

UD-ETCSUX: Toward a Better Understanding of Sumerian Syntax

Kenan Jiang

Independent Researcher
kenanj11@berkeley.edu

Adam Anderson

University of California, Berkeley
adam.anderson@factgrid.eu

Abstract

Beginning with the discovery of the cuneiform writing system in 1835, there have been numerous grammars published illustrating the complexities of the Sumerian language. However, the one thing the published grammars have in common is their omission of dependency rules for syntax in Sumerian linguistics. For this reason we are working toward a better understanding of Sumerian syntax, by means of dependency-grammar in the Universal Dependencies (UD) framework. Therefore, in this study we articulate the methods and engineering techniques that can address the hardships in annotating dependency relationships in the Sumerian texts in transliteration from the *Electronic Text Corpora of Sumerian* (ETCSUX). Our code can be found at <https://github.com/ancient-world-citation-analysis/UD-ETCSUX>.

1 Introduction

The Sumerian language has been studied academically by philologists since Henry Rawlinson's discovery of the cuneiform writing system in 1835 (Cathcart, 2011). Since then, there have been numerous grammars published illustrating the complexities of the Sumerian language, including: epigraphy, orthography, phonology, morphology, and semantics. While not all of these grammars are in agreement, the one thing they have in common is their general lack of rules for dependency-grammar. This is because Sumerian is a highly inflected language with post-position particles for cases, numbers, and persons, and an agglutinative verbal system that reflects these same features for a given clause or sentence in the verbal chain, thereby reducing the need for complex syntax rules. For this reason, we are working toward a better understanding of Sumerian syntax, by means of dependency-grammar in the Universal Dependencies (UD) framework (Nivre et al., 2017), in order

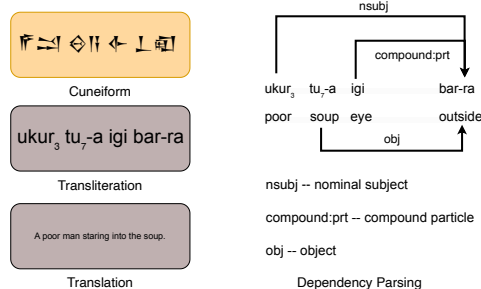


Figure 1: A dependency parsing example of Sumerian transliteration.

to model the many different dependencies of a polysynthetic sentence and illustrate the results using UD treebanks. This paper is meant to serve as the first step in motivating the much-needed collaboration of computational linguists and Sumerologists in the development of open-source tools for the Sumerian language, and the cuneiform writing system. Our contribution is summarized as follows:

- We release the first dependency dataset for Sumerian UD-ETCSUX.
- We present a dependency parser for Sumerian texts in transliteration.
- We identify the two major challenges in Sumerian syntax studies.

2 Related Work

2.1 Sumerian Language

Sumerian has a problematic past from the moment of its decipherment, in that a few modern philologists were motivated to situate Sumerian as the progenitor of their own linguistic family trees (Cooper, 1991). Further compounding the historical linguistic study of Sumerian is the fact that the natural language died out near the end of the third millennium B.C. (Michalowski, 2000). From that point onward (i.e. from 2000 B.C. to

539 B.C.), the Sumerian language was studied and taught in scribal schools throughout Mesopotamia and was preserved much like Latin in Medieval Europe (Kraus, 2020). Our modern understanding of Sumerian relies on the Sumerian-Akkadian reference works (e.g. lexicons, syllabaries, commentaries, and translations) made by many generations of Mesopotamian scribes who continued to elaborate on Sumerian’s complex morpho-graphemic orthography, and who integrated the frozen-form Sumerian logograms into the vocabulary of a considerable number of contemporary languages, like Akkadian, Elamite, and Hittite (Seri, 2010).

2.2 Computational Linguistics Tools

Due to the complex nature of Sumerian syntax, current computational tools for Sumerian transliteration have primarily focused on lemmatization and part-of-speech (POS) tagging. Of note are the recent contributions from specialists in Helsinki (Sahala and Lindén, 2023), who introduced BabyLemmatizer, a neural framework that applies machine translation methodologies to train annotators for POS and lemmatization. This approach conceptualizes tagging challenges as translation tasks, utilizing a sequence-based transformer model to generate tags. However, there remains a gap in the research as no existing studies have explored effective computational techniques for dependency labeling in Sumerian.

2.3 Cuneiform Corpora

Developing high-quality corpora is pivotal for advancing Sumerian language tools. Fortunately, the online (aka ‘electronic’) publications of Sumerian texts got an early start in ETCSL, the electronic text corpus of Sumerian literature (Black et al., 1998–2006), and ETCSRI, the Sumerian royal inscriptions (Zólyomi, Gábor - Tanos, Bálint - Sövegjártó, Szilvia, 2008). The first study to develop UD labels for Sumerian was the MTAAC project (Pagé-Perron et al., 2017), with a goal to translate 100,000 Sumerian texts from the Ur III period (2100-2000 B.C.). In preparation for this goal, they designed dependency sets tailored for Sumerian in transliteration, of which there are currently 370 published examples in the CDLI (CDLI contributors, 2024). The culmination of these efforts underscores a collaboration between NLP experts and Assyriologists to build a Sumerian text retrieval system, enhancing accessibility through a specialized NLP pipeline and linguistically linked open data. Although the

Sumerian	Lemma	Dependency	Head
ur-gir ₁₅ -gin ₇	urgir	amod	5
ki	ki	compound:prt	3
za-za	zaza	aux	5
hul	hulu	compound:prt	5
a-ab-gig	gig	root+nsubj	5

Table 1: Examples from UD-ETCSUX dataset: "Like a dog, he hates to grovel."

focus has been on morphological annotation in the MTAAC workflow, comprehensive steps for dependency parsing remain less detailed, with only a handful of examples in CONLLU format documented by (Chiarcos et al., 2018).

2.4 Syntax Parsing

Dependency parsing is a syntactic parsing technique that represents the structure of a sentence in terms of binary relations between words, capturing the head-dependent relationship (Jurafsky and Martin, 2009). This parsing method facilitates the understanding of syntactic and semantic structures, aiding various applications such as machine translation and information extraction. Among contemporary models, spaCy’s dependency parser (Honnibal et al., 2020) stands out due to its efficiency and accuracy. spaCy utilizes state-of-the-art neural network architectures and pre-trained word embeddings to capture complex linguistic patterns, making it highly effective for parsing diverse and morphologically rich languages. Its robust performance and ease of integration have made spaCy a popular choice for researchers and developers working on a wide range of NLP tasks.

3 UD-ETCSUX Dataset

In this section, we delineate the methodology of our dependency dataset, UD-ETCSUX. Initially, we extracted Sumerian transliterations along with their English equivalents from the ETCSRI (Zólyomi, Gábor - Tanos, Bálint - Sövegjártó, Szilvia, 2008) and ETCSL (Black et al., 1998–2006) datasets. Subsequently, we utilized the spaCy framework (Honnibal et al., 2020) to perform dependency tagging on the English translations. Using the dependency tags derived from the Universal Dependencies (UD) and the English translations, we manually transferred the UD labels from the English texts to the corresponding Sumerian transliterations, guided by the lemmatization and English

gloss words provided in ETCSRI and ETCSL. This methodology enables annotators with limited expertise in Sumerian to initiate the annotation of basic structures in the language. With UD labels directly attached to Sumerian transliterations, this process facilitates the later migration of the labeled data into a UD-compatible format. We present an example from UD-ETCSUX in Table 1. Currently, we have curated dependency trees for 133 Sumerian transliterations, containing a total of 573 labeled data. We also plan to incorporate feedback from the scholarly community and will modify our labels based on their input in future versions. To address the challenge of a limited number of training examples, we employed data augmentation techniques. We selected 60 labeled Sumerian transliterations and used a trained embedding model to find the most semantically similar words in the corpus, replacing the original transliterations with new words of similar meaning. This process generated 60 new transliterations, with an example presented in Table 2.

4 Sumerian Dependency Parser

In this section, we present the complete workflow that forms our dependency parser.

4.1 Compound Verbs

Compound verbs represent a distinctive but challenging aspect of Sumerian transliteration. Due to their extensive variety and frequent occurrences, coupled with morphological variations, accurately identifying compound verbs can be time-consuming for annotators (see Table 1). To streamline the annotation process and improve the accuracy of dependency labeling, we have developed a compound verb detector. This tool contains 674 general compound verbs, and a total of 1055 variations derived from the general compound verbs in its dictionary and is designed to automatically detect potential compound verbs and provide their corresponding English meanings given a Sumerian transliteration. We present some examples of our compound verb detector in the Appendix.

4.2 Word Embeddings

To enhance the performance of our dependency parser, we trained two types of word embedding models using the entire ETCSRI and ETCSL corpora, which contain 277,247 lines of Sumerian text in transliteration. We utilized FastText (Bojanowski et al., 2016) and Pointwise Mutual Infor-

Sumerian	Lemma	Dependency	Head
nita	nita	nsubj	4
zig	zig	acl	1
mumun	mumun	obj	4
al	al	root	4

Table 2: Example of augmented data: "A male aroused eats salt."

mation (PMI) (Church and Hanks, 1990) embedding techniques for this purpose. For exploration, we calculated one set of embeddings on transliterations and another on lemmas, setting the embedding dimension to 512. Four embedding models were incorporated into our dependency parsing training regimen. The comparative effectiveness of these models is thoroughly evaluated in the Experiments section.

4.3 Implementation Details

We built our dependency parser using the training framework of spaCy (Honnibal et al., 2020), tokenizing Sumerian transliterations by spaces. Given our dataset of 125 sentences, we performed a 10-fold cross-validation to evaluate the parser’s effectiveness. Utilizing a custom embedding layer, the parser was trained for 20 epochs per fold with a minibatch size of 12. To prevent overfitting and enhance robustness against minor labeling errors, we applied a dropout rate of 0.8.

5 Experiments

We evaluated both PMI and FastText embedding methods trained on lemmas and transliterations, respectively. For each 10-fold cross-validation, we report the average Unlabeled Attachment Score (UAS) (Ratnaparkhi, 1996) and Labeled Attachment Score (LAS) (Buchholz and Marsi, 2006) across the folds.

The UAS measures the parser’s ability to identify the sentence structure, focusing on the correctness of the head assignments. In contrast, the LAS evaluates the parser’s performance on both dependency tags and sentence structure, assessing both head assignments and the correct labeling of dependency relations. Both scores are reported as percentage accuracy, ranging from 0 to 100, with 100 being complete correctness. The detailed scores are presented in Table 3.

From Table 3, we observe that our parser performs slightly better with FastText embed-

	Lemma		Transliteration	
	PMI	FastText	PMI	FastText
UAS	50.69	51.54	50.47	51.27
LAS	13.19	13.23	12.84	13.29

Table 3: Results for 10-fold validation.

dings compared to PMI embeddings, regardless of whether the embeddings were trained on lemmas or transliterations. Specifically, FastText embeddings trained on lemmas yield the highest UAS at 51.54, indicating a more effective approach in capturing syntactic structure. The LAS, which evaluates both dependency tags and sentence structure, shows a similar trend, with FastText generally outperforming PMI, though the differences are less pronounced. However, both UAS and LAS scores are relatively low across all methods, with the highest UAS at 51.54 and the highest LAS at 13.29, indicating significant room for improvement. These low scores reflect the challenges of parsing Sumerian text, likely due to the limited dataset size and the language’s complexity. To improve accuracy with limited resources, we trained our parser with 60 additional augmented data. The same evaluation is presented in Table 4.

	Lemma		Transliteration	
	PMI	FastText	PMI	FastText
UAS	51.96	51.86	50.62	51.20
LAS	13.47	13.75	14.19	13.82

Table 4: Results for 10-fold validation with 60 augmented data.

Compared to Table 3, we observe consistent improvements in both UAS and LAS across all embedding settings. This highlights the promising potential of using data augmentation techniques to temporarily mitigate the negative impacts of low-resource data in enhancing parsing performance for Sumerian.

6 Qualitative Evaluation

We present two examples to illustrate the performance of our dependency parser.

Table 5 showcases a correct inference where the parser accurately identified the nominal subject (nsubj), object (obj), compound particle (comp.prt), and root. This demonstrates the parser’s ability to handle straightforward Sumerian sentences effectively. Notably, it also highlights the effectiveness

of our compound verb detector. During inference, the detector successfully identified "igi" and "bar" as a compound verb, assigning "igi" the compound particle label directly and thereby preventing potential confusion for the parser. The dependency relations and head assignments align with the expected structure, reflecting the parser’s proficiency in parsing simple syntactic constructions.

Sumerian	Truth		Predicted	
	Dep.	Head	Dep.	Head
ukur ₃	nsubj	4	nsubj	4
tu ₇ -a	obj	4	obj	4
igi	comp.prt	4	comp.prt	4
bar-ra	root	4	root	4

Table 5: Example of a correct inference: "A dog climbed up onto the roof."

Sumerian	Truth		Predicted	
	Dep.	Head	Dep.	Head
ur	nsubj	4	nsubj	4
si-im-si-im	amod	1	nsubj	4
e ₂ -e ₂ -a	obj	4	obj	4
ku ₄ -ku ₄	root	4	root	4

Table 6: Example of an incorrect inference: "A sniffing dog entering all the houses."

Table 6 presents a failed case. The parser misclassified "si-im-si-im" as a nominal subject (nsubj) instead of an adjectival modifier (amod), which affected the overall dependency structure. We believe this misclassification is due to the parser’s limited exposure to diverse sentence structures and the imbalance in the training data, making it challenging to accurately recognize and differentiate adjectival modifiers, which are less common, from nominal subjects. Such failures underscore the need to incorporate more diverse and complex sentences into UD-ETCSUX to broaden the parser’s capabilities.

7 Future Directions

We plan to incorporate additional feedback from language experts and continuously expand and enhance the quality of UD-ETCSUX, with the ultimate goal of publishing it in the Universal Dependency Treebank. Additionally, we will conduct inter-annotator agreement studies in future work to ensure the reliability and consistency of our annotations. Furthermore, we have identified two issues that require targeted solutions in future research.

7.1 Morphology Inclusion

Sumerian has a highly-inflected morphology, which in many instances encapsulates multiple parts of speech and phrasal elements into a single word, as seen in Table 1, which contains both the subject and root in three signs or one token. In order to properly identify each of the phrasal elements of a sentence, it will be necessary to annotate these sub-word particles, especially for the verbs. Fortunately, this format has been clearly articulated in recent Sumerian grammars, but it has only been applied to the ETCSRI corpus, and has not yet been extended to the rest of the electronic text corpora of Sumerian. We see this as a critical step in order to allow for an automated process of dependency parsing. As such, we plan to provide the full repertoire of Sumerian texts with annotations for sub-word particles in subsequent versions of the UD-ETCSUX dataset.

7.2 Multiple Translations

Also mentioned above is the fact that much of the vocabulary and many of the literary texts in Sumerian exhibit forms of word-play, parallelism, polysemy, and double-entendre. (Alster, 1975) A good example of this may be seen in the sentence in Table 1, which ETCSL translates: "Like a dog, he hates to grovel," but which could also be read, "he hates to grovel like a dog." The former implies the dog's hatred of groveling and the latter only likens the subject's act of groveling to a dog's. Both readings are possible because there is no separate subject in the sentence outside of the verbal chain. While including such polysemy in our model might over-complicate the process from the start, we hope to include the plurality of dependency parsings in the future to reflect the rich layers of meaning embedded in the Sumerian text.

8 Conclusion

In this work, we presented UD-ETCSUX, a concise dataset for Sumerian dependency parsing. Additionally, we introduced tools to enhance parsing accuracy, such as compound verb detection and data augmentation techniques. Our dependency parsing analysis compared various embedding methods and identified areas for future improvement. We hope our contributions will prove valuable and inspire language experts to further advance the understanding of Sumerian syntax.

9 Acknowledgements

We would like to express our gratitude to Émilie Pagé-Perron for sharing the latest work from the MTAAC project, as well as Jason Moser, Matthew Ong, Andrew Pottorf, and Manuel Molina for their extensive feedback on our initial results.

10 Limitations

While our dataset effectively supports the initial objectives of this study, its current scope is limited, restricting our ability to fully explore the diverse linguistic scenarios in Sumerian languages. Furthermore, more extensive expert validations are required to enhance our dataset's robustness. We are still in the process of receiving and incorporating feedback from Sumerian language specialists, and we are committed to expanding the dataset and deepening expert collaborations to refine the quality and applicability of our findings.

References

- Bendt Alster. 1975. [Paradoxical proverbs and satire in sumerian literature](#). *Journal of Cuneiform Studies*, 27(4):201–230.
- J.A. Black, G. Cunningham, J. Ebeling, E. Flückiger-Hawker, E. Robson, J. Taylor, and G. Zólyomi. 1998–2006. The electronic text corpus of sumerian literature. <http://etcsl.orinst.ox.ac.uk/>. Accessed: 2024-05-24.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. [Enriching word vectors with subword information](#). *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Sabine Buchholz and Erwin Marsi. 2006. [CoNLL-X shared task on multilingual dependency parsing](#). In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City. Association for Computational Linguistics.
- K. J. Cathcart. 2011. [The Earliest Contributions to the Decipherment of Sumerian and Akkadian](#). *Cuneiform Digital Library Journal*, 2011(1). [Online; accessed 2024-05-15].
- CDLI contributors. 2024. [Cuneiform digital library initiative](https://cdli.mpiwg-berlin.mpg.de/). <https://cdli.mpiwg-berlin.mpg.de/>. [Online; accessed 2024-05-24].
- Christian Chiarcos, 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).

Kenneth Ward Church and Patrick Hanks. 1990. [Word association norms, mutual information, and lexicography](#). *Computational Linguistics*, 16(1):22–29.

Jerry S. Cooper. 1991. Posing the Sumerian question: Race and scholarship in the early history of assyriology. *Aula Orientalis*, 9:47–66.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spaCy: Industrial-strength Natural Language Processing in Python](#).

Dan Jurafsky and James H. Martin. 2009. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Prentice Hall, Upper Saddle River, N.J.

Nicholas L. Kraus. 2020. *Scribal Education in the Sargonic Period*. Brill.

Piotr Michalowski. 2000. The life and death of the sumerian language in comparative perspective. *Acta Sumerologica*, 22:177–202.

Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis Tyers. 2017. [Universal Dependencies](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Valencia, Spain. Association for Computational Linguistics.

Émilie Pagé-Perron, Maria Sukhareva, Ilya Khait, and Christian Chiarcos. 2017. [Machine translation and automated analysis of the Sumerian language](#). In *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 10–16, Vancouver, Canada. Association for Computational Linguistics.

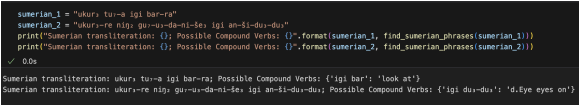
A. Ratnaparkhi. 1996. [A maximum entropy model for part-of-speech tagging](#). In *Conference on Empirical Methods in Natural Language Processing*.

Aleksi Sahala and Krister Lindén. 2023. [A neural pipeline for POS-tagging and lemmatizing cuneiform languages](#). In *Proceedings of the Ancient Language Processing Workshop*, pages 203–212, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Andrea Seri. 2010. [3. adaptation of cuneiform to write Akkadian](#). In *Visible Language: Inventions of Writing in the Ancient Middle East and Beyond*, pages 85–98, Chicago, USA. Oriental Institute Museum Publications.

Zólyomi, Gábor - Tanos, Bálint - Sövegjártó, Szilvia. 2008. [The electronic text corpus of sumerian royal inscriptions](#).

A Appendix



```
sumerian_1 = "ukur3 tu-a igi bar-ra"
sumerian_2 = "ukur3-re ni3 gu-u3-da-ni-se, igi an-si-du-du"
print("Sumerian transliteration: {}, Possible Compound Verbs: {}".format(sumerian_1, find_sumerian_phrases(sumerian_1)))
print("Sumerian transliteration: {}, Possible Compound Verbs: {}".format(sumerian_2, find_sumerian_phrases(sumerian_2)))
✓ 0.0s
Sumerian transliteration: ukur3 tu-a igi bar-ra; Possible Compound Verbs: ('igi bar': 'look at')
Sumerian transliteration: ukur3-re ni3 gu-u3-da-ni-se, igi an-si-du-du; Possible Compound Verbs: ('igi du-du': 'd.Eye eyes on')
```

Figure 2: Use case for compound verb detector.