The main sources of information regarding ancient Mesopotamian history and culture are clay cuneiform tablets. Many of these tablets are damaged, leading to missing information. Currently, the missing text is manually reconstructed by experts. We investigate the possibility of assisting scholars, by modeling the language using recurrent neural networks and automatically completing the breaks in ancient Akkadian texts from Achaemenid period Babylonian.

Choosing Digital Akkadian Corpora. One challenge in designing an automated text-completion tool is the limited data available to quantify the uncertainty in each restoration. The problem is even harder in the case of damaged texts that belong to a well-known genre-archival documents such as contracts or deeds, for example, but do not exist in duplicate form. In such cases, one must resort to predicting the content of damaged passages on the basis of conventions identified by studying intact examples of the genre.

One possible way to ameliorate these difficulties is to design an automatic process that can aid human experts engaged in the task of restoring damaged texts. In this article, we investigate an approach to automatically completing broken passages in Late Babylonian archival texts that relies on modern machine learning methods, specifically recurrent neural networks (RNNs) (6).

Due to the limited number of digitized cuneiform texts available at present, it is uncertain whether such data-driven methods would yield plausible restorations in all types of texts, but we hypothesized that for genres with highly structured syntax—such as legal, economic, and administrative Late Babylonian texts—these models should work well, as we will demonstrate here.

Furthermore, we are developing an online tool, called Atrahasis (https://babilonian.herokuapp.com/) to make our work available to a wide scholarly community. Our source code is available at GitHub (https://github.com/DHALab/Atrahasis).

Significance

The documentary sources for the political, economic, and social history of ancient Mesopotamia constitute hundreds of thousands of clay tablets inscribed in the cuneiform script. Most tablets are damaged, leaving gaps in the texts written on them, and the missing portions must be restored by experts. This paper uses available digitized texts for training advanced machine-learning algorithms to restore daily economic and administrative documents from the Persian empire (sixth to fourth centuries BCE). As the amount of digitized texts grows, the model can be trained to restore damaged texts belonging to other genres, such as scientific or literary texts. Therefore, this is a first step for a large-scale reconstruction of a lost ancient heritage.

Author contributions: E.F. and Sh.G. designed research; E.F., Y.L., E.A., and Sh.G. performed research; E.F. and Sh.G. contributed new reagents/analytic tools; E.F., Y.L., E.A., and Sh.G. analyzed data; and E.F. and Sh.G. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission. E.P. is a guest editor invited by the Editorial Board. Published under the PNAS license.

1 E.F. and Sh.G. contributed equally to this work.
2 To whom correspondence may be addressed. Email: ethan.fetaya@biu.ac.il or shaigo@ariel.ac.il.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2003794117/-/DCSupplemental.
in digital form. For similar unsupervised language modeling tasks in English, for example, one can collect practically endless amounts of texts online, and the main limitation is the computational challenge of storing and processing large quantities of data (7). For cuneiform texts this is not the case. Automatic optical character recognition cannot be used to reliably identify cuneiform signs, neither in their two-dimensional (2D) representations (hand copies) nor in three-dimensional (3D) scans of actual clay tablets (8–12). Various visual recognition algorithms are being applied to cuneiform, but the results are yet in their infancy (13–16). Therefore, one has to rely on a limited corpus of manually transcribed texts. Although Akkadian cuneiform texts span more than two millennia and the genres available for study are heterogeneous, for many periods, only a limited amount of digital text is available to train the learning algorithm.

Three temporally and geographically defined corpora in particular are well represented in the digital transcriptions available today: the Old Babylonian (approximately 1900 to 1600 BCE; see ARCHIBAB website: http://www.archibab.fr/), Neo-Assyrian (approximately 1000 to 600 BCE; see State Archives of Assyria online website: http://oracc.org/saao/), and Neo- and Late Babylonian (approximately 650 BCE to 100 CE). Our choice of texts, however, was governed by a corpus-based approach so that we might exercise greater control over the diversity of text genres and phrasing. Therefore, we have decided to gather 1,400 Late Babylonian transcribed texts from Achaemenid period Babylonia (539 to 331 BCE) in Hypertext
Algorithmic Background

In this section, we will give a very brief introduction to techniques for modeling language using RNNs (for a more detailed account, see ref. 6). We can view language as a series of discrete tokens $x_1, \ldots, x_T$, and our goal is to fit a probabilistic model for such sequences; i.e., we wish to find a parametric model that learns the distribution $p(x_1, \ldots, x_T)$ from samples. The first step is to use an autoregressive model, i.e., use the factorization $p(x_1, \ldots, x_T) = \prod_{t=1}^{T} p(x_t | x_1, \ldots, x_{t-1})$. What this means is that we can reduce the problem of modeling a full sentence to predicting the next token in a text on the basis of the tokens that precede it.

In RNNs, this autoregressive model $p(x_t | x_1, \ldots, x_{t-1})$ is fitted using a hidden memory. Given the previous hidden memory $h_{t-1}$, the network first updates the memory based on the new input $x_{t-1}$ and then uses the updated memory to predict the next token and passes the updated memory to the next step. More formally:

$$h_{t-1} = \tanh(W_{ih} h_{t-2} + W_{io} x_{t-1} + b_i),$$

$$prob_t = \text{softmax}(W_{oh} h_{t-1} + b_o),$$

where $x_{t-1}$ is a one-hot representation of the input token, $W$ indicates linear mappings, and $prob_t$ is the vector of probabilities for each possible next token. This parametric model is trained by maximizing the training log-likelihood to produce the output model. While simple and effective, due to vanishing gradients simple RNNs have difficulties in modeling long time dependencies, i.e., situations in which the probability of the next token depends on information seen many steps before. To solve this issue, various modifications that introduce a gating mechanism, such as long short-term memory (LSTM), have been proposed (22).

As a baseline model for comparison we trained an n-gram model. The n-gram model is a model that assigns a probability
to each token based on how frequently the sequence of the last \( n - 1 \) tokens in the training set ended in that token. The main limitation of n-gram models is that for small \( n \), the context used for prediction is very small, while for large \( n \), most test sequences of that length are never seen in the training set. We used a 2-gram model, i.e., each word is predicted according to the frequency with which it appeared after the previous one.

### Results

In order to generate our datasets, we collected transliterated texts from the Achemenet website, based on data prepared by F. Joannès and coworkers in the framework of the Achemenet Program (National Center for Scientific Research [CNRS], Nanterre, France) (http://www.achemenet.com/fr/tree/\ sources-textuelles/textes-par-langues-et-ecritures/babylonien/). We designed a tokenization method for Akkadian transliterations, as detailed in Materials and Methods. We trained a LSTM recurrent network and a n-gram baseline model on this dataset (see Datasets S1–S3 for model and training details).

Results for both models are in Table 1. Loss refers to mean negative log-likelihood and perplexity is two to the power of the entropy (in both cases, lower is better).

As expected, the RNN greatly outperforms the n-gram baseline, and despite the limitations of the dataset, it does not suffer from severe over-fitting.

### Completing Random Missing Tokens

In order to evaluate our models’ ability to complete missing tokens, we took random sentences from the test corpus, removed the middle token and tried to predict it using the rest of the sentence. Our model returns a ranking of probable tokens and we report the mean reciprocal rank (MRR). The MRR is the average over the dataset of the reciprocal of the predicted rank of the correct token. It is a very common and useful measure for information retrieval as it is highly biased toward the top ranks, which is what the user is mostly interested in. We also evaluate the “hit@k,” which measures the percentage of sentences where the correct completion is in the top \( k \) suggestions. For evaluation, we used all test sentences 10 or more tokens in length that contain no breaks, which yielded a total of 520 sentences.

We compared two variations of our model, one that finds the optimal completion based only on the tokens that precede the missing token, denoted “LSTM (start),” and one that takes the full sentence into account, denoted “LSTM (full).” As the “LSTM (full)” model needs to run separately for each candidate the missing token, denoted “LSTM (start),” and one that takes into account both the previous and the next token denoted “2-Gram (full).” While this is a relatively weak model, we found it to work surprisingly well, although it was still significantly inferior to the LSTM model in the accuracy (Hit@1) metric.

To further investigate our model’s ability to complete various numbers of missing tokens in various locations, we removed up to three tokens in random locations. We ranked possible completions using our model and beam search and show the results in Table 3.

It is clear from the results in Tables 2 and 3 that our algorithm can be of great help in completing a missing token, with an almost 85% chance of completing the token correctly and a 94% chance of including the correct token in the top 10 suggestions. However, as expected, the task becomes much harder and performance is degraded when more tokens are missing. We note that even with two or three missing tokens, however, the model is still useful as the correct completion is present in the top 1 (two missing) or 10 (three missing) completions almost half of the time.

### Designed Completion Test

We designed another experiment in order to evaluate our completion algorithm and understand its strengths and weaknesses. We generated a set of 52 multiple choice questions in which the model is presented with a sentence missing one word and four possible completions, and the goal was to select the correct one. Of the three wrong answers, the first was designed to be wrong semantically, the second wrong syntactically, and the third both. This allowed us to track the types of mistakes the algorithm makes. The assumption is that the learning algorithm would be more likely than a human to make semantic mistakes but should be better than a nonexpert in grammar. If this is the case, then the effectiveness of our approach as a way to assist humans should rise, as the strengths of human and machine complement each other.

When we used our model to rank four possible restorations for each of the missing words in the 52 random sentences, it achieved 88.5% accuracy in selecting the one with the highest likelihood (see Dataset S4 for the complete list of questions and answers). Looking at the six failed completions—questions 18, 26, 32, 35, 45, and 50—we see that four are semantically incorrect, one is syntactically incorrect, and one is both, which agrees with our hypothesis.

### Discussion

Further study of the different restorations of the designed completion test, taking into account the full ranking of the answers, results in some interesting patterns. This qualitative analysis considers four categories for the answer ranking: 1) correct syntax (i.e., sentence structure), 2) correct semantic identification, 3) poor syntax, and 4) poor semantic identification (see Dataset S5 for the full data analysis).

The majority of the restorations, 44 cases, shows that the algorithm best identifies correct sentence structure (category 1: questions 1 to 23, 25 to 27, 29 to 35, 38 to 42, 46, and 48 to 52). Put more accurately, this means correct syntactic sequences of parts of speech based on the statistical frequency of smaller syntactic structures. A total of 34 restorations show correct semantic identification of noun class as well as related verbs in the answer ranking (category 2: questions 1, 3, 6 to 13, 16 to 18, 22, 24 to 27, 29, 31 to 33, 35 to 40, 42, 46 to 49, and 51). In fact, a large subset of these cases, 30 questions, shares also identification of correct semantic structure (i.e., both categories 1 and 2).

Good semantic identification probably derives from paradigmatic relationships between certain classes of words. For example, five cases possibly correctly identified usage of verbal forms based on their context (e.g., in direct speech; questions 3, 6, 9, 10, 15).

### Table 1. Loss and perplexity while training the model on Achemenet dataset

<table>
<thead>
<tr>
<th>Training loss</th>
<th>Training perplexity</th>
<th>Test loss</th>
<th>Test perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-Gram</td>
<td>3.26</td>
<td>9.55</td>
<td>3.54</td>
</tr>
<tr>
<td>LSTM</td>
<td>1.41</td>
<td>4.10</td>
<td>1.62</td>
</tr>
</tbody>
</table>

### Table 2. Completing missing fifth token in sentences

<table>
<thead>
<tr>
<th>Sth index</th>
<th>MRR</th>
<th>Hit@1</th>
<th>Hit@5</th>
<th>Hit@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-Gram (start)</td>
<td>0.64</td>
<td>52.0%</td>
<td>78.2%</td>
<td>83.6%</td>
</tr>
<tr>
<td>LSTM (start)</td>
<td>0.75</td>
<td>66.1%</td>
<td>86.9%</td>
<td>91.9%</td>
</tr>
<tr>
<td>2-Gram (full)</td>
<td>0.80</td>
<td>74.8%</td>
<td>85.5%</td>
<td>90.6%</td>
</tr>
<tr>
<td>LSTM (full)</td>
<td>0.89</td>
<td>85.4%</td>
<td>93.2%</td>
<td>94.6%</td>
</tr>
</tbody>
</table>
Table 3. Completing various number of tokens

<table>
<thead>
<tr>
<th>Missing tokens</th>
<th>MRR</th>
<th>Hit@1</th>
<th>Hit@5</th>
<th>Hit@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>0.86</td>
<td>81.5%</td>
<td>92.5%</td>
<td>94.4%</td>
</tr>
<tr>
<td>Two</td>
<td>0.55</td>
<td>47.8%</td>
<td>62.2%</td>
<td>70.0%</td>
</tr>
<tr>
<td>Three</td>
<td>0.30</td>
<td>24.3%</td>
<td>37.2%</td>
<td>42.6%</td>
</tr>
</tbody>
</table>

18, 25, and 26). Take, for example, question 3: NAME ana NAME já NAME ana. Take, for example, question 3: NAME ana NAME já NAME ana. The example not only shows a correct identification of sentence structure but also a recognition of the relationship between two different forms of the verb qabû, “to speak.” It does not necessarily reflect an understanding of verbal root form, however; the model’s success probably reflects the statistical frequency of qabû in this context and the model’s recognition of its similarity to liqbuṭ. This statistical inference emerges more clearly in one of the mistakes made by the model in question 32, where it does not differentiate properly the grammatical person of the verb nadānu, “to give, pay” (taaddinu vs. ināmdin).

The level of the model’s semantic knowledge becomes apparent with regard to noun class; 14 questions show possible correct identifications of countable nouns (questions 1, 9, 17, 29, and 33), names of professions (questions 11, 38, 39, and 46), temporal designations (questions 12, 36, and 41), gender (question 31), and even a contextual formulaic legal clause (the so-called elat-clause; question 51). Four cases show correct identification of prepositions, particle use, or pronouns (questions 7, 34, 42, and 51). The choice in question 7, between the related prepositions ina and ana, makes it clear that these choices are again based on frequency in specific contexts. Moreover, a purely statistical grasp of parts of speech seems to be a decisive factor in at least eight cases of restoration, which are best identified in the analysis as those fitting category 3: poor syntax (questions 24, 28, 36 to 45, 43 to 47). Such statistical inference achieves surprisingly good results—e.g., preferring kurkur over Location after lugal (question 37)—with only one restoration ended up being erroneous (question 45). However, it is clear, by comparing this group to other restorations in the designed completion test, that statistical inference is not a consistently reliable method for choosing completions. For example, it can interfere with contextual identification of the correct restoration—by preferring ina over ina šu before question (question 35).

The model does not seem to identify alternate logographic and phonetic writings of the same words, e.g., Sum. da = Akk. itti, or Sum. im.dub = Akk. nappi (questions 14 and 19). It obviously lacks enough examples of such interchangeability in the studied corpus, since it does identify when different logograms have similar usage (e.g., a and dumu), both meaning “son” or “descendant,” are the top answers in question 40). Further confusion can occur when the model detects a similarity between the answer and another word close by in the sentence, either a noun or a verb. Especially problematic are cases when there are very few similar sentences to train on, and so the algorithm makes an “educated” guess resulting in a mistake (for example, question 45).

Conclusion

Our model—as far as can be judged by this experiment—is, as expected, good in teasing out sentence structures. However, it was also surprisingly better than we expected in making semantic identifications on the basis of context-based statistical inference (rather than finding underlying grammatical rules and morphology). In order to greatly reduce the number of false identifications based on the statistical frequency of contextual semantic
relationships, much more training material will be needed. Nevertheless, we have demonstrated that even without access to large amounts of data, we can successfully train LSTM models and use them to complete missing words. In our completion test, we show good results that, while not sufficient for fully automatic completion, prove that the model can be an invaluable tool in helping scholars with text restoration.

Our results with the Late Babylonian corpus are significant because most entry-level scholars or other interested historians and social scientists who focus on the large first-millennium BCE Babylonian archives cannot acquire the very specific knowledge and expertise to understand underlying political, social, or historical structures without reading through hundreds of texts. For this reason, we are in the process of incorporating our model into an online tool, called Atrahasis. It will be of immense help to scholars in the historical sciences, allowing them to overcome the high entry barrier to restoring fragmentary Akkadian texts. Initially, the model will achieve the most success with structured archival documents, but as the dataset grows, one can train the model on more genres, such as scientific or literary texts. Both access to the primary sources in their original state and the ability to restore broken passages are equally necessary for understanding Akkadian corpora on a macroscale.

Related Work

The task of text modeling and its applications in tasks such as text restoration, lies in the intersection of Natural Language Processing (NLP) and Computer Linguistics. The Computational Linguistics models are predominantly rule based, while current NLP models are predominantly statistical and use machine learning. Currently, machine-learning methods achieve state of the art performance on most NLP challenges. In most modern languages, basic text modeling tools can identify spelling errors such as missing, added, transposed, or wrong letters (23–26). These tasks can become more challenging when the language in question has a richer morphology, like Arabic, for example (27), or limited digital corpora (28), as in the case of ancient Near Eastern languages (29, 30).

One approach is to first parse the original text and then use this as an input for further tasks. To this end, many studies use rule-based models derived from grammar (31), finite-state machines (32), or lookup in a machine readable dictionary (33) or employ statistical models such as clustering algorithms (k-nearest neighbors or kMeans) (34). Most work on rule-based models has been done in Akkadian (see literature cited in ref. 35). The most recent study dedicated to Akkadian word segmentation used a combination of rule-based, dictionary-based, and statistical algorithms, with best results in dictionary-based models (60 to 80%) (35). Because its algorithms were fitted for East Asian languages like Chinese and Japanese, we concluded that a specific NLP model for Akkadian should be designed. Sumerian, with its simpler syntax, is in the center of the Machine Translation and Automated Analysis of Cuneiform Languages project, which employed dictionary- and rule-based models for annotation of Sumerian (36, 37). A similar study on Hitite designed rule-based models derived from grammar (38). Statistical/machine-learning models achieve better results overall, if they are tailored to the problem at hand. Our project, the Babylonian Engine, recently achieved state-of-the-art results for Akkadian prediction of sign and word transliteration and segmentation using NLP and machine-learning models on Unicode cuneiform: up to 97% using a BiLSTM Neural Network algorithm, see the web-tool Akkademia (https://babylonian.herokuapp.com/). A similar task of automatic phonological transcription of Akkadian (usually termed normalization) based on ORACC material has recently achieved promising results (39).

For the specific task of text restoration and prediction, statistical n-gram Language Models (LMs, e.g., bigrams, trigrams, etc.) are now widely used in NLP tools, including those designed for modern Indian languages (40). A bigram model was successfully applied to a hidden Markov model to restore missing or damaged sign sequences in the ancient undeciphered Indus script (41, 42). However, neural network LMs can perform better in developing meaningful patterns of representations of words and the contexts around them. When these “embeddings” are learned from unsupervised large corpora, they can be transferred to various tasks, retaining a boost in performance (43).

Specifically, RNN language models, like the one employed in this study, have shown success in encoding both semantic and orthographic data in languages of varying levels of morphological complexity (44). The best results in solving the restoration task, so far, have been achieved in studies of machine-reading comprehension, specifically of the cloze-style, in which both the level of character and word are identified (45). In the field of ancient languages, we know of only one other study that used an algorithm with neural network architecture to recover missing letters, in the context of epigraphic inscriptions in ancient Greek (46). Their model, named PYTHIA, uses a sequence-to-sequence NN with LSTM and was trained on the Packard Humanities Institute (PHI) database, the largest digital dataset of ancient Greek inscriptions. It gives best predictions with 30.1% character error rate, compared with the 57.3% error rate of human epigraphists (based on testing the performance of two doctoral students on the training material over 2 h).

Lastly, joining fragments of texts is one of the major challenges of restoring cuneiform manuscripts as close as possible to their original state. Matching fragments are usually only identified by a handful of experts, and the fragments are often so small as to retain only a few signs. An initial study on the Hitite corpus employed “match” classifiers, achieving good results with Maximum Entropy Classifiers (47). The Electronic Babylonian Literature project aims to reconstruct the tens of thousands of fragments that make up the remnants of ancient Babylonian and Assyrian literature. A digital corpus of largely inaccessible tablet fragments from museum collections (15,510 fragments as of June 2020) allows users to query these fragments with sequence-alignment algorithms based on the word method Basic Local Alignment Search Tool algorithm. Initial results already show that one can identify new pieces of text as well as many possible text joins. Advances in cost-effective high quality 3D scanning allow exact measurements of inscribed objects that can lead to the joining of broken tablet fragments with a matching algorithm in 3D space, as done for example by the Virtual Cuneiform Tablet Reconstruction Project (48, 49).

Materials and Methods

Neo-Babylonian Archives. Babylonian archives from the end of the sixth to the fourth century BCE are one of the main sources for reconstructing the official and ephemeral heritage of the Achaemenid Empire and its subject peoples in Mesopotamia. These “archives” were not recovered in situ but are artificial constructs imposed by modern scholars. Most Neo-Babylonian texts come from uncontrolled or poorly documented excavations, and the majority are kept in large museum collections (see below). One cannot rely on physical proximity between texts in a given find context to define an archive, since such a context is frequently unavailable or was disturbed in antiquity.

The organization of Neo-Babylonian archives by modern scholars is based mostly on an artificial division between private and institutional ownership (21). Further criteria employed to define an archive include prosopography (i.e., grouping tablets that feature a common core of principal actors engaged in connected activities), document type and content or a common setting in a social or political institution, such as a business firm, temple, or palace. Several studies try to mitigate the lack of archaeological context by employing museum-based archaeology to trace the acquisition history of related texts within a single collection or across different museums (50).

The end result, nevertheless, is that for the most part, groupings of tablets according to any of the aforementioned criteria are artificial constructs, with few exceptions.
Table 4. Breakdown of Achemenet dataset used to train our algorithm into archival and administrative text types; top 17 categories (see Dataset S6 for full list)

<table>
<thead>
<tr>
<th>Text type</th>
<th>Quantity</th>
<th>Akkadian keyword(s)</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory/list/record of transfer</td>
<td>464</td>
<td>harrānu</td>
<td>19</td>
</tr>
<tr>
<td>Business partnership</td>
<td>265</td>
<td>eṭir i mahir</td>
<td>62</td>
</tr>
<tr>
<td>Receipt</td>
<td>236</td>
<td>mahiru inbē (immovable)</td>
<td>19</td>
</tr>
<tr>
<td>Purchase</td>
<td>75</td>
<td>imittu</td>
<td>20</td>
</tr>
<tr>
<td>Promissory note for assessed field rent</td>
<td>57</td>
<td>e.g., dabābu</td>
<td>63</td>
</tr>
<tr>
<td>Statement in court/deposition</td>
<td>45</td>
<td>ana nukurribūtī</td>
<td>64</td>
</tr>
<tr>
<td>Summon/oath/injunction</td>
<td>40</td>
<td>ana id; zitti</td>
<td></td>
</tr>
<tr>
<td>Lease of arable land/orchard</td>
<td>35</td>
<td>nikkassu epēšu</td>
<td>65</td>
</tr>
<tr>
<td>Lease of movable property</td>
<td>33</td>
<td>ana id; maddatti</td>
<td>66</td>
</tr>
<tr>
<td>Letter order</td>
<td>19</td>
<td>ina muhhi</td>
<td>67</td>
</tr>
<tr>
<td>Balanced accounts</td>
<td>18</td>
<td></td>
<td>68</td>
</tr>
<tr>
<td>Lease of immovable property</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Promissory note</td>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fragmental: legal contract</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Correspondence</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marriage agreements and dowry texts</td>
<td>7</td>
<td></td>
<td>69</td>
</tr>
<tr>
<td>Work contract</td>
<td>7</td>
<td>dullu</td>
<td></td>
</tr>
</tbody>
</table>

Fortunately, the three largest text groups recognized as private archives in Achaemenid Babylonia can each be traced back to a building or room where they were deposited in antiquity: the business archives of the Egibi and Nûr-Sîn families from Babylon, and of the Mūrasu “firm” from Nippur, as well as the closely contemporary archive of the Persian governor Bêlûnu from the palace complex of Babylon, known as the Kasr archive (designated Kasr N6, ref. 50). The Mûrasû texts especially, along with another cluster of texts written in several rural centers known as the Yahudu “archive,” provide significant information on foreign minority communities in the Achaemenid Empire during a period of nearly 200 y and illuminate the fate of the Judean community in Babylonian exile (51). However, the largest textual groups from this period by far are the two multifile archives associated with city temples: the Eanna archive from Uruk and the Ebabbar archive from Sippar. These two institutional archives make up the bulk of the Achemenet dataset, alongside the private Egibi/Nûr-Sîn archive and the Mûrasû material.

Part of the Egibi/Nûr-Sîn archive in the Achemenet website (80 texts) belongs to the period of the Neo-Babylonian Empire, which technically lies outside of our chosen chronological framework. Most of these date to the reign of the last Babylonian king, Nabonidus (556 to 539 BCE). (In fact, many of the archives mentioned here are attested both in the Neo-Babylonian and early Achaemenid period, especially the larger institutional and private archives.) We nevertheless included some of the Neo-Babylonian period texts in our dataset because they are similar in type and content to the early Achaemenid period texts; in any case, they form a negligible fraction of our total dataset. Altogether, the Achaemenid period Babylonian texts on Achemenet are representative of archival groups from almost every large city in Babylonia: Babylon (Ea-eppē-û, Gahal, Nappûhu), Kiš (Eppē-û), Sippar (Bêl-rêmanî, Ea-eppē-û III, Iššar-taribi, Marduk-rêmanî, Rê’sîšî), and Uruk (Atû). The need for text restorations varies from archive to archive, depending either on their method of excavation and preservation in recent times or on the archival selection processes practiced in antiquity (e.g., some archives were regarded as discarded or “dead” archives). The best-preserved tablets found their way into museum collections in Europe and the United States following their discovery in the initial period of exploration during the late 19th and early 20th century. Many came from illicit or clandestine excavations and were acquired through a process of active selection, as curators preferred complete or nearly complete tablets over broken ones. In contrast, tablets from official excavations in Babylon and Uruk, for example, contain a higher percentage of fragmentary texts. Some large archives like Mûrasû or Kasr (which was already vitrifled from an ancient fire) were damaged by poor handling following excavation or suffered from the effects of war. A large number of Eanna tablets produced before the reign of Darius I were deliberately discarded or smashed already in antiquity after they were no longer needed by the temple administration (56, 57). The obverse of the fragmentary upper half of one of the Eanna tablets, dating to the reign of Cyrus, can be seen in Fig. 3, along with a proposed restoration that is based on known parallels and scholarly study (SI Appendix, Fig. S1).

Neo-Babylonian Text Types and Content. The Neo-Babylonian archival texts are divided into text types based on their form and content. Each Neo-Babylonian archival document, such as a promissory note or a contract, has at least three main parts: 1) an operative section made up of one or more formal clauses, usually beginning with a statement on the object(s) in question by the relevant protagonis; 2) a list of witnesses (58) (seldom accompanied by their seals on the tablet; refs. 59 and 60); and 3) a scribal signature. The latter includes not only the name and lineage of the scribe but also the place of issue and precise date given in month, day, and regnal year of the reigning king. Administrative texts, on the other hand, appear mostly in list form detailing involved objects and parties using abbreviated formulae and specific keywords. They are usually dated but do not have a scribal signatures and practically no witness lists (19, 61).

Table 4 shows the numerical breakdown of the Neo-Babylonian texts used to train our algorithm according to their respective archival and administrative text types, based on summaries of their content recorded in the Achemenet database. The division into subcategories of economic, juridical, and administrative genres is not meant to be granular, but rather inclusive, in order to reflect the different thematic elements of the corpus. Overall, there is a higher percentage of different legal archival documents, most of which contain highly structured formulae. On the other hand, the relatively high number of inventory lists, transfer documents, and other administrative material is considerably less standardized in form and content. It remains to be seen if this affected the results of our training. This is not the place to elaborate on individual text typologies, which are usually based on analysis of the main operative section of each document and take into consideration specific legal clauses or keywords, the issuing person or institution, and the prosopographical study of parties, witnesses, and scribes (see ref. 20 and the extensive references therein). Nevertheless, in order to exemplify the structure and consistency of Neo-Babylonian archival and legal formulae, most text types described in the table are also accompanied by relevant Akkadian keywords and primary reference materials.

---

* Kasr has, in fact, a mixed private and institutional background. See ref. 51 for an overview of cuneiform archives from Achaemenid period Babylonia and their time span. A more detailed discussion of each text group is found in ref. 20.

* Designations of archives are listed in parentheses following each city name. Despite being mentioned in the description on the Achemenet website, the Ur archives are not yet represented in that collection. Archives already mentioned above, like Mûrasû from Nippur, are not included in this list.

---

* The Mûrasû texts were damaged during their transport out of Nippur (54), and the Kasr texts partially survived a grim sequence of events triggered by the First World War. Many of them had already suffered ancient fire damage during or after the Achaemenid period (55).
Tokenization is an automatic process in which the text is split into words and each one is replaced by a numeric token. This is an important process that requires language-specific knowledge to prevent the loss of a great deal of semantic content. A classic example in English is tokenizing a word like “aren’t.” If we do not break it into two tokens, then it is considered a word on its own and loses the connection to “are” and “not.” While it might be possible for the learning algorithm to learn that “aren’t” is equivalent to “are not,” bad tokenization can complicate matters considerably by creating a mechanical (unnormalized) bound transcription: Akkadian phonetic spellings and logographic writings are taken at face value, by simply removing connecting hyphens between syllables and between logograms.

Open source Akkadian tokenizers use the ASCII Transliteration Format (ATF) format that our dataset does not support. Therefore, we created an alternative Akkadian tokenizer. We took into consideration some of the aspects of the current form of the transliterations. We retained the distinction between phonetic and logographic writings (i.e., italic type or roman type): during tokenization, we used the same token for both values of the aspects of the current form of the transliterations. We retained the distinction by the superscript determinative “uru” before a toponym, or by the superscript “NAME,” “GODNAME,” or “FEMALENAME” token. Locations, identified by the superscript determinative “uru” before a toponym, or by the superscript “ki” after a toponym, were replaced by a “LOCATION” token. Month names, with the superscript determinative “iti” before the noun, were replaced by MONTH, and simple numbers were replaced by “NUM.”

For this reason, we have chosen not to train the algorithm in any kind of normalization practices for the time being. In our training corpus, we remained on the level of (unbiased) transliteration, by creating a mechanical (unnormalized) bound transcription: Akkadian phonetic spellings and logographic writings are taken at face value, by simply removing connecting hyphens between syllables and between logograms.

In order to simplify the tokenization of damaged parts of the text, each sign was replaced with a tag, written in capitals, that identifies the particular type of proper name: (lû) ha-aditi t-uri, its variants range from (lû) ha-da-fa-tar, (lû) ha-da-rar-tar, and (lû) ha-da-lit a-di tary.


5. C. Götschow, Methoden zur Restaurierung von ungeräumten und gebrannten Keilschrifttexten (PettWe-Verlag, 2012).


