In Proceedings of the 11th Language Resources and Evaluation Conference (LREC-2018), Miyazaki, Japan, 7–12 May 2018; European Language Resources Association (ELRA): Miyazaki, Japan, 2018.
22. Owen, D.I. *Garshana Studies*; CDL Press: Bethesda, MD, USA, 2011.
23. Das, S.; Sundara, S.; Cyganiak, R. R2RML: RDB to RDF Mapping Language; Technical Report; 2012. Available online: https://www.w3.org/TR/r2rml/ (accessed on 5 June 2016).
24. Tinney, S. Sumerian Lemmatization Primer; Technical Report; 2017. Available online: http://oracc.museum.upenn.edu/doc/help/languages/sumerian/sumerianprimer/index.html (accessed on 5 June 2016).
25. Tablan, V.; Peters, W.; Maynard, D.; Cunningham, H.; Bontcheva, K. Creating tools for morphological analysis of Sumerian. In Proceedings of the 5th Language Resources and Evaluation Conference (LREC-2006), Genoa, Italy, 22–28 May 2006.
26. Jaworski, W. Contents modelling of neo-Sumerian Ur III economic text corpus. In Proceedings of the 22nd International Conference on Computational Linguistics—Volume 1, Manchester, UK, 18–22 August 2008; Association for Computational Linguistics: Stroudsburg, PA, USA, 2008; pp. 369–376.
27. Tinney, S. Annotation of Sumerian Syntax; 2017. Available online: http://oracc.museum.upenn.edu/doc/help/languages/sumerian/syntax/index.html (accessed on 14 September 2018).
28. Nivre, J.; Hall, J.; Nilsson, J.; Chanev, A.; Eryigit, G.; Kübler, S.; Marinov, S.; Marsi, E. MaltParser: A language -independent system for data-driven dependency parsing. *Nat. Lang. Eng.* **2007**, *13*, 95–135. [CrossRef]
29. Jagersma, A.H. A Descriptive Grammar of Sumerian. Ph.D. Thesis, Faculty of the Humanities, Leiden University, Leiden, The Netherlands, 2010.
30. Hayes, J.L. *A Manual of Sumerian Grammar and Texts. Second Revised and Expanded Edition*; Number 5 in Artanes, Undena Publications: Malibu, CA, USA, 2000.
31. Björkelund, A.; Bohnet, B.; Hafdell, L.; Nugues, P. A high-performance syntactic and semantic dependency parser. In Proceedings of the Coling 2010: 23rd International Conference on Computational Linguistics, Beijing, China, 23–27 August 2010; pp. 33–36.
32. Koehn, P. *Statistical Machine Translation*; Cambridge University Press: Cambridge, UK, 2009.
33. Klein, G.; Kim, Y.; Deng, Y.; Senellart, J.; Rush, A. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In Proceedings of the ACL 2017, System Demonstrations, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 67–72.