Automated Phonological Transcription of Akkadian Cuneiform Text

Aleksi Sahala[1], Miikka Silfverberg[1,3], Antti Arppe[2], Krister Lindén[1]

[1]University of Helsinki, [2]University of Alberta, [3] University of British Columbia

{aleksi.sahala, krister.linden}@helsinki.fi, msilfver@mail.ubc.ca,arppe@ualberta.ca

Abstract

Akkadian was an East-Semitic language spoken in ancient Mesopotamia. The language is attested on hundreds of thousands of *cuneiform* clay tablets. Several Akkadian text corpora contain only the transliterated text. In this paper, we investigate automated phonological transcription of the transliterated corpora. The phonological transcription provides a linguistically appealing form to represent Akkadian, because the transcription is normalized according to the grammatical description of a given dialect and explicitly shows the Akkadian renderings for Sumerian logograms. Because cuneiform text does not mark the inflection for logograms, the inflected form needs to be inferred from the sentence context. To the best of our knowledge, this is the first documented attempt to automatically transcribe Akkadian. Using a context-aware neural network model, we are able to automatically transcribe syllabic tokens at near human performance with 96% recall @ 3, while the logogram transcription remains more challenging at 82% recall @ 3.

Keywords: transcription, Akkadian, cuneiform script

1. Introduction

Akkadian was an East-Semitic language spoken in ancient Mesopotamia. The language is attested from hundreds of thousands of *cuneiform* clay tablets and their fragments excavated from modern day Iraq and the surrounding regions. Although the first exemplars of Akkadian texts date back to 2400 BCE, two later dialects, i.e. Babylonian (1900 BCE - 100 CE) and Assyrian (1950 – 600 BCE) provide the largest source of available text material. Akkadian was written in the cuneiform script mostly on clay tablets as the example shown in Fig. 1. In modern electronic text corpora, cuneiform text is represented in two distict latinizations: (1) sign-to-sign level graphemic transliteration and (2) phonemic transcription based on an approximation of the Akkadian language and its reconstructed sound system (Kouwenberg, 2011).

A large number of transliterated Akkadian texts are freely available in digitized corpora, however, there exist far fewer phonologically transcribed texts. The principal reason for this discrepancy is that the transcription process is labor-intensive, and the transcription itself is not considered very useful for scholarly interpetation of the cuneiform texts. It is, however, indispensable for the development of computational resources for Akkadian because cuneiform script often omits phonological elements which provide important information for language analysis tools. For example, vowel quantities are not marked consistently, and dialects may use slightly different writing conventions (Kouwenberg, 2011). Moreover, the underlying readings of logograms are not transliterated. This information is added during phonological transcription. In addition to computational modeling, phonological transcriptions also play an important role in teaching of the Akkadian language.

In this paper we investigate automated phonological transcription of transliterated Akkadian texts. Conceptually, transcription consists of two subtasks. For syllabically spelled words which contain only syllabic cuneiform signs, transcription is a string transduction task closely related to grapheme to phoneme (G2P) conversion. For logograms,

Figure 1: Old Assyrian cuneiform clay tablet (Spar, 1988)

transcription chiefly reduces to context dependent dictionary lookup. We apply four different supervised machine learning models to the transcription task: a statistical model, which aims to model the correspondence of transliterations and transcriptions using string substitution operations, and three deep learning models based on the encoder-decoder framework. Our models deliver promising results. Using a training set of some 270k tokens, our best models can accurately transcribe around 90% of the syllabic tokens and 69% of the logograms in a held-out test set.

The main motivation for our work is to facilitate annotation of Akkadian text corpora using tools like syntactic parsers and morphological analyzers which typically operate on transcribed, not transliterated text. To assess the usefulness of automated transcription, we measured the performance of a rule-based Akkadian morphological analyzer BabyFST (Sahala et al., 2019 submitted) on our transcribed text. Our experiments indicate that the analyzer can retrieve the correct morphological analysis for 94% of the automatically transcribed tokens.

2. Akkadian

This section presents the Akkadian language and the phonological transcription task.

2.1. Transliteration

Cuneiform script is represented in text corpora in a standardized transliteration,[1] which aims to provide maximum objectivity for researchers who cannot access the primary sources. All imperfections, reconstructions and crucial ortographic exceptions such as scribal errors are marked consistently using common conventions. These conventions are not only used in Akkadian, but also in Sumerian, Elamite, Hittite and many other cuneiform languages.

Homophonic signs are indexed with standardized subscript numbers after each transliterated sign (e.g. bi_2 is a distinct sign from *bi*), and logograms are transliterated in capital letters according to their earlier Sumerian names to distinguish them from syllabic values. Syllabic signs are separated from each other with dashes, and logogram compounds with dots, as in DUMU.MUNUS-*ia* 'my daughter' (Akk. *mārtīja*).

Most signs can be read both as syllables and logograms depending on context. For example, the sign ⊬ depicting a star can be read as a logogram DINGIR 'god' (Akk. *ilu*) or AN 'heaven' (Akk. *šamû*), but also as a syllable *an*. Moreover, a limited set of signs can also be used as determinatives to categorize a preceding or a following word. Determinatives can, for example, indicate that the word they are attached to is a wooden object or tree, stone or mineral, city, bird, river, profession, personal name, or a deity, just to mention a few (Kouwenberg, 2011). This, originally Sumerian system was useful, as it allowed disambiguation between possible readings of a logogram: LUL = *sarru* 'false', {*lu₂*}LUL = *parriṣu* 'criminal, wrongdoer' (literally {man}FALSE). In modern digital text corpora, determinatives are written in braces, although paper publications favor indicating them in superscript.

2.2. Phonological Transcription

Phonological transcription provides a linguistically more appealing form to represent Akkadian texts than transliteration, because it normalizes the language according to the grammatical description and shows explicitly Akkadian renderings for various logograms. Because the cuneiform script rarely marks inflection for logograms, the correct inflected form needs to be inferred based on its syntactic context. For syllabically spelled words, transcription involves reconstructing their correct phonemic quantities.[2] The example below illustrates how Standard Babylonian[3] Akkadian transliteration is rendered as a phonological transcription.

Transliteration: (5) {d}AMAR.UTU *kab-ta-ta i-na* DINGIR.DINGIR GAL.GAL (6) *ši-mat-ka la ša-na-an si₃-qar-ka* {d}*a-nu-um*

Transcription: (5) *Marduk kabtāta ina ilāni rabûtim* (6) *šīmatka lā šanān siqarka Anum*

Translation: (5) Marduk, you are the most honoured among all the great gods, (6) your destiny is unequalled, your command is like that of Anu! (Enūma Eliš, tablet IV 5-6, cf. (Talon, 2005))

Phonological transcription is essential for automatic morphological analysis of Akkadian, as several grammatical aspects of the language depend on the vowel and consonant quantity which is not always consistently spelled out in cuneiform. For example, singular and plural nouns are distinguished only by vowel length, which is often not explicitly marked in writing. The word 'man' may be spelled syllabically *a-wi-lu* or *a-wi-li*, having phonemic renderings *awīlu* (nominative sg.), *awīlū* (nominative pl.)' and *awīli* (genitive, sg.), *awīlī* (oblique pl.) respectively. The same applies to verbs, which distinguish third person masculine plural and subjunctive, both common forms, only by vowel length: *inaddinū* 'they (masc.) give', *inaddinu* 'he gives (subjunctive)'. More ambiguity emerges from final weak verbs, where the final vowel can be part of the stem: *imannu* 'he counts', *imannû* 'they (masc.) count; he counts (subjunctive)'.

Words may be spelled syllabically, logographically or logo-syllabically. Moreover, phonemic syllables can also spelled in various ways by using the four basic sign types (V, VC, CV, CVC) or their combinations (V-VC, CV-V, CV-(V-)VC).[4] For instance a verbal form *iddin* 'he gave' may be rendered as *id-din*, *i-din*, *id-di-in*, *i-di-in*, SUM and SUM-*in*, and a verbal form *idâk* 'he kills' as *i-dak*, *i-da-ak*, GAZ, GAZ-*ak* or UŠ₂.[5] One can immediately see that, for instance a syllabic sign compound CV-VC or a sign CVC can represent a closed syllable with different vowel lengths. Similarly, a singly spelled consonant can be a phonological geminate (this applies also vice versa). Although in many cases syllabic graphemes align somewhat nicely with the phonemic representation if the phonemic length is not taken into account, weak consonants /w j '/ are not regularly visible in the spelling and they have to be inserted in the phonological transcription: *da-a-(a-)nu = dajjānu*, *ma-a-da-tu = ma'dātu* (Kouwenberg, 2011).

The most difficult words to transcribe are logograms, which do not give any information on their surface form unless some of their morphology is explicitly spelled out as in SUM-*in*, which restrict the possible grammatical forms to those ending with /in/. Still, this may leave out numerous possibilities: in the case of SUM-*in* the personal prefix is not spelled out at all, and the theoretically possible forms could be, among dozens of others, *taddin* 'you gave', *attadin* 'I have given', *inaddin* 'he gives', *idin* 'give!' and

---

[1]There are in fact two systems: the von Soden system (von Soden, 1995) and the Gelb system (Gelb, 1970). Their differences mostly concern the readings of isolated signs in specific dialects (Kouwenberg, 2011).

[2]In addition to long /ā ī ē ū/ and short /a i e u/ vowels, Akkadian has a set of contracted vowels /â ê î û/ that originate from merged vocalic clusters (von Soden, 1995).

[3]Standard Babylonian was an artificial literary language used by the Babylonians and the Assyrians (Kouwenberg, 2011).

[4]Forms with explicitly spelled long vowels, as *ba-nu-u₂* for *banû* 'to create' are called plene-writing.

[5]The spelling variants have been collected from Oracc.

nadin 'it is being given'. The correct reading is only interpretable by the syntactic context of the word.

2.3. BabyFST

BabyFST (Sahala et al., 2019 submitted) is the most comprehensive finite-state based morphological analyzer for Akkadian up-to-date. Its source code is written according to the Xerox finite-state syntax with the *lexc* and *xfscript* formalisms, and can be compiled using the Foma Finite-State tookit (Hulden, 2009) or HFST - the Helsinki Finite State Transducer toolkit (Lindén et al., 2011). BabyFST is designed primarily for the Babylonian dialect and is capable of processing phonological words. Several Akkadian text corpora contain only the transliteration, so some robust support is needed for producing phonologically transcribed text for BabyFST to analyse.

3. Related Work

To the best of our knowledge, this is the first system for automatically transcribing Akkadian cuneiform text but Smith (2007) presents a system for grapheme-to-phoneme transcription for the Elamite language, another extinct language using the cuneiform script. The system uses optimality constraints and applies the gradual learning algorithm for learning a ranking between the constraints.

Many grapheme-to-phoneme transcription systems formulate the task as a probabilistic sequence model on grapheme-phoneme pairs. For example, Bisani and Ney (2008) use joint n-gram models and Novak et al. (2012) use weighted finite-state transducers. These methods rely on some alignment between the grapheme and phoneme strings because they gather statistics about symbol alignments.[6] Because our strings contain logographic material which may not have a clear connection to the phonological representation, alignment is challenging. Therefore we opted for using models based on the neural encoder-decoder architecture. These models like Rao et al. (2015) directly model the relation between the input grapheme sequence and output phoneme sequence which does not require string alignment.

Unlike many grapheme-to-phoneme transcription settings, the transcription of cuneiform Akkadian text differs in the sense that the transcription of logograms is partly dependent on sentence context as explained in Section 2.2.. This suggests that it is beneficial to model the sentential context of the transliterated input token. In this respect, our task is related to historical text normalization where context can often be helpful. Korchagina (2017) investigates neural machine translation for historical text normalization. In contrast to token-based normalization methods applied in grapheme-to-phoneme transcription, the model here is a character-level encoder-decoder which is applied on complete sentences. This approach proved to be challenging in the context of Akkadian transcription based on our preliminary experiments. The character-based NMT model would overfit the training data possibly because of our rela-

tively small dataset.[7] Instead of a full fledged NMT model, we present experiments on several more restricted ways to model the sentence context.

4. Models

Conceptually, automatic transcription of Akkadian consists of two subtasks. On the syllabic level, it is a grapheme to phoneme (G2P) transcription task, and on the lographic level it chiefly reduces to context dependent dictionary lookup. We apply five models to the transcription task. These five models are compared against a straightforward majority baseline. Our first model is a statistical model which learns associations between the graphemes in transliterations and phonemes in transcriptions. This models chiefly targets the grapheme to phoneme transcription task. Our remaining models are character-based neural attentional encoder-decoder systems which translate transliterated words into transcribed words. These models are able to address both the grapheme to phoeneme and context dependent lookup of logograms. The neural models differ with regard to context modeling. The first one performs transcription without considering the sentence context, the second one conditions its transcription on the neighboring tokens in the sentence and the third one represents the context in a character-based manner.

4.1. Baseline

The baseline system simply memorizes all transcriptions in the training set. During testing, it generates the most frequent transcription for the input transliteration if the transliteration occurred in the training set. Otherwise, it generates the empty sequence. When generating more than one output candidate, the system orders them by their frequency in the training set.

4.2. Statistical Model (Stat)

Our statistical model maps syllabic transliteration–transcription correspondences, such as *ta-pa-ra-si* : *taparrasī* into relations of abstract sequences: $C_1a-C_2a-C_3a-C_4i : C_1aC_2aC_3C_3aC_4\bar{\imath}$.[8] If the correct transcription is not found in the baseline dictionary, the mapping is often able to generalize transcriptions with correct vowel and consonant lengths for all unseen words that belong to the same conjugation type and occur in the same form as in the training data. This works well in Akkadian, as its verbal morphology is non-linear and based on a rather regular interdigitation of root radicals into patterns. For example, the above-mentioned abstraction can be used to transcribe the verbal form *ta-ga-ma-ri* into *tagammarī*, as the verb *gamāru* belongs to the same vowel class (a/u) and conjugation type (strong) as our example verb *parāsu*. Because transliterations can be mapped to several transcriptions (e.g. *i-par-ra-su* : *iparrasu, iparrasū*), each pair is assigned a probability based on

---

[6] Although this alignment can be latent as in Bisani and Ney (2008).

[7] For example, we observed several occasions where the output of the NMT system was considerably longer or shorter than the input sentence and several examples of model hallucination. This resulted in very bad performance for the NMT system.

[8] The mapping is simply based on character counting. Sophisticated machine learning methods are not involved.

	Baseline	Stat	Enc-Dec	Enc-Dec+Context	Enc-Dec+Char-Context
Recall @ 1	81.37	87.25	89.44	**90.01**	89.59
Recall @ 2	83.65	91.13	**95.44**	95.03	94.77
Recall @ 3	83.74	91.93	**96.65**	96.19	95.91
Recall @ 5	83.75	92.55	**97.61**	97.11	96.78
Recall @ 10	83.75	92.55	**98.14**	97.58	97.33

Table 1: Transcription results for syllabic tokens. Baseline refers to the baseline system, Stat to the statistical system, Enc-Dec to the plain neural system and Enc-Dec+Context and Enc-Dec+Char-Context to the context-aware neural systems.

	Baseline	Stat	Enc-Dec	Enc-Dec+Context	Enc-Dec+Char-Context
Recall @ 1	60.70	60.64	57.72	**69.10**	68.70
Recall @ 2	76.10	76.10	74.13	**78.59**	78.26
Recall @ 3	82.15	**82.16**	81.14	81.97	81.86
Recall @ 5	85.96	**86.03**	85.65	84.42	84.26
Recall @ 10	**88.90**	**88.90**	88.79	86.09	86.17

Table 2: Transcription results for logograms. Baseline refers to the baseline system, Stat to the statistical system, Enc-Dec to the plain neural system and Enc-Dec+Context and Enc-Dec+Char-Context to the context-aware neural systems.

the training data, which can be used to choose the most likely option. The model fails to generalize logograms and syllabic sequences which do not align consonant by consonant with the transcription and ignores such pairs altogether. This is often the case if the editor of the text has made normalizations or corrections in the transcription phase, as in an archaistic normalization of *ni-šu* 'people' to *nišum*.

4.3. Encoder-Decoder (Enc-Dec)

Our first neural system is a character-based attentional encoder-decoder system (Bahdanau et al., 2014) which translates a sequence of transliterated characters like *t a - m i t* into a sequence of transcribed characters *t ī m ī t*.[9] Here we do not distinguish between syllabic and logograms: also logoograms like *DINGIR* 'god' are split into character sequences: *D I N G I R*. As encoder and decoder networks, we use single layer LSTM networks.

4.4. Encoder-Decoder with Context Modeling (Enc-Dec+Context)

Our second encoder-decoder system conditions its transcription on neighboring words. We accomplish this by encoding them into the input sequence. For example,

$$i\,p\,-\,p\,a\,-\,l\,a\,[\,lib_3 - bi - \check{s}u\,]\,<\,\{d\}e_2 - a\,>$$

where $i\,p\,-\,p\,a\,-\,l\,a$ is the input token represented as a character sequence, $lib_3 - bi - \check{s}u$ is the transliterated token immediately preceding the input word and $\{d\}e_2 - a$ is the transliterated token immediately following the input word. These are represented as monolithic tokens. We decided to limit the context to neighboring words instead of the complete sentence due to the limited amount of available train-

ing data. The structure of the network and training details are identical to the first encoder-decoder model (Enc-Dec).

4.5. Encoder-Decoder with Character-Level Context Modeling (Enc-Dec+Char-Context)

Our final model is very similar to the Enc-Dec+Context model except that it encodes both the input word and the context words as character sequences instead of monolithic tokens. As an example of input, consider:

$$i\,p\,-\,p\,a\,-\,l\,a\,[\,l\,i\,b_3\,-\,b\,i\,-\,\check{s}\,u\,]\,<\,\{d\}\,e_2\,-\,a\,>$$

Again the structure of the network and training details are identical to the first encoder-decoder model (Enc-Dec).

5. Experiments

This section presents the experimental setup and the results of the experiments.

5.1. Intrinsic Evaluation of Transcriptions

We perform transcription experiments on a subset of the Open Richly Annotated Cuneiform Corpus (Oracc) (Oracc, 2014) which provides both transliterations and transcriptions for tokens. The dataset spanning 21,078 sentences (337,260 tokens) was randomly divided into a training set (271,311 tokens), a development set (31,884 tokens) and a test set (34,065 tokens).[10]

For our neural models, we use single layer LSTM networks (Hochreiter and Schmidhuber, 1997) with character embedding dimension and hidden dimension 512, and we optimize the network using the Adam algorithm (Kingma and Ba, 2014). We train for 100k steps with batch size 30. In all experiments, we use beam width 20 during decoding.

We evaluate the transcription based on whole token accuracy and present the results as recall of the gold standard transcription at n (Recall @ n) based on extracting the n

---

[9]Even though the units in the transliterated sequence denote syllables, we decided to model transliterations as character sequences. This allows the network to easily learn phonetic relations between related syllables like ta and la.

[10]The dataset consisted of transcribed Standard Babylonian texts in Oracc.

	Gold	Baseline	Stat	Enc-Dec	Enc-Dec+Context	Enc-Dec+Char-Context
Recall @ 1	96.55	76.66	84.40	87.25	**89.85**	89.30
Precision @ 1	41.19	38.75	38.33	37.22	**38.82**	38.79

Table 3: Recall and precision of the morphological analysis (on Lemma+POS level) based on the top-1 predicted transcriptions.

	Gold	Baseline	Stat	Enc-Dec	Enc-Dec+Context	Enc-Dec+Char-Context
Recall @ 3	96.55	80.05	89.70	**94.31**	93.70	93.45
Precision @ 3	41.19	**35.10**	34.73	31.54	30.49	29.64

Table 4: Recall and precision of the morphological analysis (on Lemma+POS level) based on the top-3 predicted transcriptions.

	Gold	Baseline	Stat	Enc-Dec	Enc-Dec+Context	Enc-Dec+Char-Context
Recall @ 10	96.55	80.60	90.50	**96.50**	95.55	95.80
Precision @ 10	41.19	**31.42**	31.12	26.34	22.24	22.19

Table 5: Recall and precision of the morphological analysis (on Lemma+POS level) based on the top-10 predicted transcriptions.

best output candidates from the model. We present results separately for syllabic tokens and logograms in Tables 1 and 2, respectively.[11]

As seen in Table 1, the Enc-Dec+Context delivers the best accuracy (Recall @ 1) for syllabic tokens closely followed by the other two neural models. All neural models and the statistical model deliver substantial improvements over the baseline majority voting system. The neural models deliver superior performance compared to the statistical model. When examining recall @ n for $n > 1$, the Enc-Dec model which ignores the sentence context delivers the best results, but all neural models are on par.

As seen in Table 2, the Enc-Dec+Context model delivers the best accuracy for logograms. It is closely followed by the Enc-Dec+Char-Context model. Both of the contextual neural models deliver clear improvements over the baseline majority voting system for recall @ 1 and recall @ 2. However, for recall @ n, where $n > 2$, the majority voting strategy wins closely followed by the Enc-Dec model.

5.2. Extrinsic Evaluation of Transcriptions

We evaluated the automatically produced transcriptions by feeding 2,000 auto-transcribed tokens into BabyFST and by comparing the recall and precision with human-transcribed text (Gold).[12] The gold results exhibit maximum possible recall (96.55) and precision (41.19) scores achievable with BabyFST at its current state with our test data, and show the extent of the inherent morphological ambiguity of the language, even when using unambiguous human-transcribed input. As we did not have a morphologically annotated gold standard available, our goal was to produce a lemma and POS-tag that matches the human-made annotations in

Oracc. We measure **recall** to indicate the percentage of tokens that were given at least one correct analysis and **precision** to show the percentage of correct analyses over all given analyses. The results are presented in Tables 3, 4 and 5.

A recall very close to human-transcribed text is achievable by using the Enc-Dec model with top-10 predictions. Neural net models have a more negative impact on precision than the statistical model due to the fact that the statistical model does not often produce the full number of predictions. Choosing a smaller selection of predictions naturally improves precision at the cost of recall.

6. Conclusions and Future Work

Our experiments show that transcription for syllabic tokens is clearly very successful delivering 90% recall @ 1 and 96% recall @ 3. This is likely to be of considerable help in semi-automatic phonological transcription.

For logograms, results are weaker than for syllabic tokens but we can still clearly outperform the baseline which shows that context modeling is beneficial. Character-based context modeling delivered very similar results to the simpler token-based context modeling. This is consistent with the observation that indeclinable words like prepositions, as well as other logograms, tend to be the most important cues for the inflectional information associated with nearby words.

For transcription of logograms, the best approach seems to be a combination of context-modeling for the top candidate (or possibly top-2 candidates) and baseline majority voting for extracting additional transcription candidates. As Table 2 shows, context modeling provides a substantial improvement of 8%-points over other models, when we extract a single transcription candidate. When extracting more candidates, the performance of the various systems is quite similar.

---

[11]In Table 2, we present results for all examples where the transliterated input token contains a logogram. The input token can sometimes additionally contain syllabic material.

[12]The dataset was extracted from Oracc and comprised 655 logograms and 1345 syllabic words

The coverage of BabyFST on automatically transcribed text is high. The gold standard lemma and POS can be recovered for 90% of tokens when a single candidate transcription is considered and 94% of the tokens when three candidate transcriptions are considered. This comes at the cost of some added ambiguity: three analyses per input translation. This can be considered acceptable given the high coverage. Further rule-based or statistical disambiguation of analyses could limit the ambiguity but this remains future work at this time. Disambiguation must also be done at the morphological level, as several lemmas may have multiple morphological analyses due to ambiguous spellings. This is, however, a task that is more closely connected to the future development of BabyFST. Morphological disambiguation of Akkadian will likely become easier in the near future, as a manually morphologically annotated corpus of Akkadian royal inscriptions is currently in the making by the Akkadian Treebanking project in the University of Helsinki.[13] The morphological annotation can be used for weighting the BabyFST transducer, which hopefully solves a part of the ambiguity alone.

Some of the ambiguity and unwanted predictions emerge from oddities in the training data. For example, occasionally human-transcribers have removed mimation (*an-nu-tum : annūtu*), added morphemes (*li-tir : litīrma*), changed the case of a noun (*a-wi-lu : awīli*) or transcribed words inconsistently *me-lam-mu : melemmu, melammu*. Although these decisions are usually justified in the original context, the models can generalize these discrepancies in unwanted contexts, e.g. for a coincidentally similar looking lexeme. Thus, cleaning and more careful selection of the training data would likely have a positive impact on the results. In the future, it would also be beneficial to train models for different dialects and time periods, as they feature a number of consistent differences in their spelling.

One potential way to improve the results would be to try multitask learning by combining simple dictionary-lookups and predictive models. For example, a logogram could be first lemmatized and POS-tagged based on the context, and this information could later be used to disambiguate the predicted inflected forms. Also in syllabic renderings, many transcriptions map unambiguously to one phonological representation. Thus it would be useful to minimize ambiguity by first using a large dictionary lookup (e.g. consisting of the whole corpus of a given dialect in Oracc), and then trying to predict the correct phonological rendering only if the transcription is clearly ambiguous.

Finally, although the present work concerns phonological transcription of Akkadian cuneiform text, the approach is by no means specific to Akkadian texts. It could equally well be applied to other languages which used cuneiform script like Elamite and Hittite.

Acknowledgements

Bibliographical References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. cite arxiv:1409.0473.

Bisani, M. and Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech communication*, 50(5):434–451.

Gelb, I. J. P. (1970). Comments on the akkadian syllabary. *Orientalia*, 39:516–546.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hulden, M. (2009). Foma: a finite-state compiler and library. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, pages 29–32. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Korchagina, N. (2017). Normalizing medieval German texts: from rules to deep learning. In *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, pages 12–17, Gothenburg, May. Linköping University Electronic Press.

Kouwenberg, B. (2011). Akkadian in general. In Stefan Weninger, editor, *Semitic Languages*, chapter 12, pages 330–339. De Gryuter Mouton, Germany.

Lindén, K., Axelson, E., Hardwick, S., Pirinen, T. A., and Silfverberg, M. (2011). HFST—Framework for Compiling and Applying Morphologies. In Cerstin Mahlow et al., editors, *Systems and Frameworks for Computational Morphology*, pages 67–85, Berlin, Heidelberg. Springer Berlin Heidelberg.

Novak, J. R., Minematsu, N., and Hirose, K. (2012). Wfst-based grapheme-to-phoneme conversion: Open source tools for alignment, model-building and decoding. In *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, pages 45–49.

Rao, K., Peng, F., Sak, H., and Beaufays, F. (2015). Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4225–4229. IEEE.

Sahala, A., Silfverberg, M., Arppe, A., and Lindén, K. (2019 (submitted)). BabyFST - towards a finite-state based computational model of ancient babylonian. In *LREC*.

Smith, E. (2007). Phonological reconstruction of a dead language using the gradual learning algorithm. In *Proceedings of Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology*,

---

[13]PI of the project is Niek Veldhuis (UC Berkeley / University of Helsinki) and the manual annotation is being made by Mikko Luukko (University of Helsinki).

---

[14]PI of both of these projects is Saana Svärd (University of Helsinki).

pages 57–64, Prague, Czech Republic, June. Association for Computational Linguistics.

Spar, I. (1988). *Cuneiform Texts in the Metropolitan Museum of Art. Volume I: Tablets, Cones, and Bricks of the Third and Second Millennia*. The Metropolitan Museum of Art.

Talon, P. (2005). *The Standard Babylonian Creation Myth: Enūma Eliš*. State Archives of Assyria Cuneiform Texts IV. The Neo-Assyrian Text Corpus Project, Helsinki.

von Soden, W. (1995). *Grundriss der Akkadischen Grammatik*. Editrice Pontificio Instituto Biblico, Rome, 3 edition.

Language Resource References

Oracc. (2014). Oracc: The open richly annotated cuneiform corpus. http://oracc.org.