At the Crossroad of Cuneiform and NLP: Challenges for Fine-grained Part-of-Speech Tagging

Gustav Ryberg Smidt[1], Katrien De Graef[1], Els Lefever[2]
[1]The Department of Languages and Cultures, Ghent University, Blandijnberg 2, Ghent, Belgium
[2]Language and Translation Technology Team, Ghent University, Groot-Brittanniëlaan 45, Ghent, Belgium
{GustavRyberg.Smidt, Katrien.DeGraef, Els.Lefever}@UGent.be

Abstract

The study of ancient Middle Eastern cultures is dominated by the vast number of cuneiform texts. Multiple languages and language families were expressed in cuneiform. The most dominant language written in cuneiform is the Semitic Akkadian, which is the focus of this paper. We are specifically focusing on letters written in the dialect used in modern-day Baghdad and south towards the Persian Gulf during the Old Babylonian period (c. 2000-1600 B.C.E.). The Akkadian language was rediscovered in the 19th century and is now being scrutinised by Natural Language Processing (NLP) methods. However, existing Akkadian text publications are not always suitable for digital editions. We therefore risk applying NLP methods onto renderings of Akkadian unfit for the purpose. In this paper we want to investigate the input material and try to initiate a discussion about best-practices in the crossroad where NLP meets cuneiform studies. Specifically, we want to question the use of pre-trained embeddings, sentence segmentation and the type of cuneiform input used to fine-tune language models for the task of fine-grained Part-of-Speech tagging. We examine the issues by theoretical and practical approaches in a way that we hope spurs discussions that are relevant for automatic processing of other ancient languages.

Keywords: Cuneiform, Old Babylonian Akkadian, NLP, Part-of-Speech tagging

1. Introduction

Amongst ancient text corpora, the cuneiform corpus is one of the largest and in a time where computational analysis is increasingly dominating research, it must naturally follow that digitisation of cuneiform texts is essential for evolving the study of cuneiform texts and cultures. However, digitisation, and in that vein linguistic augmentation of data, of the cuneiform corpus is a complex case. We will discuss some core elements of working with digital cuneiform texts and the decisions necessary for reaching a successful outcome. To do so, we will first outline our corpus and research goals in Section 2. Section 3 describes various aspects of the cuneiform corpus, while Section 4 gives an overview of useful Natural Language Processing (NLP) methods. Following, is a description of our preliminary NLP experiments (Section 5), which is the basis for a discussion of how to approach the cuneiform corpus with NLP in mind (Section 6). The discussion is meant as a preliminary view on our corpus' functionality in connection with NLP and will hopefully result in further discussions on how to approach digital ancient texts.

2. CUNE-IIIF-ORM and NLP

The CUNE-IIIF-ORM project is a cooperation between the Royal Museums of Art and History, KU Leuven and Ghent University. It aspires to test different avenues of digital methods to implement state-of-the-art solutions for dissemination, automatic reading and computational analysis of cuneiform texts. In this paper we focus on the automatic reading and computational analysis part or NLP. We are examining the corpus of Old Babylonian (OB) Akkadian texts (c. 2000-1600 B.C.E.) by using freely available NLP Machine Learning frameworks to create robust language models. The first objective is to semi-automatically increase the number of linguistically annotated texts in the corpus and the second objective is to examine the OB Akkadian language through computational analyses of the corpus. We are currently exploring the first step and have completed a set of preliminary experiments for fine-grained Part-of-Speech tagging (see Section 5), which has highlighted both the potential and the pitfalls of our approaches. Addressing these pitfalls should allow us to improve the results of our future NLP analyses. In the proposed research, we will focus on the usage of pre-trained embeddings, sentence segmentation and Unicode cuneiform.

3. Cuneiform

3.1. The Script

Cuneiform is a logo-syllabic script with signs made up of a number of wedges impressed or engraved onto various materials, where clay tablets dominate the corpus (seeFigure 1 for an example). It was used to write multiple languages, notably are

LREC-COLING 2024, pages 1745–1755
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

the linguistic isolate Sumerian, the Semitic Akkadian and the Indo-European Hittite. The earliest cuneiform writing is first attested around 4500 B.C.E. in modern-day southern Iraq to write Sumerian and it was last used in the 1st cent. C.E. The current number of cuneiform texts is likely past 500.000 ((Streck, 2010)) and many historic sites are still untouched or only partially excavated.

Figure 1: Front side of an Old Babylonian cuneiform clay tablet written in Akkadian. It is kept in the Royal Museums of Art and History in Brussels (museum number O.222).

3.2. Cuneiform for Akkadian

The cuneiform used to write Akkadian has three different types of signs: syllables, word signs or logograms, and determinatives or classifiers. The syllabic signs are used to write the words by approximating their pronunciation in writing and are for Akkadian transliterated with lowercase italic letters. It is common to see spelling variations of words with syllables, since not all double consonants and vowel lengths are consistently written. Word signs can have pre- or suffixes, but they are typically written in simple form without inflections. In Akkadian, word signs are mostly inherited from Sumerian, why they are also called Sumerograms and for Akkadian texts they are transliterated with uppercase letters. The determinatives disambiguate the often semantic ambiguity of word signs and words in general. They are not pronounced in speech but only used in writing where they are usually transliterated as superscript. Signs are not limited to only one of these categories, **dingir** (U+1202D) can be read as the syllable *an*, the Sumerogram DINGIR and the classifier ^{d}. To complicate the matter further, cuneiform also exhibits polyphony and homophony. Polyphony means that one cuneiform sign can represent different phonologically unrelated syllables e.g., the sign **ur** (U+12328) can be read *ur*, *lik* and *tas*. Where homophony means that one syllable can be represented by multiple signs e.g., **bi** (U+12049) can be *bi*, but so can **ne** (U+12248) and **pi** (U+1227F), amongst others. We distinguish between the same phonetic values written with different signs by using subscript numerals, they would in the mentioned case be *bi*, bi_2 and bi_3, respectively. Which sign is used for a given sound can carry meaning, such as the rather consistent use of the sign $\mathbf{u_3}$ (U+12147) for the conjunction *u*, instead of **u** (U+1230B) or $\mathbf{u_2}$ (U+12311). But there is no system that encompass the whole set of signs and sounds. Furthermore, even the use of $\mathbf{u_3}$ for the conjunction *u* is inconsistent across Akkadian and not all sound values of a given sign is applicable across different time periods of Akkadian.

3.3. Akkadian Texts

The Akkadian language is a highly inflectional Semitic language attested from c. the 24th cent. B.C.E. to the 1st cent. C.E. It has mainly been found in modern-day Iraq and Syria. As typical for Semitic languages, Akkadian words are mainly constructed by three radicals or consonants surrounded by a pattern of consonants and vowels that determines the Part-of-Speech (PoS), semantics and inflection of a word. These radicals carry meaning in their combination such as the paradigmatic PRS, which as a verb typically means ‘to cut off’ (parāsu) and as a noun can mean ’a separated place’ (parsu). The variations from the patterns stem from phonological influences that dominate different dialects and they act predictable in many cases. Akkadian was highly influenced by the isolate language Sumerian. Akkadian inherited the cuneiform script itself, word signs and its word order from Sumerian. That is why Akkadian has a great number of Sumerian loan words and different from most Semitic languages, Akkadian predominantly wrote in the order Subject-Object-Verb (SOV) with syllabic signs. Across the many different text genres and periods Akkadian is used for, the texts vary significantly. The OB contracts exhibit a very limited vocabulary, the continuously used lexical lists have a rich vocabulary but typically no syntactical structure and the Neo-Assyrian (c. 911-612) royal reliefs has an excessive use of word signs.

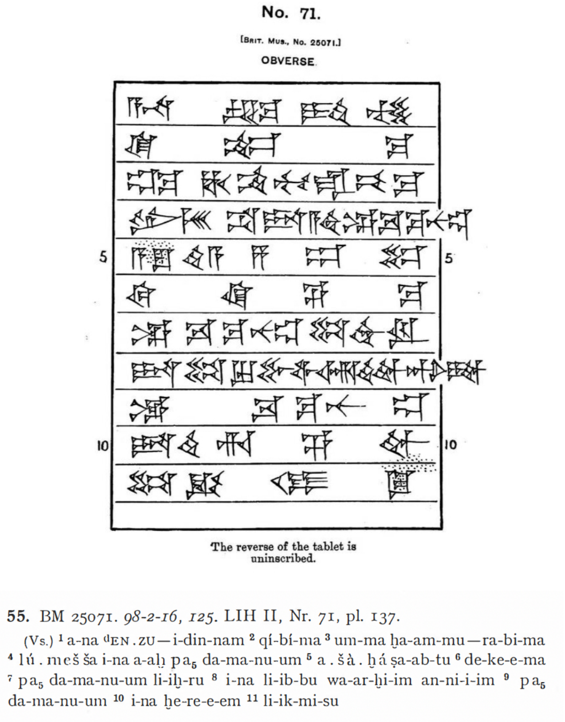
For the presented experiments, we work on OB Akkadian letters. We chose to work with these texts because they contain many different topics, the occasional direct quote, they seem to express a language closer to the vernacular, and most often we also know who conveyed the messages and who was meant to receive them. It is also our opinion that Akkadian had limited external influence from other languages during the OB period.[1] Nonetheless, it has to be taken into account that the OB corpus covers around 400 years of a language's history in a very volatile political reality.

3.4. Rediscovery of Akkadian

Since the last known cuneiform text from the 1st cent. C.E. until the 19th cent. C.E. the knowledge of how to read cuneiform and Akkadian was lost. When studying OB Akkadian in the 21st cent. we approximate understanding of the texts within a modern and largely Western conceptual field.[2] Most texts have been published as line drawings (Figure 2: above) and/or transliterations (Figure 2: below), which are alphabetic and phonetic approximations of the language. In some cases publications also contain transcriptions of the language where the information of the written reality of the texts is removed in favour of including more explicit grammatical information (see Figure 3 for an example of the differences in notation). For many years these formats were the most suitable for the context of their publications.

3.5. Digital Cuneiform Editions

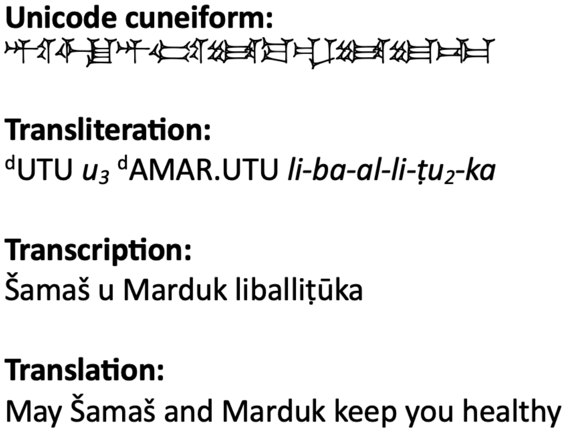
In digitised corpora we see that the ASCII transliteration format or ATF dominates accessible text publications with and without annotations. It is used in different versions by the Cuneiform Digital Library Initiative (CDLI), The Open Richly Annotated Cuneiform Corpus (ORACC) and the electronic Babylonian Library, just to mention a few. Besides ATF, also the CoNLL-U format is growing in popularity for cuneiform studies e.g., BabyLemmatizer 2.0 (Sahala and Lindén, 2023), CDLI-CoNLL (Chiarcos et al., 2018), Luukko et al. 2020 and Ong and Gordin 2024, but it is still not as widely used as ATF. Digital publications are usually based on transliterations as the basic textual data. Both ATF and CoNLL-U provide ample possibility to annotate a word based on its place in a text,[3] its morphological features and its lemma. It is typical to provide a citation form, a translation, PoS-tag and a transcription, like often seen in ORACC. The parsing of the word level information might differ in digital publications, but the template from ORACC is considered a representable example of current digital editions of Akkadian texts. This template is also well aligned with physical publications that at most provide transcriptions, but more commonly only have sporadic notes for difficult words when going from transliteration to translation e.g., the Altbabylonische Briefe series (Kraus and Veenhof, 1964–2005). CoNLL-U can support a similar structure as ORACC ATF,



55. BM 25071. *98-2-16, 125.* LIH II, Nr. 71, pl. 137.

(Vs.) [1] a-na ^{d}EN.ZU—i-din-nam [2] qí-bí-ma [3] um-ma ḫa-am-mu—ra-bi-ma [4] lú.meš ša i-na a-aḫ pa$_5$ da-ma-nu-um [5] a.šà.ḫá ṣa-ab-tu [6] de-ke-e-ma [7] pa$_5$ da-ma-nu-um li-iḫ-ru [8] i-na li-ib-bu wa-ar-ḫi-im an-ni-i-im [9] pa$_5$ da-ma-nu-um [10] i-na ḫe-re-e-em [11] li-ik-mi-su

Figure 2: Above: typical line drawing of a tablet (King, 1900 no. 71). Below: corresponding transliteration to line drawing above, notice the 66 years between publications (Frankena, 1966 p. 32).



Unicode cuneiform:
𒀭𒌓𒅇𒀭𒀫𒌓𒇷𒁀𒀠𒇷𒌅𒅗

Transliteration:
^{d}UTU *u$_3$* ^{d}AMAR.UTU *li-ba-al-li-ṭu$_2$-ka*

Transcription:
Šamaš u Marduk liballiṭūka

Translation:
May Šamaš and Marduk keep you healthy

Figure 3: Typical example from the preamble of an OB Akkadian letter displaying the difference between Unicode cuneiform, the phonetic interpretation of the signs, an analysis of the composed words and a translation of the passage

[1] For a short overview of OB Akkadian language contact see Streck, 2022 (pp. 11–15).

[2] We will not consider the impact of an Eurocentric or Occidental lens, but acknowledge that such efforts could likely be beneficial in the context of our discussion.

[3] We do not go so far as to say syntax. Even though ATF should provide the functionality for it, we are yet to see it implemented.

but it differs in grammatical annotation. Instead of including a transcription, CoNLL-U has each morphological feature tagged. This is more in line with our project, as we wish to provide our textual editions with grammatical feature tags. By doing so, one avoids ambiguities that can be found in transcriptions and it makes the content more accessible for people without inside knowledge of Akkadian. Currently, there is no dominant CoNLL-U standard for Akkadian.

4. Cuneiform NLP

4.1. Enlarging the Corpus

With the modest number of freely available digitised editions of OB cuneiform, it seems obvious to implement NLP for both automatic annotation and textual analysis. Few projects are currently trying to utilise NLP on cuneiform texts; for a more comprehensive account see Sahala, 2021 (pp. 31–72).

A first challenge is the moderate size of the digital corpus that can be used to train and evaluate NLP approaches for Akkadian. There are different approaches to increase the number of available digital editions, either by (1) digitising physical publications or (2) creating them from new. (1) The first is typically done by manually copying the texts from physical publications to a digital format. In best case scenario this is done by specialists that understand the specifics of Assyriological notation, which varies across scholarly traditions, types of texts and research history. Alternatively, line drawings or transliterations can be automatically read from scans of physical publications. The Cuneiform Recognition (CuRe) utilises a machine learning model trained to recognise line drawings (Gordin and Romach, 2022) and Cuneiform Recognition Documents (CuReD) uses an OCR model to read transliterations. Both are developed within the Digital Pasts Lab (DigPasts-Lab). (2) The process of automatically extracting new texts relies on images of cuneiform tablets. Currently, there is no general OCR model for cuneiform tablets. However, according to Sahala, 2021 (p. 42), it is not unlikely that there will soon be domain-specific models (see for instance Gordin et al., 2020) that can (semi-)automatically recognise cuneiform signs on certain types of images.

4.2. Linguistically Augmenting the Data

In parallel with increasing the size of the corpus, it is necessary to augment the digital editions with linguistic annotations. ORACC has a glossary-based automatic lemmatizer called L2 that can draw from any glossary within the ORACC framework. The lemmatizer is quick and increasingly effective as glossaries grow, but it does not account for spelling variations and glossary mistakes can be difficult to root out. Despite the effectiveness of L2, especially for specialists in the subject, it cannot recognise unseen forms, which is important for a highly inflectional language like Akkadian and a complex script like cuneiform. We will mention two attempts to mitigate this. The first is a finite-state based morphological model for the Babylonian dialect of Akkadian developed at the University of Helsinki (Sahala et al., 2020). It relies on the regularity of the inflections of Akkadian words. By knowing the root consonants of a word and their use of vowels, it is possible to reconstruct all possible inflections expressed in transcriptions. For Akkadian words this method works fine and the rules of the language could potentially be changed quickly between texts of different genres, periods and places. The evaluation of the model reached a recall of up to 93,65 % on tokens in Standard Babylonian texts. We cannot ascertain how well this model works for OB Akkadian letters as they are largely missing from the evaluation data taken from ORACC. There are two main issues with this model, one is that the coverage is limited to Akkadian words or loanwords that have been ascribed an Akkadian equivalent, the other is the issue of disambiguation of identical forms. The former issue is dominated by the nature of the cuneiform script where word signs from Sumerian are included in Akkadian mostly without any indicator of morphological reality. Furthermore, the word signs do not have a fixed Akkadian translation across dialects. The latter issue of transcription disambiguation was not solvable with the current model as the data necessary to weigh the final-state model was not available. More recently a different Akkadian lemmatizer was developed, also by the University of Helsinki. It is called BabyLemmatizer and it uses the Open Neural Machine Translation Toolkit (OpenNMT, see Klein et al., 2017) in order to predict the PoS and lemma of a word given the transliteration formatted in CoNLL-U+ (Sahala and Lindén, 2023). The implementation of OpenNMT with post-correction rules scores an accuracy of c. 94 % for the combined PoS and lemma predictions of texts written in the 1st mill. Babylonian dialect of Akkadian (Sahala and Lindén, 2023 p. 209). However, where the PoS-tag and lemma might be helpful for downstream predictions, the model does not offer morphological analyses. For many of the questions Assyriologists typically ask, this would be a minimum.

As will be mentioned below (Section 6.2), sentence segmentation or chunking is a difficult task with the current data structure of the digital editions. To solve this issue, Luukko et al. (2020) have developed a treebank that could potentially be used to do automatic syntactical annotation. This Akkadian treebank is built on Neo-Assyrian royal inscriptions for which it works well. However, the inscriptions are written more than 1000 years later than the OB Akkadian letters. They are meant to convey a political message and the material reality of the stone slabs they are engraved into results in very long lines of text. Such factors could have played a role in how the scribes conceived coordination between sentences, which would make the sentence coordination very different from those written in OB Akkadian letters.

5. Part-of-Speech Tagging Experiments

Our aim is to provide a corpus of OB Akkadian letters annotated with PoS-tags, citation forms and a morphological analysis. In order to develop a Part-of-Speech and morpheme tagger for Akkadian, we have first created a fully annotated training corpus of 121 letters from the city of Sippar. By limiting ourselves to Sippar, which has a large corpus of extant letters, we hope to have good immediate results that can make the process of semi-automatic annotation of all Sippar letters quicker. The specific texts were chosen because they are fairly well preserved and thereby provide more content per text. Three specialists in the field of Old Babylonian Akkadian annotated all the texts. Firstly, they annotated each word with a PoS-tag, citation form and transcriptions in the ORACC framework. Annotating in the ORACC framework reduces mistakes for already seen words by doing dictionary look-up in the annotation project glossary. Secondly, the transcriptions were analysed with regular expressions in Python to get the morphemes and those that could not be disambiguated were manually analysed by the specialists. The dataset used for the presented experiments is made publicly available, and can be used for replication experiments[4].

In order to perform preliminary tests with various machine-learning approaches, we used the FLAIR toolkit transformer-based architecture for sequence tagging. We considered it a good entry point as it is easy to use and provides a number of readily available pre-trained embeddings. Bansal et al. (2021) also experimented with the FLAIR sequence tagger to see how to best improve PoS tagging of Sumerian, but they reported poor results (p. 49). They used monolingual Sumerian data to pre-train, whereas we used stacked embeddings with forward and backward embeddings pre-trained on the following languages: **multilingual** (343 languages, see Agić and Vulić, 2019)[5], **Arabic**, **Spanish** and **Japanese**[6]. We chose to test the multilingual pre-trained embeddings because we hypothesised it might have benefits as OB Akkadian shows traits of the agglutinative language Sumerian, the Semitic language family and the logo-syllabic cuneiform script. These traits are not shared by any single pre-trained embedding we could access, but they can all be found in the multilingual model. Arabic was included because of its affinity with Akkadian as they are both Semitic languages, Japanese was included because it is logo-syllabic like the cuneiform script and Spanish was included as a control language because it does not have any major similarity with Akkadian.

To begin with, we trained a sequence tagger that predicted PoS-tags and in some cases a limited number of morphemes. This gave us a fairly stable baseline to evaluate different approaches on our data. Because of the modest size of our corpus, we opted for a k-fold cross-validation setup, where the text segments are randomly divided into 5 equal parts (*folds*), and training and testing is performed 5 times on different partitions of 80% (4 folds) and 20% (1 fold) of the data, respectively.

5.1. Pre-trained Models of Different Languages

Akkadian is a highly inflectional logo-syllabic language and digitally a low-resourced language. In future research, we aim to build our own language model for Akkadian, and increase our corpus with Akkadian from other periods than OB, which could significantly increase the amount of resources. Previous research for ancient languages has indeed shown that texts from different periods of a language's history can be useful in augmenting similar data. This was, for instance, the case for Ancient and Byzantine Greek, where adding modern Greek texts considerably improved the language model (Swaelens et al., 2023). Potentially, this can be extended to languages of the same family, as related work showed that typologically similar languages can provide useful data to improve the language model as well (Singh et al., 2023; de Vries et al., 2022). In this research, we wanted to test if there were indications that this would be a viable path to follow for Akkadian as well. To investigate this idea, we performed experiments to predict the following PoS-tags and select number of morphemes:

- **Verbs:** Stem, Tense,[7] Person, Genus and Number
- **Nouns:** Genus, Number, Case and State

[4] https://github.com/assyrugent/LREC-Coling2024.

[5] The dataset is currently not available for download.

[6] The three latter languages are only described on GitHub, see Arabic, Spanish and Japanese.

[7] The definition of *tense* here covers tense, aspect and mood (TAM) in Akkadian. Choosing the term *tense* is for simplicity.

- **Independent and Possessive Pronouns:** Person, Genus, Number and Case
- **Adjectives and Demonstrative Pronouns:** Genus, Number and Case
- **Interrogative and Reflexive Pronouns:** Case

The input data consists of transliterations separated into sentences. Sentences are defined by ending in verbs excluding those with the suffixes *-u* and *-ma*. We chose to also include a limited number of morphemes, because the simple test of only predicting PoS-tags can potentially draw information other than the patterns of consonants and vowels. Whereas, the more complex task of also predicting a limited number of morphemes requires the ability to distinguish between smaller variations in the consonant and vowel structures. As illustrated by Table 1, fine-tuning the Semitic language model for Arabic with our training data for Akkadian PoS-tagging performed the best, obtaining an average accuracy of 76 % over 5 runs. Second best was Spanish (average accuracy of 74 %), whereas Japanese performed just slightly better than the multilingual pre-trained embeddings. These results are supported by the Macro and Weighted average F1-results which are listed in the Appendix (Table 4). It is not surprising that Arabic performed the best considering that Semitic languages share a structure very different from what we see in Indo-European languages such as Spanish. We could likely expect a larger advantage for Arabic if the input was given as transcription instead of transliterations, because that would remove the influence of the cuneiform script on Akkadian. Choosing an Arabic pre-trained model with more focus on historic sources would possibly improve the results further.

Embeddings	Accuracy
Multilingual	71,0%
Arabic	**76,2%**
Spanish	74,1%
Japanese	72,6%

Table 1: The average accuracy over five folds from the results of predicting PoS and morphological tags with transliterated text as input and verb separated text (excluding the suffixes *-u* and *-ma*). The first column mentions the pre-trained FLAIR embeddings used for fine-tuning the model on.

5.2. Line Separations

The data we are using to fine-tune should be divided into sentences for the model to deliver the best results, as context is very important for a sequence-tagging task such as PoS-tagging. However, cuneiform Akkadian does not use punctuation and our data has not been syntactically tagged. This lead us to consider the best way to segment our data into sentences. As mentioned in Section 3.3, Akkadian generally uses the word order SOV and we will therefore test that word order as our definition of a sentence. As noun cases can be inconsistent in Akkadian, we did not consider the subject for sentence beginnings, but we chose to only use the verbs to define sentence endings. We also wanted to test the effect of grouping clauses based on subordination, typically marked with the verbal suffix *-u*, and two markers of clause coordination: the verbal conjunctive suffix *-ma* and the conjunctions (CNJ) directly following a verb. We are aware that breakage complicates the matter, but we have chosen to ignore that (see Zemánek, 2007 for a discussion of the fragmentary state of cuneiform texts). These tests were compared to simple line separation i.e., based on the physical lines of a tablet (see Figure 2:above), and separation into the individual texts. All tests predicted PoS-tags based on transliterated text with the Arabic pre-trained embeddings.

As shown by Table 2, our results indicate that it is worthwhile not using lines as sentence definitions for only predicting PoS-tags. There is a 2,3 percentage points difference from the best performing verb separation (94,8 %) and line separation (92,5 %). As seen in the Appendix (Table 5), the case is slightly different for the Macro average F1. Line separation outperforms 'Verb – *u*' and 'Verb – *u* & *ma* & CNJ', but not 'Verb' and 'Verb – *u* & *ma*'. However, based on these experiments we can not draw final conclusions on what level of clause coordination should be included. All four verb based separation types were within 1,3 percentage points (93,5-94,8 %). These results will likely change a lot depending on type of object, genre and period as they can exhibit varying sentence lengths and complexity, and conjunction choices.

Separation type	Accuracy
Text	88,2%
Line	92,5%
Verb	**94,8%**
Verb – *u*	93,9%
Verb – *u* & *ma*	94,1%
Verb – *u* & *ma* & CNJ	93,5%

Table 2: The average accuracy over five folds for predicting PoS-tags with Arabic pre-trained embeddings and transliterated text as input, for different sentence splitting approaches.

5.3. Unicode Cuneiform

After Gutherz et al. (2023) recorded promising results by using Unicode cuneiform in Akkadian to English translations, we wanted to test the difference of predicting PoS-tags based on transliteration and Unicode cuneiform. From our corpus we could construct datasets based on transliteration and Unicode cuneiform. The transliterations were already given from our corpus and the Unicode cuneiform was made by comparing the transliterations with a sign list made by T. Jauhiainen published as Nuolenna.[8] We again experimented with predicting PoS-tags based on the pre-trained multilingual, Arabic, Spanish and Japanese embeddings.

It can be seen in Table 3 that the rather simple task of only predicting PoS-tags on the transliterated text, shows no clear best performer of the three monolingual pre-trained embeddings, as they are within less than 1 percentage point. However, Japanese is outperforming all other pre-trained embeddings by at least 4,5 percentage points of accuracy when predicting PoS-tags based on Unicode cuneiform. This trend is especially clear when measuring the Macro average F1 differences in the Appendix (Table 6), here Japanese outperforms the Arabic pre-trained embeddings by 8,9 percentage points. It is worth noticing that all embeddings have a considerable performance loss ranging from 19,5 percentage points (Japanese) to 30,7 percentage points (Spanish). Both Akkadian and Japanese are logo-syllabic languages, so when Unicode cuneiform is used as input data, it is no surprise that the relative performance of Japanese increases. The question arising from this test is how a mixture of Japanese and Arabic pre-trained embeddings would perform on Unicode cuneiform.

Embeddings	Translit.	Unicode	Loss
Multilingual	91,3%	61,3%	30%pt.
Arabic	**94,1**%	69,4%	24,7%pt.
Spanish	93,7%	63,0%	30,7%pt.
Japanese	93,4%	**73,9**%	19,5%pt.

Table 3: The average accuracy over five folds from the results of predicting PoS-tags on verb-separated (excluding *-u* and *-ma*) sentences with different pre-trained embeddings and input in either transliteration or Unicode cuneiform.

6. Discussion

6.1. Augmenting Training Data

If we want high performing transformer models for Akkadian, it is necessary to augment the training data. Streck estimates that there are c. 9.900.000 Akkadian words extant and out of these c. 2.560.000 would be from the OB period (Streck, 2010 p. 54). It is currently impossible to verify Streck's estimates, but even if the Akkadian corpus contains c. 10 million words, we do not know how many of the words and texts are unique.[9] As the Akkadian corpus is estimated by Streck to be approximately 20 times as large as all other ancient Semitic languages put together (Streck, 2010 p. 55), there is little data to augment from these. That is why it is important to establish the best possible candidates of larger resourced languages to use for augmentation. Based on the results of our initial testing, Semitic languages could be worthwhile using for augmentation (see Table 1) and if the intention is to work on Unicode cuneiform it seems reasonable to assume that Japanese would be a beneficial addition to the augmentation corpus (see results Table 3).

6.2. Sentence Segmentation

Currently, no treebank for OB Akkadian exists and creating one is outside the scope of this project. The main bulk of the data we intend to include lacks syntactical information. We therefore have to think alternatively to get the best sentence segmentation with what is available. Similarly to Sukhareva et al. (2017 p. 99), our approach was to define a sentence based on the word order. We also considered expressed syntactical features. The former was simple, a verb defines the sentence end. For the latter, we considered both the subordinate *-u* and conjunctive *-ma* as they are distinctly marked. They indicate a coordination of clauses and therefore, not a sentence division. The data we have available tends to have no morphological tagging, but a transcription. For transcribed verbs where the stem is known, neither suffix *-ma* or *-u* have ambiguity. The latter suffix is not expressed when the verb has a vocalic ending or the ventive (see Huehnergard, 2011 pp. 183–4). In these cases the other indicators are either a number of words that can introduce a subordinate clause e.g., *kīma* or *īnuma* (see Huehnergard, 2011 pp. 283–7), or the negation *lā*. The former group could be used to define the following verb as a subordinate, but most of them can also be used as other parts-of-speech and it is our experience that the data available do not have the

[8] We have not been able to verify the quality of the sign list.

[9] As there is still a possibility that many Akkadian texts can be found in the earth of the Middle East, we might have a sufficiently large corpus in the future.

granularity that gives us the option to automatically distinguish between the two. The use of the negation *lā* instead of the more common *ul* is determined by multiple factors between which we cannot automatically distinguish, so *lā* can currently only serve as an subordinate clause marker while performing manual annotation. Taking the approach we have, the data has uniform standards for clause coordination that are clearly marked in the texts. However, this does not account for asyndetic paratactic relations i.e., two or more non-subordinate clauses that have a semantic dependency.[10] Because we cannot currently account for that, we argue that we should also not account for coordinated paratactic relations marked with a conjunction immediately following a non-subordinate verb (in Table 2 this is 'Verb – *u* & *ma* & CNJ').

6.3. Unicode Cuneiform

As of yet, we are not aware of any discussions about why one should feed the machine-learning models with Unicode cuneiform instead of transliterations. In our opinion, we see two reasons for doing so: for developing models that can potentially use computer read cuneiform signs as input and having data closer to the original material. We will not delve into the development of OCR models here. The latter argument, that Unicode cuneiform is closer to the original material than phonological approximations, is at face value reasonable. If it would be possible to get closer to the original text, less external bias would presumably be included. The question is, however, if Unicode cuneiform actually is closer to the original texts than transliterations. Since there is no OCR model able to read cuneiform signs directly off images and transform them into Unicode solely based on the graphical input, we rely on the available data. This data can be a modern reader's interpretations of signs based on their graphical appearance, such as line drawings (Figure 2: above), or conversion of transliterations into Unicode cuneiform (Figure 3). Both types of data obviously have human interpretation, but the question is if they vary significantly. First, we need to consider how a text can be read by a human. Reiner already suggested half a century ago an algorithmic approach to reading an Akkadian text (1973). It relies on reading the cuneiform signs, recognising the possible sign interpretations, specifying word boundaries based on a simple set of phonetic rules (see Reiner, 1966 chap. 4.3) and morphosyntactical rule-based analyses. These four steps require information from a sign list, a grammar and a dictionary. Strictly following this approach would result in a reasonable lack of human interpretation, where the main issue would lie in our need to transform the texts into a format readable for Western scholars. The resources that the computer has available are also defined by our understanding of different Akkadian dialects and related languages, knowledge that the ancient scribes did not possess (Reiner 1973 p. 41n51). Furthermore, the presented method is not feasible as it either assumes that every line can be broken into words in one way only or knowledge of how a certain type of text behaves. This is where we claim that the human interpretation can hardly be taken out of consideration in any way that the material has been transformed. When identifying signs in a text, most are fairly simple to read, but not all are graphically clear, correct or disambiguated from similar signs. This means we have to rely on someone deciding of all the possible readings, which is the "correct" one in order to give the right sign ID. If the reader is making a transliteration, the same decision needs to be taken, but now the transliteration makes the choice of sign reading explicit. The transformation from transliteration to Unicode cuneiform simply reintroduces ambiguity, which also explains the performance loss as seen in Section 5.3. As long as cuneiform signs cannot be delineated solely based on their graphical appearance, the computer will always need human interpretation. We might reduce that by modern scanning methods that can make graphically unclear signs readable or disambiguate similar signs by small variations not understood by humans. But correcting a spelling or not will likely always be based on a normative reading of a text. Therefore, we cannot see that any current text format will be free of or considerably reduced in human bias. We have not yet had the possibility to test the influence of human bias on sign readings, but we believe it can be tested and we intent to do so in the future. What we can say is that, the influence of human interpretation will likely vary a lot for a text depending on genre, period, writing material and state of preservation. It is therefore important to consider the purpose for using Unicode cuneiform as a text representation.

7. Conclusion

As the use of digital and computational methods are rapidly spreading for ancient languages, it can be worthwhile having a discussion about how cuneiform and computational researchers go about the material. What might seem like a tweak of a few lines of code can have a large impact on the outcome of any analysis, both on the performance scores, but also on how the given language is modelled.

In this study, we performed a set of preliminary experiments of fine-grained Part-of-Speech tagging

[10] See Deutscher, 2000 p. 14 for this definition.

for Akkadian. To this end, we applied a transformer architecture, and fine-tuned a pre-trained language model. Our results show that the Semitic language Arabic could work well as pre-trained embeddings for Akkadian and when the data is in the form of Unicode cuneiform it seems beneficial to also include a logo-syllabic script like Japanese as pre-trained embeddings. Furthermore, when fine-tuning a language model for Akkadian, the training data should be segmented into sentences. That is possible based on PoS-tags, but when a transcription is supplied a more detailed segmentation can be performed. In future research, we will experiment with a mixture of Semitic and languages written in a logo-syllabic script as pre-trained embeddings for Akkadian. In addition, we will also build our own Akkadian language model, incorporating text from various periods and genres. Based on the outcome of the current research, we think it is also useful to investigate the impact of incorporating data from other (Semitic or logo-syllabic) languages.

It seems that conscious decisions on corpus enrichment and formatting in the process of fine-tuning language models for Akkadian, plays a big role on the output. This leads us to conclude, in a very similar vein to Sommerschield et al. (2023), that interdisciplinary teams are the best way forward (p. 26). In this way it is possible to account for the essential discussions relating to the philological, linguistic and material aspects of the texts while implementing the most suitable computational solutions to reach the research goals.

8. Acknowledgements

The project is funded by Belspo under the BRAIN-be 2.0 framework (B2/212/P2/CUNE-IIIF-ORM).

9. Bibliographical References

Ž. Agić and I. Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

R. Bansal, H. Choudhary, R. Punia, N. Schenk, J.L. Dahl, and É. Pagé-Perron. 2021. How low is too low? a computational perspective on extremely low-resource languages. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 44–59.

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

W. de Vries, M. Wieling, and M. Nissim. 2022. Make the best of cross-lingual transfer: Evidence from POS tagging with over 100 languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7676–7685, Dublin, Ireland. Association for Computational Linguistics.

G. Deutscher. 2000. *Syntactic Change in Akkadian: The Evolution of Sentential Complementation*. Oxford University Press.

R. Frankena. 1966. *Briefe aus dem British Museum (LIH und CT 2-33)*. Number 2 in Altbabylonische Briefe in Umschrift und Übersetzung. E. J. Brill.

S. Gordin, G. Gutherz, A. Elazary, A. Romach, E. Jiménez, J. Berant, and Y. Cohen. 2020. Reading akkadian cuneiform using natural language processing. *PLoS ONE*, 15(10).

S. Gordin and A. Romach. 2022. Optical character recognition for complex scripts: A case-study in cuneiform. In *Digital Humanities 2022: Responding to Asian Diversity*, pages 212–215.

G. Gutherz, S. Gordin, L. Sáenz, O. Levy, and J. Berant. 2023. Translating akkadian to english with neural machine translation. *PNAS Nexus*, 2(5):1–10.

J. Huehnergard. 2011. *A Grammar of Akkadian*. Eisenbrauns.

L.W. King. 1900. *The Letters and Inscriptions of Hammurabi, King of Babylon, About B.C. 2200. Vol. II*. Number 3 in Luzac's Semitic Text and Translation Series. Luzac And Co.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

F.R. Kraus and K.R. Veenhof, editors. 1964–2005. *Albabylonische Briefe in Umschrift und Übersetzung*. 14 vols. Brill.

M. Luukko, A. Sahala, S. Hardwick, and K. Lindén. 2020. Akkadian treebank for early Neo-Assyrian royal inscriptions. In *Proceedings of the 19th International Workshop on Treebanks and Linguistic Theories*, pages 124–134, Düsseldorf, Germany. Association for Computational Linguistics.

M. Ong and S. Gordin. 2024. Linguistic annotation of cuneiform texts using treebanks and deep learning. *Digital Scholarship in the Humanities*, pages 1–12.

E. Reiner. 1966. *A Linguistic Analysis of Akkadian*. Number 21 in Janua Linguarum, Series Practica. Mouton & Co.

E. Reiner. 1973. How we read cuneiform texts. *Journal of Cuneiform Studies*, 25(1):3–58.

A. Sahala. 2021. *Contributions to Computational Assyriology*. Ph.D. thesis, University of Helsinki.

A. Sahala and K. Lindén. 2023. Babylemmatizer 2.0 – a neural pipeline for pos-tagging and lemmatizing cuneiform languages. In *Proceedings of the Ancient Language Processing Workshop associated with RANLP-2023*, pages 203–212.

A. Sahala, M. Silfverberg, A. Arppe, and K. Lindén. 2020. BabyFST - towards a finite-state based computational model of ancient babylonian. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3886–3894, Marseille, France. European Language Resources Association.

P. Singh, A. Maladry, and E. Lefever. 2023. Too many cooks spoil the model: Are bilingual models for Slovene better than a large multilingual model? In *Proceedings of the 9th Workshop on Slavic Natural Language Processing 2023 (SlavicNLP 2023)*, pages 32–39, Dubrovnik, Croatia. Association for Computational Linguistics.

T. Sommerschield, Y. Assael, J. Pavlopoulos, V. Stefanak, A. Senior, C. Dyer, J. Bodel, J. Prag, I. Androutsopoulos, and N. de Freitas. 2023. Machine learning for ancient languages: A survey. *Computational Linguistics*, pages 1–45.

M.P. Streck. 2010. Großes fach altorientalistik: Der umfang des keilschriftlichen textkorpus. *Mitteilungen der Deutschen Orient-Gesellschaft zu Berlin*, 142:35–58.

M.P. Streck. 2022. *Old Babylonian Grammar, Vol. 1*. Number 168.1 in Handbook of Oriental Studies, Section One: Ancient Near East. Brill.

M. Sukhareva, F. Fuscagni, J. Daxenberger, S. Görke, D. Prechel, and I. Gurevych. 2017. Distantly supervised pos tagging of low-resource languages under extreme data sparsity: The case of hittite. In *Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, Proceedings*, pages 95–104, Vancouver, Canada.

C. Swaelens, I. De Vos, and E. Lefever. 2023. Medieval social media: Manual and automatic annotation of byzantine Greek marginal writing. In *Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII)*, pages 1–9, Toronto, Canada. Association for Computational Linguistics.

P. Zemánek. 2007. A treebank of ugaritic. annotating fragmentary attested languages. In *Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories*.

10. Appendices

10.1. Results Related to Section 5.1, 5.2 and 5.3

Embeddings	Accuracy	Macro avg F1	Weighted avg F1
Multilingual	71,0%	25,6%	69,8%
Arabic	76,2%	32,8%	75,5%
Spanish	74,1%	30,6%	74,4%
Japanese	72,6%	28,3%	72,3%

Table 4: The average of the accuracy, macro average F1 and weighted average F1 over five folds from the results of predicting PoS and morphological tags with transliterated text as input and verb separated text (excluding the suffixes -*u* and -*ma*). The first column mentions the pre-trained FLAIR embeddings used for fine-tuning the model on. This table elaborates on the results relevant for Table 1.

Separation type	Accuracy	Macro avg F1	Weighted avg F1
Text	88,2%	52,3%	87,4%
Line	92,5%	73,6%	92,3%
Verb	94,8%	74,4%	94,6%
Verb – *u*	93,9%	70,1%	93,6%
Verb – *u* & *ma*	94,1%	75,9%	93,9%
Verb – *u* & *ma* & CNJ	93,5%	70,4%	93,3%

Table 5: The average accuracy, macro average F1 and weighted average F1 over five folds for predicting PoS-tags with Arabic pre-trained embeddings and transliterated text as input, for different sentence splitting approaches. This table elaborates on the results relevant for Table 2.

Multilingual			
Scores	Transliteration	Unicode	Loss
Accuracy	91,3%	61,3%	30%pt.
Macro avg F1	66,5%	22,8%	43,7%pt.
Weighted avg F1	90,9%	59,0%	31,9%pt.
Arabic			
Scores	Transliteration	Unicode	Loss
Accuracy	94,1%	69,4%	24,7%pt.
Macro avg F1	75,9%	31,8%	44,1%pt.
Weighted avg F1	93,9%	67,8%	26,1%pt.
Spanish			
Scores	Transliteration	Unicode	Loss
Accuracy	93,7%	63,0%	30,7%pt.
Macro avg F1	71,7%	23,4%	48,3%pt.
Weighted avg F1	93,4%	60,6%	32,8%pt.
Japanese			
Scores	Transliteration	Unicode	Loss
Accuracy	93,4%	73,9%	19,5%pt.
Macro avg F1	70,9%	40,7%	30,2%pt.
Weighted avg F1	93,1%	72,8%	20,3%pt.

Table 6: The average accuracy, macro average F1 and weighted average F1 over five folds from the results of predicting PoS-tags on verb-separated (excluding -*u* and -*ma*) sentences with different pre-trained embeddings and input in either transliteration or Unicode cuneiform. This table elaborates on the results relevant for Table 3.