BabyFST : Towards a Finite-State Based Computational Model of Ancient Babylonian

Sahala, Aleksi

European Language Resources Association (ELRA)
2020-05-17

Sahala , A , Silfverberg , M , Arppe , A & Linden , K 2020 , BabyFST : Towards a Finite-State Based Computational Model of Ancient Babylonian . in N Calzolari ... [et al.] (ed.) , Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020) . European Language Resources Association (ELRA) , Paris , pp. 3886-3894 , International Conference on Language Resources and Evaluation , Marseille , France , 11/05/2020 . < https://www.aclweb.org/anthology/2020.lrec-1.479/ >

http://hdl.handle.net/10138/317691

BabyFST - Towards a Finite-State Based Computational Model of Ancient Babylonian

Aleksi Sahala[1], Miikka Silfverberg[1], Antti Arppe[2] & Krister Lindén[1]
[1]University of Helsinki, [2]University of Alberta
{aleksi.sahala, miikka.silfverberg, krister.linden}@helsinki.fi, arppe@ualberta.ca

Abstract

Akkadian is a fairly well resourced extinct language that does not yet have a comprehensive morphological analyzer available. In this paper we describe a general finite-state based morphological model for Babylonian, a southern dialect of the Akkadian language, that can achieve a coverage up to 97.3% and recall up to 93.7% on lemmatization and POS-tagging task on token level from a transcribed input. Since Akkadian word forms exhibit a high degree of morphological ambiguity, in that only 20.1% of running word tokens receive a single unambiguous analysis, we attempt a first pass at weighting our finite-state transducer, using existing extensive Akkadian corpora which have been partially validated for their lemmas and parts-of-speech but not the entire morphological analyses. The resultant weighted finite-state transducer yields a moderate improvement so that for 57.4% of the word tokens the highest ranked analysis is the correct one. We conclude with a short discussion on how morphological ambiguity in the analysis of Akkadian could be further reduced with improvements in the training data used in weighting the finite-state transducer as well as through other, context-based techniques.

Keywords: Finite-State Transducer, Computational Modeling, Akkadian, Morphology

1. Introduction

In this paper, we present a finite-state based general model for Babylonian morphology. At first we give a brief description of the Babylonian and its morphological features. Then we present the modeling principles of our FST approach and measure its performance against a manually tagged corpus. At last, we will discuss potential ways to improve, expand and use the analyzer.

2. Brief Description of the Babylonian Language

Akkadian, the language of ancient Babylonians and Assyrians, is known to us by several hundreds of thousands of clay tablets, their fragments and other inscriptions written on various media in cuneiform script. This text material covers a timespan of over two millennia (2400 BCE to 100 CE), making Akkadian one of the world's earliest written languages, clearly predated only by Sumerian, Elamite and Egyptian (all written already in the fourth millennium BCE). Although a vast majority of the excavated material consists of administrative documents, the Akkadians also left behind a vivid selection of cultural historically significant literary works and inscriptions, including the Epic of Gilgameš and the Code of law of Hammurāpi. Alongside Eblaite, Akkadian is the only known, and the longest surviving member of the now-extinct East-Semitic branch of languages. As a spoken language, Akkadian was gradually replaced by Aramaic after the Persian conquest of Babylonia in 539 BCE, but its prestigious status kept it alive as a literary language until the first century CE. Unlike Sumerian, of which the grammatical description is still widely debated, the understanding of Akkadian morphology and grammar is well-established (cf. grammatical analyses by Reiner 1966, von Soden 1995 and Kouwenberg 2010).

2.1 Historical development

Babylonian, a dialect of the Akkadian language, is divided into several stages connected with the South Mesopotamian historical events: Old Babylonian (OB: 1900–1600 BCE), Middle Babylonian[1] (MB: 1500–1100 BCE), Neo-Babylonian (NB: 1000–600 BCE) and Late Babylonian (LB: 600–100 CE) (Kouwenberg 2011: 332). Additionally, there was an artificial literary language known as Standard Babylonian (SB), which was used by the Babylonians and Assyrians from the middle of the second millennium onwards. Standard Babylonian had its roots in Old Babylonian, but it was later affected by some lexical, phonological and morphophonemic features of the contemporary spoken Babylonian (especially MB and NB) and Neo-Assyrian dialects (Kouwenberg 2011: 332).

Different stages of Babylonian are distinguished from each other mostly by morphophonemic alternation and phonological changes. During the Old Babylonian period, vocalic clusters became generally contracted: *rabīum* → *rabûm* 'great', *iqīaš* → *iqâš* 'to bestow' (Buccellati 1996: 37). Mimation, an archaic Semitic word-final morpheme {m} was gradually lost and it disappeared regularly by the Middle Babylonian period: *rabûm* → *rabû* (Streck 2011: 374). From this period onwards, Babylonian was affected by an increasing

[1] MB was used as a diplomatic *lingua franca* in the Middle-East and the surrounding regions.

number of assimilations (*imtagar* → *indagar* 'it was expensive'), dissimilations (*ibbi* → *imbi* 'he named'), as well as other sound changes (*issi* → *ilsi* 'he shouted', *wabālu* → *abālu* 'to carry') (Buccellati 1996: 37). In the course of the Neo- and Late Babylonian periods, the case system was reduced and the distinction especially between nominative {u} and accusative {a} was lost. Additionally, many short vowels were omitted in word-final positions: *abbēšunu* 'their (masc.) fathers' → *abbēšun*, *šarrātu* 'queens' → *šarrāt* (Streck 2011: 385).

BabyFST is mainly designed to model Standard Babylonian, which means that it covers most of the synchronic and diachronic variation within different stages of the Babylonian dialect, as well as some common Assyrian features found in Standard Babylonian texts known as Assyrianisms.

2.2 Akkadian morphology

Akkadian features a typical Semitic non-linear morphology particularly in its verbal system. Verbal roots consist of three or four radical consonants (referred henceforth as radicals) and a vowel class, which are interdigitated into templates in order to produce verbal stems and their derivations. Some of the most common derivations include the G-stem (basic stem), D-stem (factitive, transitive), Š-stem (causative) and N-stem (mostly passive) derivations, which all occur in different tenses (present, preterite, perfect), moods (indicative, imperative) and nominal forms (infinitive, active participle, verbal adjective, stative), and can be modified with additional *-t-* and *-tan-* infixes to produce more nuanced meanings: iterative, intensive, reciprocal, causative passive etc. (von Soden 1995). Verbal stems are conjugated by applying prefixes, suffixes and circumfixes, which are used to mark subject, object, indirect object, direction of movement (ventive), certain modal aspects, subjunction, and conjunction with other verbs (Table 1).

SLOT 1	vetitive marker
SLOT 2	personal prefix
SLOT 3	N/Š/ŠD-stem preformative
SLOT 4	verbal stem
SLOT 5	personal suffix or subjunctive
SLOT 6	ventive (direction)
SLOT 7	dative (indirect object)
SLOT 8	accusative (direct object)
SLOT 9	enclitic conjunction *-ma*

Table 1: Morphotactics of the Akkadian verb.

Personal conjugation distinguishes between two genders: masculine and feminine, and three numbers: singular, dual and plural, although the use of dual is very restricted. There are distinct personal affixes for indicative, precative, imperative and stative moods, of which prefixal parts have illabial /i, a/ and labial /u/ variants used in conjunction with different stems: G: *i-prus*, N: *i-pparis*, D: *u-parris*, Š: *u-šapris* (all 3rd person singular masc.).

Non-derivative nominal morphology is linear with a few archaic exceptions. Thus, Akkadian nouns do not form plurals by interdigitation as some other Semitic languages like Arabic or Maltese do. Nouns may take a feminine marker, an abstract or particularizing suffix, a dual or plural marker, a case ending and a possessive suffix, as well as a few archaic local case suffixes. They may also be used as statives to form predicative clauses: *šarru* 'king' → *šarrāku* 'I am king'.

The most complicated part of the Akkadian grammar is its verbal morphology. Radicals can be either strong or weak, of which the latter, /ʾ w j/, are subject to several (morpho)phonemic alternations that make surface forms fairly opaque. Most weak verbs contain only one weak radical, but there are also several verbs that consist of two or even three weak radicals. In typical cases, a weak radical is lost completely or assimilated into an adjacent consonant, and the surrounding vowels are contracted together, lengthened and/or colored. Consider the following G-stem preterites in 3rd person plural masculine: {i-bniʾ-ū} → *ibnû* 'they built', {i-wšib-ū} → *ūšibū/ušbū* 'they sat down', {i-wṣiʾ-ū} → *ūṣû* 'they set forth'. Additional assimilations occur at morpheme boundaries: {ta-ʾrub-ma} → *tērumma* 'you entered'; {i-ndin-kim-šu} → OB *iddikkiššu,* MB+ *imdikkiššu* 'he gave it (masc.) to you (fem.)' (Buccellati 1996).

2.3 Graphemic and phonemic representation

As Akkadian was written in logosyllabic cuneiform script, it is represented in Latin characters as a graphemic **transliteration**. The basic guidelines for transliteration are quite uniform and standardized. For instance, E_2 *ra-bu-um* 'big house' consists of a logogram written in capitals, followed by a subscript that distinguishes the sign from homophonous signs E and E_3. The three signs in italics represent syllabic values, and square brackets indicate that the cuneiform sign representing *bu* has been destroyed and reconstructed by the editor. In phonological **transcription**, this is read *bītum rabûm*. Long vowels (and occasionally even geminates) are not consistently spelled out in cuneiform, and thus the transcriber has to have knowledge of the Akkadian language in order to produce valid transcriptions. Yet, there are some differences in how words are transcribed. Some scholars do not distinguish between long /ā ī ū ē/ and contracted /â î û ê/ vowels (e.g. Buccellati 1996), and there are inconsistencies between transcribing the vowels /i/ and /e/, mostly because cuneiform writing did not always make a distinction between them: BI is read *bi* or be_2; RI = *ri*, *re*; NI = *ni*, ne_2; KI = qi_2, qe_2 just to mention a few.

At this stage, BabyFST operates only on the transcription. This is adequate, as there are several thousands of transcribed texts in Oracc (the Open Richly Annotated Cuneiform Corpus, see 2.4.) available.

2.4 Resources[2]

Considering the fact that Akkadian is an extinct language studied by a small research community, it is fairly well resourced. Currently the largest digital resources for Akkadian are ARCHIBAB (30k Babylonian texts), CDLI - Cuneiform Digital Library Initiative (320k texts of which 76k are labeled as Akkadian), SEAL - Sources of Early Akkadian Literature (550 compositions), and Oracc, which is a collection of texts from dozens of different projects.[3] Oracc comprises 1.98M tokens (17k texts) in various cuneiform languages. Of these, 1.67M tokens are labeled as various dialects or stages of Akkadian and 783k as different stages of Babylonian. In total, 1.42M Akkadian and 614k Babylonian tokens have been lemmatized and POS-tagged.[4] For Neo-Assyrian, the most notable collection of texts is the State Archives of Assyria online (504k tokens), initiated already in 1986 by Simo Parpola in Helsinki and later lemmatized and added to Oracc by Karen Radner and her team. Currently, none of the afore-mentioned corpora contain a morphological analysis of Akkadian beyond lemmatization and POS-tagging.

3. Description of the FST-based Computational Model of Babylonian

3.1 Previous and other relevant attempts

Kataja and Koskenniemi (1988) created the first computational description of the Akkadian morphology using the two-level formalism. They handled interdigitation of verbs by intersecting two regular lexicons, of which one described the root and its affixation, and the other the pattern formalisms. As the intersection approach was highly overgenerating, Kataja and Koskenniemi experimented with constraining the morpheme combinatorics by using unification-based features. This work was, however, stated to be still in progress when their paper was published.

Bamman and Andersson[5] (2012) is a finite-state description of Old Assyrian grammar purely implemented using *lexc* and *xfscript* formalisms (Beesley and Karttunen, 2003) and the Foma compiler (Hulden 2009). It is capable of analyzing several different parts of speech (with a lexicon comprising 255 verbal roots, 1918 nouns, 235 adjectives, 625 names, and 40 prepositions and adverbs) and it operates on transliteration. Automatic transcription works by duplicating or de-duplicating all vowels and consonants in the input string, and then by constraining the given options with the morphological analyzer. Some common logograms are treated by mapping them on their corresponding lemmas. The analyzer also has a guesser for unseen lexical items, which tries to give a correct POS-tag to unknown words. In this model, interdigitation is handled by describing verbs as sequences of morpheme slots, radicals and vowel classes; e.g. `_š_r_q_i/i`, and then by filling in the slots with morphemes; e.g. `i_ta_0_ā` → *ištarqā* 'they (f.) have stolen'. The analyzer was evaluated with a test corpus of 10,000 tokens. It returned a non-guessed analysis for 67.6% of the word form tokens (41.7% of unique word form types). Manual analysis of the 50 most frequent words in the corpus revealed that a correct morphological analysis was among the generated possibilities in 93.6% of the cases.

Contributions specifically to automatic analysis of Akkadian verb morphology are Barthélemy (1998), Macks (2002) and Sahala (2014). Barthélemy's analyzer/generator is based on Prolog Definite Clause Grammar (DCG) rules. The verb morphology is described in two levels. The first describes the paradigm for strong verbs and the second describes phonetic transformations. In the strong verb paradigm, verbal forms are split into nine smaller slices, of which each is described using a proper non-terminal (1. personal prefix, 2. stem prefix, 3. derivative infix, 4. R_1, 5. *t*-infix, 6. R_2 reduplication, 7. R_2 and its vocalization, 8. R_3, 9. personal suffix). The phonetic transformation level contains rewrite rules that produce, for instance, weak surface forms based on the strong verb paradigm, as well as phonological alternations. The system is abstract and does not include a dictionary of Akkadian roots. Macks' analyzer/generator is also written in Prolog by using DCG rules. The system is able to recognize and generate strong and singly weak verbs (in transcription) in G-, N-, Š- and D-stems, but does not handle morphophonemic alternation in the affixation. Similarly to Barthélemy 1998, Macks' description operates without any knowledge of the Akkadian lexicon. Thus, it does not contain information about valid roots or their vowel classes, which makes it highly overgenerating.

Sahala's approach to Akkadian verb morphology, *Babyparser*, is implemented in Python. The system analyses Akkadian (especially OB and SB) verbs from the transcription and syllabic transliteration by a reductive

[2] Unfortunately, the resources do not give a transparent description of their content in terms of token counts or language distribution.
[3] http://oracc.org/projectlist.html
[4] Counts are based on the Korp version of Oracc at http://korp.csc.fi/. The latest snapshot corresponds to Oracc as of May 2019.
[5] This report is authored by Bamman alone.

process, that first recursively strips out affixation and then compares the remainder with a series of regular expressions automatically generated from a root dictionary and a set of verbal stem templates. Mapping between transliteration and transcription is done by removing all non-alphabetic symbols, ignoring vowel lengths, and by simple heuristic rules (e.g. **Ca-a(-a)-aC** → **CajjaC**). The analyzer covers all stem derivations and verbal affixes for every verb class except for doubly weak verbs, of which description by using a reductive process was discovered to be too difficult and ambiguous due to the high degree of vowel contractions. Evaluation of 347 unique verbal forms in SB yielded a coverage of 86.1% (transliteration) and 89.0% (transcription). The wanted morphological analysis was among the results in 93.3% (transliteration) and 96.5% (transcription) of the cases. Our current FST implementation is based in many aspects on ideas in the *Babyparser*.

Currently, the most widely used analysis tool for Akkadian is a lemmatizer known as L2 by Steve Tinney (2018), which has been the main component for POS-tagging and lemmatizing Oracc. This tool is essentially an Emacs macro that works by mapping transliterated words with their corresponding lemmas, of which the annotator is supposed to pick the relevant one, or to add a new lemma manually if the word form is previously unseen.

3.2 Modeling Principles

Instead of modeling the interdigitation of the verbal stems dynamically, we chose to pre-generate the lexicon of the Akkadian verbal stems by using Python. In total the enumerated lexicon consists of 352k verbal stems (178k if vowel variation like *parris ~ parres* is excluded) for 1410 Babylonian lemmas. Such an enumeration was feasible, as we had already collected and classified Akkadian verbal roots and stems in our previous work (cf. Sahala 2014), and practical, as the minimization algorithms in finite-state compilers are quite efficient in identifying recurrent character strings, thus substantially reducing the final size of the transducer. This approach, chunking together a complex sequence several theoretical morphemic elements into a single unit, as well as having multiple stems associated with a single lemma, has previously been used successfully in the computational modeling of typologically comparable Dene languages, such as Tsuut'ina (Arppe et al., 2017).

The lemma-stem generator script combines root and template information stored in two files. First, an XML file that contains verbal roots, their conjugation class, vowel classes, vowel color, attestations in different dialects and time periods, and basic translations.[6] Second, it reads a description of verbal templates for different conjugation classes (1341 patterns in total for 42 conjugation classes, including irregular verbs). Following Akkadian text books, templates are represented by using symbols P-R-S for radicals R_1, R_2 and R_3. Positions for vowel-class dependent vowels are marked as V1 and V2 (e.g., P R V2 S represents strong G-preterite: *-prus-, -šbir-, -ndin-* etc.). Additional symbols are used when necessary. For instance, the position of the disappearing weak radical aleph is marked with a symbol X. This is useful for providing unambiguous contexts for rewrite rules that handle allomorphy, vowel contraction or lengthening and gemination at morpheme boundaries. Also, temporary symbols ♂ and ♀ are used to constrain suffixation of the middle weak verbs. Stems that may be followed by only a vocalic suffix are marked with the former (*i-dukk-ū* 'they kill') and the others with the latter (*i-dâk-ma, i-dâk-Ø* 'he kills'). This solution was more convenient than splitting the middle weak stems and all verbal suffixes into two groups in the lexicon (Table 2).

```
iialef-a-indicative ;   P V: a S ♀  ; G-Present
iialef-a-indicative ;   P â S ♀     ; G-Present
iialef-a-indicative ;   P V2 S S ♂  ; G-Present
```

Table 2: Patterns for middle-weak indicative G-present stems, e.g. /qīaš, qâš, qišš/, /dūak, dâk, dukk/.

We extracted lexicons of lemma-stem pairings for nouns and adjectives from Oracc.[7] The starting point are the Oracc lemmas, which we stemmed automatically by removing the nominative ending {u}. For final weak words, we replaced the contracted nominative ending with the symbol X to preserve the position of the weak radical. If the stem contained a final consonant cluster, we added an epenthetic vowel between them to produce construct forms: *parsu* → *par(a|i|u|e)s-*. This solution causes overgeneration, but it was the simplest one as the epenthetic vowel is often determined lexically and cannot be guessed. The first version of the noun and adjective lexicon, combined with morphology and rewrite rules, could recognize about 70% of the relevant word forms in Oracc. The remainder consisted of different lexically restricted spelling irregularities (e.g. *gikkigu ~ gigakku ~ giggigu*) and the syllabic alternation in the stems. In these, about 2000 cases, we extracted the transcriptions from Oracc and stemmed them by hand. For other parts of speech, stemming was done manually as the number of lexical entries was reasonably small (Table 3). Compound words are currently not supported due to their rarity.

[6] This root dictionary was originally manually composed by Sahala (2014) from Black et al. (2000) and Parpola & Whiting (2007).

[7] This data corresponds to Oracc's content in August 2018.

Lexicon	Entries	Transducer size
Nouns	35,354	380.9 kB
Adjectives	2755	
Verbs	352,115	3.8 MB
Adverbs	518	55.2 kB
Numerals	202	
Pronouns	289	
Particles	29	
Conjunctions	30	
Adpositions	293	
Interjections	23	
Proper nouns	12,230	443.4 kB
Total	**403,873**	**4.68 MB**

Table 3: Overview of the lexicon and POS coverage.

The relevant morphology and morphotactics for each part of speech is described in the corresponding lexicon. For example, labial personal prefixes are only permitted in front of D-, Š-, ŠD- and R-stems (and their *-t-* and *-tan-* derivations) by dividing the stems into labial and illabial groups. Circumfixes, which consist of prefixal person and suffixal number/gender parts, such as the third person plural {i...ū} (masc.) and {ī...ā} (fem.), are constrained by using flag diacritics. Interestingly, this morphological phenomenon as well as its practical computational implementation is very similar to that applied to the Algonquian languages, e.g. Plains Cree (Harrigan et al., 2017). The flag-diacritics present the manifested person prefix component of the circumfix, rather than the associated feature, which then determines which matching suffix components are allowable, and the morphological feature represented by the circumfix is determined upon encountering the suffix component.

Allomorphy, such as ventive /nim, am, m/, vetitive /ajj, ē/ and first person singular possessive /ja, ī, a/, which all have different realizations depending on the context, are described as special morphophonemic symbols. We use rewrite rules to change these symbols into correct surface representations; e.g. `[ AJJ ] -> ē || _ %< C`, which maps vetitive {ajj} to /ē/ before a consonant at the prefixal morpheme boundary marked with <.

Phonemic and morphophonemic alternation, as well as changes in orthography are expressed by using a composition of 26 rewrite rules (of which most are compositions of several rules). This handles previously mentioned morphophonemics, typical assimilations and dissimilations (*ibbi* → *imbi, inbi*), metatheses (*zitkar* → *tizkar*), variation in transcription conventions (*iprusma* ~ *iprus-ma*), spelling variation (*awīlu* ~ *amēlu*), aleph preservation/omittance (*ibanniū* ~ *ibannû* ~ *ibanni²ū*), morphotactic constraints in middle weak verbs (*idâk* ~ *idukkū*) and syncope (**taptarasī* → *taptarsī*). In the current version, these rules govern features attested in all stages of Babylonian and most of them are defined as optional. This makes it possible to analyze several stages of Babylonian with a single model, as well as Standard Babylonian, which often contains both, archaisms, and features from the contemporary spoken dialects.

In total, the compiled and minimized transducer is 6.2MB in size and consists of 143,244 states, 405,597 arcs and 1,867,800,170,342 paths.

4. Evaluation of the Model

We evaluated the morphological analyzer with all available transcribed Babylonian texts from Oracc, which we split into five sub-corpora based on their dialect. As we did not have a gold standard with complete morphological feature tagging, our goal was to produce the lemma and POS-tag that matched the corresponding annotation given in Oracc. Our hypothesis was that due to the complexity of Akkadian morphology, the analyzer is unlikely to produce a correct lemma and POS-tag without also producing a valid morphological analysis. We tested the hypothesis on a small scale by comparing 100 manually produced annotations with the BabyFST results. Of these, BabyFST did not give the wanted annotation only in five cases: three times due to Assyrianisms, once due to a missing feminine stem in the lexicon and once due to undefined spelling of the locative marker.

We measure the performance with three metrics for tokens and types in running text (Table 5 and 6). **Coverage** indicates the percentage of word forms that were accepted and analyzed by the transducer regardless of the analysis. **Recall** indicates the percentage of word forms that were given an annotation matching the analysis in Oracc. **Precision** measures the ratio of correct analyses (matches with Oracc) to the number of total analyses.

Dialect	OB	MB	SB	NB	LB
Word count	170,339	145,805	333,559	184,439	249,263
Coverage	95.97	95.82	97.37	95.97	97.06
Recall	91.01	90.33	93.65	91.02	93.11
Precision	41.89	40.62	41.15	40.26	41.87

Table 5: Evaluation by dialect (tokens).

Dialect	OB	MB	SB	NB	LB
Word count	22,132	17,861	39,950	23,226	20,667
Coverage	90.02	89.12	92.61	88.33	88.50
Recall	80.50	78.23	83.64	77.43	78.56
Precision	46.59	44.67	45.03	44.61	47.39

Table 6: Evaluation by dialect (types).

Dialect	OB	MB	SB	NB	LB
ADJ	77.84	76.99	66.21	76.24	79.14
ADV	75.29	73.90	85.28	78.33	76.94
N	93.00	92.08	95.45	93.14	94.04
PN	83.37	83.13	88.97	84.06	84.31
PRON	89.71	86.35	94.08	90.35	91.66
V	89.74	88.56	91.05	87.50	87.00
MISC	91.20	88.27	89.51	89.26	90.77

Table 7: Recall by POS (tokens). Rare POS are collapsed under MISC.

Low recall (Tables 5, 6, 7) is partly explained by differing lemmatization conventions, spelling variation, Oracc lemmatization errors and over-analysis of certain word forms. We examined 150 random unique mismatches with Oracc's annotation, of which 68 were true errors. Of these errors, 44 were caused by missing lexical items in our lexicon, or otherwise incorrect analyses, such as errors in verbal stems. The rest of the errors were caused by undefined Assyrianisms, incorrectly defined infinitives, and undefined loss of weak radicals.

The remaining 82 mismatches actually contained correct analyses that did not match the Oracc lemma due to lack of normalization. Of these mismatches, 41 were a result of over-analysis. For example, where Oracc has a lemma *kullu*+V, the analyzer returns *kâlu*+V+D+Inf, which is essentially the same form, but it is just represented in a more atomistic way. Similarly, several adverbs formed with {iš} and feminine nouns formed with {at} are broken up into smaller components, while the Oracc lemmatization displays them as lexicalized units (*ūmu*+N+Adv ~ *ūmiš*+AV; *qerbu*+N=Fem ~ *qerbetu*+N). In 29 of the cases a correct analysis mismatched due to spelling variation in the lemma (*melammu*+N ~ *melemmu*+N), or because the lemma was represented in Oracc in its Assyrian form (*walādu*+V ~ *ulādu*+V). The rest of the mismatches were caused by lemmatization errors in Oracc (e.g. *gašri*+N instead of *gašru*+N). In these cases, BabyFST returned the correct lemma. Taking this into account, up to half of the missing recall is caused by a lack of normalization between BabyFST and Oracc.

4.1 Ambiguity

Considering the whole corpus on the token level, the average number of given analyses (correct or not) for each POS is as follows: adjectives (6.38), adpositions (3.44), adverbs (5.75), conjunctions (1.52), interjections (3.72), proper nouns (4.42), nouns (4.12), numerals (4.40), pronouns (3.89) and verbs (3.28).

The confusion matrix (Table 8) represents the distribution of POS tags (horizontal) given to an input (vertical). For example, nouns receive on average 4.12 analyses. Of these analyses on average 56.66% are tagged as nouns, 34.89% as verbs, 5.57% as adjectives and 2.88% as proper nouns or something else.

I/O	ADJ	ADV	N	PN	PRON	V	MISC	Σ
ADJ	**21.89**	0.04	37.68	0.11	0.01	39.07	1.20	100.00
ADV	17.09	**17.68**	31.06	0.37	0.14	26.8	6.87	100.00
N	5.57	0.10	**56.66**	0.91	0.26	34.89	1.60	100.00
PN	0.24	0.00	1.10	**96.16**	0.00	1.49	1.00	100.00
PRON	0.20	0.12	30.35	0.56	**26.75**	10.77	31.25	100.00
V	3.49	0.01	12.44	0.50	0.15	**82.05**	1.36	100.00
MISC	1.78	0.40	42.09	1.07	5.34	11.13	**38.18**	100.00

Table 8: POS confusion matrix.

4.2 Unweighted Model

We set the baseline for disambiguation by analyzing the Standard Babylonian corpus and sorting the results by lemma frequency. We calculate recall in three settings: taking into account the morphological analyses with only (1) the most frequent, (2) two most frequent, and (3) three most frequent lemmas (Table 9).

	Recall @ 1	Recall @ 2	Recall @ 3
Tokens	78.12	90.19	92.65
Types	63.89	77.87	81.70

Table 9: Baseline recall for tokens/types.

4.3 Weighted Model

As stated in Section 4.1, the average number of possible morphological analyses for Akkadian words is relatively high (Fig. 1). In this section, we present a simple weighting algorithm for finite-state analyzers. The algorithm utilizes a manually disambiguated list of training examples consisting of input word forms and morphological analyses. The aim is that each analysis in the training data receives a higher likelihood than other plausible analyses for the relevant input word forms. Because of shared states and transitions in finite-state networks, this behavior is expected to generalize to other word forms as well. We provide no theoretical guarantees for the weighting algorithm, since success depends on network topology, and the experimental results presented in Section 4.3.2 show that having *only* hand-validated lemma and POS information but *not* validated complete morphological analyses in the training of a weighted finite-state transducer provides moderate success in ranking the most likely analyses highest.

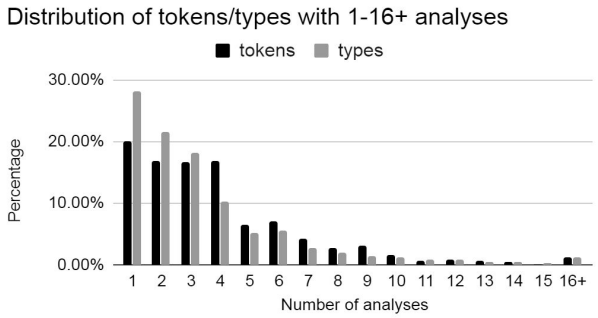

Figure 1. Token/type ambiguity of BabyFST analyses.

4.3.1 Weighting Algorithm

Our weighting algorithm is based on traversing the states and transitions in a deterministic finite-state transducer using aligned string pairs given as training examples. The algorithm loops through all examples in the training set and adds counts of state-to-state transitions in the unweighted finite-state transducer. It then normalizes these counts into probability distributions in each state. Below, we give a more formal explanation of the algorithm.

Following Allauzen et al. (2007), we view finite-state transducers as finite-state acceptors of strings consisting of symbol-pairs *x:y,* where *x* and *y* belong to finite input and output alphabets, respectively. As a special case, either *x* or *y* can be the empty symbol *ε*.[8] We denote transitions in *T* as 5-tuples *(s, t, x, y, w)*, where *s* and *t* denote the source and target state, respectively; *x:y* is a symbol pair and w_t is a real-valued weight. A subset of the states in *T* are final states. Each final state *s* has a real-valued final weight w_s. If *T* is a transducer, it can be determinized as a finite-state acceptor. Below, we will assume that all transducers are deterministic in this sense.

In order to assign weights to the transitions of transducer *T*, we need a set $P = \{p_1,...,p_m\}$ or strings $p = x_1{:}y_1, \ ..., \ x_n{:}y_n$, where each string *p* is accepted by transducer *T*. We start by associating each state *Q* of transducer *T* with a transition counter C_Q , which maps symbol-pairs *x:y* to counts $C_Q(x{:}y)$, and a finality count f_Q. If string $p=x_1{:}y_1, \ ..., \ x_n{:}y_n$ is accepted by transducer *T*, then each symbol-pair $x_i{:}y_i$ corresponds to a unique accepting state Q_i because transducer *T* is deterministic. Moreover, each string *p* is associated with a unique final state F_p.

Let

$$C_Q(x{:}y) = \alpha + \sum_p \sum_{xi:yi \in p} [Q_i = Q]$$

$$f_Q = \alpha + \sum_p [F_p = Q]$$

Here indicator $[a = b]$ evaluates to 1, if $a = b$, and 0, otherwise. In the equations above, α is a smoothing term which we set to 1 in all experiments.

For a transition *(s, t, x, y, w)* in *T*, we now define the transition weight *w* as:

$$w = C_s(x{:}y) \Big/ (f_s + \sum_{z:u} C_s(z{:}u))$$

We define the final weight w_S of a final states *S* as:

$$w_s = f_s \Big/ (f_s + \sum_{z:u} C_s(z{:}u))$$

4.3.2 Experiments

Currently, the human-validated corpora available for Akkadian, while substantial, only indicate the lemma and POS tag of each word form token, rather than a full morphological feature analysis disambiguated in context, which would normally be necessary for training a weighted finite-state transducer. Thus, we set forth to explore whether we could nevertheless make use of such validated data in combination with our unweighted finite-state transducer to prune the extent of ambiguous analyses. Therefore, we first ran through the 333,560 tokens in the Standard Babylonian subcorpus of Oracc through our finite-state transducer to retrieve full morphological analyses, which are potentially ambiguous as noted above in Section 4.1. Second, we used the validated lemmas and part-of-speech tags provided in the Oracc subcorpus per each token to select only those full morphological analyses which matched the validated coding. As a result, the remaining ambiguity amounted on average to 1.57 morphological analyses per token, as a majority (57.4%) of the tokens had a single, unambiguous full morphological analysis (Table 10).

%	Cumul. %	Tokens	No. of analyses
57.4	57.4	191,502	1
20.1	77.5	66,889	2
12.2	89.7	40,799	3
1.0	90.7	3,230	4
0.3	90.9	886	5
9.1	100	30,254	6-93

Table 10: Ambiguity of morphological analysis remaining after pruning with validated lemma and POS tags, which is used as the training data in weighting.

Third, we used these pairings of tokens and their lemma-disambiguated full morphological analyses to weight our Akkadian finite-state transducer according to the algorithm described in Section 4.3.1. Throughout this, we used HFST - Helsinki Finite-State Technology

[8] Note that *ε:ε* is not a valid pair.

compiler (Lindén & al., 2011) which is capable of compiling and running weighted transducers.

Fourth, we re-analyzed all the tokens from the training corpus with the weighted finite-state transducer in order to evaluate its baseline performance. Since the only validated linguistic information we had for these tokens were their lemma and part-of-speech, we focused on the rank of the best-weighted full morphological analysis that corresponded to the validated POS tag.

The results in Table 11 should be compared with Fig. 1 which shows that in the not yet disambiguated data only 20% of the tokens had a single reading. After weighting all the readings, 55% of the tokens had the correct analysis (in terms of matching a validated lemma and part-of-speech tag) ranked as the first analysis[9], while the average rank of the (highest) ranked "correct" analysis was 2.03, and when considering the three top-most ranked analyses together we exceeded (80.9% match with the validated lemma and part-of-speech coding) what could be achieved with a baseline heuristic of using lemma-frequency alone for selecting the most likely analysis (with a recall of 78.12%).[10]

%	Cumul. %	Tokens	Rank
55	55	183,310	1
16.6	71.6	55,487	2
9.4	80.9	31,204	3
5.5	86.4	18,353	4
2.4	88.8	7,903	5
4.9	93.7	16,242	6+
6.3	100	21061	N/A

Table 11: Rank of morphological analysis matching with validated lemma and POS tags.

Nevertheless, we may conclude that due to the inherent ambiguity of Akkadian, validated lemma+POS information simply is neither enough for selecting the correct complete morphological analyses, nor for adequately weighting a finite-state transducer for Akkadian.

[9] The results did not essentially improve if we used as training data in the weighting of the finite-state transducer only those tokens for which the Oracc lemma and POS tag accepted a single, unambiguous complete morphological analysis.

[10] Note that the discrepancy between the 55.0% proportion of first-ranked analyses for tokens with the weighted finite-state transducer vs. the 57.4% proportion for tokens receiving a single morphological analysis using the Oracc lemma and POS tags, as *post hoc* disambiguation after the application of the unweighted finite-state transducer, is due to the fact that the hand-validated information may prune some unweighted analyses that the weighted finite-state transducer may still output and assign a better weight than other analyses matching the Oracc tags.

5. Further Work and Future Directions

In order to properly evaluate the morphological tagging, a completely morphologically analyzed and disambiguated gold standard is required. Fortunately, a manually analyzed corpus of royal inscriptions written in Standard Babylonian is currently being annotated by the Akkadian Treebanking project at the University of Helsinki.[11] The gold standard will provide accurate data for weighting the transducer and will also allow us to evaluate methods for disambiguating the morphological analyses in context.

We aim to include support for analyzing transliterated input. The support for transliterated text will make it possible to lemmatize, POS-tag and morphologically analyze transliterated but not yet transcribed large text corpora such as CDLI. Being able to operate directly on transliterated input can later be combined with OCR in order to process scanned pictures of cuneiform tablets.

6. Conclusions

BabyFST is a unified morphological model for different stages of Babylonian dialects (1900 BCE – 100 CE), and currently the most comprehensive morphological analyzer for Akkadian. It operates on transcriptions and is able to achieve a coverage up to 95.82–97.37% with a recall of 90.33–93.65% (on tokens) depending on the dialect. Up to half of the missing recall is caused by normalization issues rather than an incomplete definition of the lexicon or lacking morphotactic descriptions. We still do not have a robust way to disambiguate the results due to the lack of a morphologically tagged gold standard. Nevertheless, in combination with *post hoc* disambiguation using the lemma and POS tags in Oracc, BabyFST can already now be used to provide complete, unambiguous morphological analyses for 57.4% (191,502) of the tokens in the Standard Babylonian subcorpus and, using only a weighted transducer, the correct analysis can be provided among the top-3 for 80.9% of the input tokens. Our next goal is to tackle morphological disambiguation and transliteration in context applying the analyzer to available text corpora.

Acknowledgements

We acknowledge funding from the Academy of Finland for the Centre of Excellence in Ancient Near Eastern Empires and from the University of Helsinki for the Deep Learning and Semantic Domains in Akkadian Texts Project (PI Saana Svärd for both).

[11] PI of the project is Niek Veldhuis (UC Berkeley / University of Helsinki) and the annotations are being made by Mikko Luukko (University of Helsinki).

Bibliographical References

Allauzen, C.; Riley, M.; Schalkwyk, J.; Skut, W. & Mohri, M. (2007). *OpenFst: A General and Efficient Weighted Finite-State Transducer Library*. In Conference on Implementation and Application of Automata.

Arppe, A.; Cox, C.; Hulden, M.; Lachler, J.; Moshagen, S. N.; Silfverberg, M. & Trosterud, T. (2017). Computational Modeling of Verbs in Dene Languages: The Case of Tsuut'ina. In Jaker, A. (ed.), *Working Papers in Athabaskan (Dene) Languages*, 51-69. Alaska Native Language Center Working Papers 13. Fairbanks: Alaska Native Language Center.

Bamman, D. (2012). 11-712 NLP Lab Report. https://github.com/dbamman/akkadian-morph-analyzer/blob/master/doc/latex/whitepaper.pdf

Barthelémy, F. (1998). A Morphological Analyzer for Akkadian Verbal Forms with a Model of Phonetic Transformations. In *Computational Approaches on Semitic Languages*.

Beesley, K, and Karttunen, L. (2003). *Finite State Morphology*. Palo Alto, California: CSLI Publications.

Black, J. A., George, A. R. and Postgate J. N. (2000). *A Concise Dictionary of Akkadian*. Wiesbaden: Harrassowitz Verlag.

Buccellati, G. (1996). *A Structural Grammar of Babylonian*. Wiesbaden: Harrassowitz Verlag.

Harrigan, A.; Schmirler, K.; Arppe, A.; Antonsen, L.; Moshagen, S. N.; Trosterud, T. & Wolvengrey, A. (2017). Learning from the Computational Modeling of Plains Cree Verbs. *Morphology*, 27(4), 565–598.

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 (EACL): Demonstrations Session, 29–32. Athens: Association for Computational Linguistics.

Kataja, L. and Koskenniemi, K. (1988). *Finite-State Description of Semitic Morphology: A Case Study of Ancient Akkadian.* Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics.

Kouwenberg, N. J. C. (2010). *The Akkadian Verb and its Semitic Background.* Languages of the Ancient Near East 2. Winona Lake: Eisenbrauns.

Kouwenberg, N. J. C. (2011). Akkadian in General. In *Semitic Languages. An International Handbook.* S. Weninger, G. Khan, M. P. Streck & J. C. E. Watson (Eds.), De Gruyter Mouton, pp. 330–340.

Lindén, K., Axelson, E., Hardwick, S., Pirinen, T. A. & Silfverberg, M. (2011). HFST-Framework for Compiling and Applying Morphologies. In *Systems and Frameworks for Computational Morphology*. Mahlow, C. & Piotrowski, M. (eds.). Springer-Verlag, pp. 67-85 19 p. (Communications in Computer and Information Science; vol. 100).

Macks, A. (2002). *Parsing Akkadian Verbs with Prolog.* SEMITIC '02 Proceedings of the ACL-02 workshop on Computational approaches to Semitic languages. pp 1–6.

Parpola, S. and Whiting, R. M. (2007). *Assyrian-English-Assyrian Dictionary*. Helsinki: The Neo-Assyrian Text Corpus Project.

Reiner, E. (1966). *A Linguistic Analysis of Akkadian.* Janua Linguarum, Series Practica 21. The Hague: Mouton.

Sahala, A. (2014). *Babylonian diskontinuatiivisen morfologian ohjelmallinen jäsentäminen*. MA Thesis, University of Helsinki.

Soden, von, W. (1995). *Grundriss der akkadischen Grammatik*, 3., ergänzte Auflage (Analecta Orientalia 33/47) Roma: Pontificio Istituto Biblico.

Streck, P. M. (2011). Babylonian and Assyrian. In *Semitic Languages. An International Handbook.* S. Weninger, G. Khan, M. P. Streck & J. C. E. Watson (Eds.), De Gruyter Mouton, pp. 359–398.

Tinney, S. (2018). *L2: How It Works.* Oracc: The Open Richly Annotated Cuneiform Corpus. http://oracc.org/doc/help/lemmatising/howl2works/

Language Resource References

ARCHIBAB. (2008). Archives babyloniennes (XX^{e}-XVII^{e} siècles) http://www.archibab.fr/

CDLI. (2000). The Cuneiform Digital Library Initiative. https://cdli.ucla.edu/

Oracc. (2014). The Open Richly Annotated Cuneiform Corpus. http://oracc.org

SAA(o). (1986). State Archives of Assyria (online). http://www.helsinki.fi/science/saa/

SEAL. (2008). Sources for Early Akkadian Literature. https://www.seal.uni-leipzig.de/