Language and Dialect Identification of Cuneiform Texts

Tommi Jauhiainen
Department of Digital Humanities
University of Helsinki
tommi.jauhiainen@helsinki.fi

Heidi Jauhiainen
Department of Digital Humanities
University of Helsinki
heidi.jauhiainen@helsinki.fi

Tero Alstola
Department of Digital Humanities
University of Helsinki
tero.alstola@helsinki.fi

Krister Lindén
Department of Digital Humanities
University of Helsinki
krister.linden@helsinki.fi

Abstract

This article introduces a corpus of cuneiform texts from which the dataset for the use of the Cuneiform Language Identification (CLI) 2019 shared task was derived as well as some preliminary language identification experiments conducted using that corpus. We also describe the CLI dataset and how it was derived from the corpus. In addition, we provide some baseline language identification results using the CLI dataset. To the best of our knowledge, the experiments detailed here represent the first time that automatic language identification methods have been used on cuneiform data.

1 Introduction

We have compiled a corpus of cuneiform texts intended to be used in language identification experiments. As the basis for our corpus, we used the Open Richly Annotated Cuneiform Corpus (Oracc).[1] In Oracc, the texts are stored in transliterated form. We created a tool, Nuolenna, which can transform the transliterations back to the cuneiform script. Selecting all monolingual lines from Oracc and transforming the transliterations into cuneiform, we created a new corpus for Sumerian and six Akkadian dialects.

This corpus was used in the initial experiments where the possibility of language identification in cuneiform texts was verified. In this paper, we report some of the results from the initial experiments. To the best of our knowledge, this is the first time that automatic language identification methods have been used on cuneiform data. The methods we use for language identification utilize mainly character *n*-grams and their observed probabilities in text.

For the use of the Cuneiform Language Identification (CLI) 2019 shared task[2], we extracted a dataset from the corpus. The dataset is divided into training, development, and test portions to be used in the CLI shared task which is part of the third VarDial Evaluation Campaign. We implemented four baseline language identifiers and evaluated their performance using the CLI dataset. The results of the evaluation are presented here.

2 Related work

So far, no research into language identification using cuneiform texts has been openly reported. Language identification studies involving other contemporary scripts, such as Egyptian hieroglyphs, also seem to be non-existent.

2.1 Cuneiform script and computational methods

In this section, we survey some of the research where computational methods related to language identification have been used with the cuneiform script.

Kataja and Koskenniemi (1988) discuss the description and computational implementation of phonology and morphology for Akkadian. They give examples of the rules in two-level formalism they used with the TWOL rule compiler (Karttunen et al., 1987).

Barthélemy (1998) developed and tested a morphological analyzer for Akkadian verbal forms. The analyzer works with Akkadian represented in Latin encoding (transcription).

Tablan et al. (2006) describe their project, which aims to create a tool for automatic morphological analysis of Sumerian.

[1] [http://oracc.museum.upenn.edu]

[2] [https://sites.google.com/view/vardial2019/campaign]

Proceedings of VarDial, pages 89–98
Minneapolis, MN, June 7, 2019

Among several languages, Rao et al. (2009) analyzed Sumerian written in cuneiform using conditional entropy to compare it with the Indus script. Normalized entropy of sign *n-grams* between the two scripts was used as further evidence to indicate the possible linguistic content of the texts written in the Indus script.

Ponti et al. (2013) used the K-means clustering algorithm to cluster transliterated cuneiform texts. The texts analyzed were 51 Old Babylonian letters from Tell Harmal/Šaduppûm. *Term frequency* (TF) and *term frequency - inverse document frequency* (TF-IDF) weighted words were used as features with the clustering methods. Each document was depicted as a feature vector with the length of the whole vocabulary. In K-means, the number of clusters has to be given before the algorithm is applied and Ponti et al. (2013) experimented with 2 to 15 clusters.

Luo et al. (2015) describe an unsupervised Named-Entity Recognition (NER) system for transliterated Sumerian. They compared the use of different lengths of transliterated word *n*-grams in combination with the Decision List CoTrain algorithm, and their evaluations show that word bigrams obtain the highest F1-score. In another article (Liu et al., 2015), they describe how they managed to find unannotated personal names in a corpus and suggest that the NER system could be used as an automated tool for the annotation task. Liu et al. (2016) continue the NER research on Sumerian using a variety of supervised classification methods to detect named entities.

Homburg and Chiarcos (2016) researched automated word segmentation of Akkadian cuneiform script. They used a sign list to restore CDLI[3] transliterations back to cuneiform (represented as UTF-8 characters). This is the only related work we are aware of in which cuneiform texts encoded in Unicode cuneiform have been processed previous to our experiments.

Pagé-Perron et al. (2017) present a project dedicated to creating a pipeline for Sumerian texts. The pipeline is planned to take in transliterated Sumerian and to produce POS tag annotations and lemmatization as well as machine translation into English. Chiarcos et al. (2018) further describe the work done in the project.

In order to measure inter-textual relations, Monroe (2018) calculated the cosine similarity between word vectors consisting of transliterated Late Babylonian words.

Svärd et al. (2018) used Pointwise Mutual Information (PMI) to find collocations and associations between words and word2vec to highlight paradigmatic relationships of the words of interest. They used transliterated and lemmatized Akkadian texts from Oracc.

2.2 Language identification in texts

Automatic language identification is the task of determining the language of a piece of text from the clues in the text itself. The computational methods used in language identification vary from simple wordlists to state-of-the-art deep learning methods. A recent comprehensive survey on language identification was conducted by Jauhiainen et al. (2018c). Language identification for long texts in well-resourced languages is not a difficult task, but it becomes increasingly more challenging when we target short, fragmentary, and multilingual texts in languages where the amount of training material is severely restricted. A separate challenge for language identification is dealing with closely related languages or with several dialects of an individual language. The challenge of discriminating between closely related languages has been investigated in a series of shared tasks that have been organized as part of VarDial workshops (Zampieri et al., 2014, 2015; Malmasi et al., 2016; Zampieri et al., 2017, 2018).

3 Cuneiform texts in Oracc

Our data comes from the Open Richly Annotated Cuneiform Corpus (Oracc). Oracc is an international cooperative effort containing free online editions of cuneiform texts from various projects.[4] Oracc is one of the largest electronic corpora of Sumerian and Akkadian texts, and it is regularly updated. Our data is a snapshot of Oracc from October 2016 from XML files downloaded with the permission of the site administrators. The data is comprised of 13,662 separate texts, most of which were originally written on clay tablets. Some of the texts are duplicates, and before the language identification we removed the duplicates of texts with identical Oracc identification numbers. This procedure removes modern duplicates

[3] Cuneiform Digital Library Initiative [https://cdli.ucla.edu]

[4] [http://oracc.museum.upenn.edu/doc/about/aboutoracc/index.html]

which have come into existence because a single text has been edited in several Oracc projects. Those duplicates that have different numbers and are thus different ancient manuscripts were not removed. Cuneiform writing does not mark the end of a sentence, and this is not indicated in the XML files either. Our data can, hence, be divided either into lines or texts with one or more lines. Oracc also contains some texts or words written in languages other than Sumerian and Akkadian, such as Hittite, Ugaritic, and Greek, but their number is so small that they were left out of this research.

The metadata of the texts in Oracc contains information about, for example, the provenance (the locality where the text was found), the genre, the time period in which the text was written, and so forth. The basic units in Oracc XML files are the transliterations of words, which are representations of the cuneiform signs in Latin script and which are given even if nothing else is stated about the words. Some of the cuneiform signs have, however, been broken off or are otherwise unreadable on the original tablets. In those cases, the word in question, or part of it, is replaced with an 'x' in the transliteration. The metadata of a word usually indicates its language, and some of the projects have also provided the cuneiform signs for each syllable or word of the transliteration.

The data contains many bilingual documents written in Sumerian and Akkadian. These documents often have the same text in both languages, sometimes on the same line.

3.1 Sumerian and Akkadian

Sumerian and Akkadian are ancient languages which were spoken and written primarily in Mesopotamia, present-day Iraq (Michalowski, 2004; Kouwenberg, 2011). Both languages were written in cuneiform script, but they are not related, Sumerian being a language isolate and Akkadian an East Semitic language. The cuneiform script was originally logographic in essence, then syllabic sign values were introduced to facilitate writing Sumerian, and only later was the script adapted for Akkadian. Consequently, some features of the cuneiform writing system are not ideal for Akkadian and many logograms are used side by side with syllabic spellings of Akkadian words (for further information see Seri, 2010).

Sumerian was one of the first languages ever written, and the oldest texts survive from the turn of the fourth and third millennia before the Common Era (BCE). Akkadian replaced Sumerian as the spoken language during the late third and early second millennia BCE, but Sumerian was used as a liturgical and scholarly language until the end of the cuneiform tradition at the beginning of the Common Era.

Written Akkadian is known from circa 2400 BCE onwards until the first century CE. The Akkadian language had two main dialects, Assyrian and Babylonian, both of which are present in our data. Assyrian was used in northern Mesopotamia and Babylonian in the south. There is written evidence for the simultaneous use of these dialects for 1,400 years, and both of them changed over time. The dialects are, hence, further divided into varieties designated as Old, Middle, and Neo-Assyrian and Old, Middle, and Neo-Babylonian. There was also a literary variety called Standard Babylonian which was used by both Assyrian and Babylonian scribes to write texts in certain genres. In Oracc, Middle-Babylonian is, furthermore, divided into the dialect spoken in Mesopotamia proper and the one spoken outside Mesopotamia. The latter, referred to as Middle Babylonian peripheral, is not a coherent dialect but varies somewhat from site to site. After Assyrian ceased to be a written language around 600 BCE, a variety called Late Babylonian was written for some 700 years. The differences between the dialects and their varieties are relatively small, and after learning a variety one can read the other dialect and varieties as well. In the Oracc metadata, the different dialects and varieties are given for Akkadian words in most cases.

4 Cuneiform representation in Unicode

The effort to provide a standard encoding for cuneiform began in 1999 at Johns Hopkins University as the Initiative for Cuneiform Encoding (ICE). The initiative ended up with an approved proposal for cuneiform Unicode in 2004 (officially accepted into Unicode 5.0 in 2006).[5] The final list of cuneiform signs included is a combination of work done earlier at the University of Chicago, Universität Göttingen, and the University of California, Los Angeles (Cohen et al., 2004).

[5]The Unicode Standard Version 11.0 Core Specification [http://www.unicode.org/versions/Unicode11.0.0/appC.pdf]

In the current Unicode standard, there are three blocks of cuneiform signs for the "Sumero-Akkadian" script. The first one is the block covering the base cuneiform signs ranging from U+12000 to U+123FF. The second block, from U+12400 to U+1247F, covers the cuneiform punctuation and numerals and the third, from U+12480 to U+1254F, is an extension containing additional signs for the Early Dynastic period. Unicode has only one character for each sign, even though the signs evolved through the ages. The different ways of writing the signs could be used to determine the language or dialect used or the time period of writing.

The cuneiform texts from the Oracc corpus we use in this research were provided primarily as transliterations using the ASCII Transliteration Format (ATF). ATF was first defined by CDLI and is a standardized way of electronically transliterating cuneiform, following the conventions used by cuneiform scholars in general (Koslova and Damerow, 2003). The data from the Oracc corpus is also available as JSON files in an "XCL" format (Tinney and Robson, 2018)[6], which includes a similar XML representation of the data as the CDLI archival XML format (Koslova and Damerow, 2003). We extracted the individual sign transliterations in Unicode ATF from the XML representation and recreated the transliteration for each line.

There was no available software to automatically transform the transliterations to Unicode cuneiform. As part of Oracc, there is a facility called "Cuneify," which can be used online to transform ATF into Unicode cuneiform.[7] However, it is not possible to download the software and it does not handle the Unicode ATF transliteration. In order to generate the original lines in cuneiform, we implemented a program called "Nuolenna" which takes in the transliteration generated from the XML files and re-produces the lines in Unicode cuneiform.[8] The Nuolenna program uses a list[9] of over 11,000 transliteration-sign pairs. As the base for our sign list, we used a JSON export from the Oracc Global Sign List (OGSL)[10] provided by Niek Veldhuis, to which we added some missing signs. In order to produce the original cuneiform lines, the program uses *ad hoc* rules to remove any additional annotations related to the signs. For example, sometimes an older or more precise reading can be found within parentheses directly after the reading of a sign. In such cases, we just remove the parentheses and anything between them.

5 Preliminary language identification experiments

To find out to what extent identifying the language of cuneiform text is possible, we performed initial language identification experiments using a state-of-the-art language identification method called HeLI (Jauhiainen et al., 2016). The HeLI method has recently fared well in VarDial shared tasks for Swiss-German dialect and Indo-Aryan language identification (Jauhiainen et al., 2018a,b). The experiments were conducted on individual lines as well as texts spanning several lines.

5.1 Corpus for the preliminary experiments

In Oracc, the transliterated words are separated by whitespaces, which is not the case in the original documents. In order to mimic the original documents, we removed all the whitespaces from each line of cuneiform text. We also ignored any completely broken signs, which were marked with an 'x'.

The individual words in Oracc are tagged with language or dialect information, and sometimes a single line includes words in different languages or dialects. As we set out to do language identification on monolingual texts, we used all those lines which had words in only one language, leaving out multilingual lines. The language tagging in Oracc is not always precise, and therefore some lines in our dataset might still include several languages.

In the preliminary experiments, we experimented with the language identification of both monolingual lines and monolingual texts spanning several lines with the information about line breaks retained. Mostly, each text had the lines of one original document, but if the document was multilingual, it was divided into different texts according to the languages attested.

[6] [http://oracc.museum.upenn.edu/doc/opendata/]

[7] [http://oracc.museum.upenn.edu/doc/tools/cuneify/index.html]

[8] [https://github.com/tosaja/Nuolenna]

[9] [https://github.com/tosaja/Nuolenna/blob/master/sign_list.txt]

[10] [http://oracc.museum.upenn.edu/ogsl/]

We left out the Akkadian varieties of Old and Middle Assyrian as the number of lines available for those dialects was less than 1,000. We had datasets in the Sumerian language as well as the Akkadian varieties of Old Babylonian, Middle Babylonian peripheral, Standard Babylonian, Neo-Babylonian, Late Babylonian, and Neo-Assyrian. The statistics of the corpus used in the preliminary experiments are shown in Table 1.

We were interested in experimenting in both in-domain and out-of-domain test settings as well as with language identification on two different levels: individual lines and texts. In supervised machine learning, the testing data is in-domain if it is similar to the training data. For example, if sentences are from texts that belong to the same genre or collection they are considered more in-domain than if they are not. An even stronger in-domain case is if the sentences are from the same text. Classification of test data which is in-domain with the training data is usually much easier than when it is out-of-domain. The texts in the Oracc export were in the order[11] of "projects," which are collections of texts that have some common theme. The texts in different projects can be considered to be more out-of-domain with each other than those from the same project. The projects from which the texts were extracted are listed in Table 2.

From this corpus, we generated four different test settings. For the out-of-domain experiments, we divided the corpus so that we used the first half of the corpus for training and the second half was divided between development and testing. For the in-domain experiments, we divided the corpus into parts of 20 lines or texts and took the 10 first lines or texts from each part for training, the next 5 for development, and the last 5 for testing. We, thus, ended up with four different datasets,[12] two for lines and two for texts. Each of the datasets had 50% of the material for training, 25% for development, and 25% for testing.

5.2 Results of the preliminary experiments

The HeLI method is a supervised-learning language identification method where the language models are created from a correctly tagged training corpus. The language models consist of words and sign (character) *n*-grams. When *n*-grams are extracted from a corpus, the number of unique *n*-grams is higher the longer the *n*-grams are. The actual number of occurrences of the longer *n*-grams is lower than the shorter *n*-grams. The exact optimal value for *n* depends on, among many other variables, the size of the training corpus, the length of the text to be identified, and the repertoire of the languages considered. Sometimes the longer *n*-grams could carry important information even though they are seldom found in the text to be identified. The basic idea in the HeLI method is to score individual words using the longest length *n*-grams possible. For each individual language, the words are scored first, after which the whole text gets the average score of the individual words. In the case of cuneiform text, as it is not divided into words, we use just sign *n*-grams and consider a line of text as a word as far as the HeLI method is concerned.

The individual words, or in this case lines, are scored by taking the average score of the found *n*-grams. Using the notation introduced by Jauhiainen et al. (2018c), the individual *n*-grams f found from the line to be tested, get a score R as in Equation 1:

$$R_{HeLI}(g, f) = -\log_{10} \frac{c(C_g, f)}{l_{C_g^F}} \quad (1)$$

where $c(C_g, f)$ is the count of the feature f in the training corpus C_g of the language g and $l_{C_g^F}$ is the total number of occurrences of all the *n*-grams of the same length in the training corpus. As smoothing, in case the count of a feature is zero in some languages, this version of the HeLI method uses a score $R_{HeLI}(g, f)$ for the count of one multiplied by a penalty multiplier.

Using the development sets, we optimized the sign *n*-gram range and the penalty multiplier for each setting individually. The results of these experiments are presented in Table 3. As the performance measure, we use the F1-score which is the harmonic mean of precision and recall. The results clearly show how the task of identifying a single line is much harder than that of a complete text. The task of out-of-domain identification is also clearly more difficult than that of in-domain, as was expected.

Quite many of the misclassified lines were very short; many consisted only of one sign and were truly ambiguous and often present in different dialects and even languages. Nevertheless, it was still possible to attain reasonably good language identification results. The hardest test setting was

[11] The projects were in the alphabetical order by their abbreviations.

[12] See Table 3.

Language or Dialect (abbreviation in the CLI dataset)	Texts	Lines	Signs
Sumerian (SUX)	5,000	107,345	*c.* 400,000
Old Babylonian (OLB)	527	7,605	*c.* 65,000
Middle Babylonian peripheral (MPB)	365	11,015	*c.* 95,000
Standard Babylonian (STB)	1,661	35,633	*c.* 390,000
Neo-Babylonian (NEB)	1,212	19,414	*c.* 200,000
Late Babylonian (LTB)	671	31,893	*c.* 260,000
Neo-Assyrian (NEA)	3,570	65,932	*c.* 490,000

Table 1: Number of texts, lines, and signs for each language or variety in the corpus.

Project (abbreviation used in Oracc)	SUX	OLB	MPB	STB	NEB	LTB	NEA
Bilinguals in Late Mesopotamian Scholarship (**blms**)	x	x		x			
CAMS/Anzu (**cams-anzu**)				x			
CAMS/Barutu (**cams-barutu**)		x		x			
CAMS/The Standard Babylonian Epic of Etana (**cams-etana**)		x					x
CAMS/Geography of Knowledge Corpus (**cams-gkab**)	x			x		x	x
CAMS/Ludlul (**cams-ludlul**)				x			
CAMS/Seleucid Building Inscriptions (**cams-selbi**)				x			
Cuneiform Commentaries Project on ORACC (**ccpo**)	x			x			
Corpus of Kassite Sumerian Texts (**ckst**)	x				x		
The Amarna Texts (**contrib-amarna**)			x	x			
Cuneiform Texts Mentioning Israelites, Judeans ... (**ctij**)					x	x	x
Lexical Texts in the Royal Libraries at Nineveh (**dcclt-nineveh**)	x			x			
Reading the Signs (**dcclt-signlists**)	x			x			
Digital Corpus of Cuneiform Lexical Texts (**dcclt**)	x	x	x	x			
Digital Corpus of Cuneiform Mathematical Texts (**dccmt**)	x	x		x			
Electronic Text Corpus of Sumerian Royal Inscriptions (**etcsri**)	x						
Corpus of Glass Technological Texts (**glass**)				x			
Hellenistic Babylonia: Texts, Iconography, Names (**hbtin**)						x	
Law and Order: Cuneiform Online Sustainable Tool (**lacost**)	x						
Old Babylonian Model Contracts (**obmc**)	x						
Old Babylonian Tabular Accounts (**obta**)	x	x					
The Inscr. of the Second Dynasty of Isin (**ribo-babylon2**)	x						
The Inscr. of the Period of the Uncertain Dynasties (**ribo-babylon6**)	x						
Rim-Anum: The House of Prisoners (**rimanum**)	x	x					
The Correspondence of Sargon II, Part I (**saao-saa01**)							x
Neo-Assyrian Treaties and Loyalty Oaths (**saao-saa02**)				x	x		x
Court Poetry and Literary Miscellanea (**saao-saa03**)					x		x
Queries to the Sungod (**saao-saa04**)					x		
The Correspondence of Sargon II, Part II (**saao-saa05**)							x
Legal Trns. of the Royal Court of Nineveh, Part I (**saao-saa06**)							x
Imperial Administrative Records, Part I (**saao-saa07**)							x
Astrological Reports to Assyrian Kings (**saao-saa08**)				x	x		x
Assyrian Prophecies (**saao-saa09**)							x
Letters from Assyrian and Babylonian Scholars (**saao-saa10**)				x	x		x
Imperial Administrative Records, Part II (**saao-saa11**)							x
Grants, Decrees and Gifts of the Neo-Assyrian Period (**saao-saa12**)							x
Letters from Assyrian and Babylonian Priests to ... (**saao-saa13**)					x		x
Legal Trns. of the Royal Court of Nineveh, Part II (**saao-saa14**)							x
The Correspondence of Sargon II, Part III (**saao-saa15**)							x
The Political Correspondence of Esarhaddon (**saao-saa16**)							x
The Neo-Babylonian Correspondence of Sargon and ... (**saao-saa17**)					x		
The Babylonian Correspondence of Esarhaddon and ... (**saao-saa18**)					x		
The Correspondence of Tiglath-Pileser III and ... (**saao-saa19**)					x		x

Table 2: The list of Oracc projects from which the texts in the corpus were collected.

where the language of individual out-of-domain lines was to be identified. To us, this seemed to be the most interesting and relevant setting to be used in the CLI shared task, especially if we leave out the extremely short and possibly ambiguous lines.

6 The CLI shared task

The CLI shared task 2019, part of the third VarDial Evaluation Campaign, focused on discriminating between languages and dialects written with cuneiform signs. The task included two different languages: Sumerian and Akkadian. Furthermore,

Type of setting	*n*-gram range	F1
Lines, out-of-domain	1–3	60
Lines, in-domain	1–3	72
Texts, out-of-domain	1–4	84
Texts, in-domain	1–3	93

Table 3: The F1-scores attained by the HeLI method in the preliminary experiments.

Language or Dialect	Training
Sumerian	53,673
Old Babylonian	3,803
Middle Babylonian peripheral	5,508
Standard Babylonian	17,817
Neo-Babylonian	9,707
Late Babylonian	15,947
Neo-Assyrian	32,966

Table 4: Number of lines for each language or dialect in the training set provided during the VarDial 2019 Evaluation Campaign.

the Akkadian language was divided into six dialects: Old Babylonian, Middle Babylonian peripheral, Standard Babylonian, Neo-Babylonian, Late Babylonian, and Neo-Assyrian. First, we explain how the dataset for the shared task was constructed from the corpus described earlier, and then we present the baseline language identifiers and the results we attained using them.

6.1 The dataset for the shared task

For the CLI task, we created a separate, especially tailored dataset. The participants were given texts for training and development and separate texts were given for testing at the end of the campaign. The training set was exactly the same as the one we used in the preliminary experiments[13] and the number of lines in the training portion for each language or dialect is shown in Table 4.

For the CLI development and test sets, we performed some further operations. The original datasets included duplicate lines, so we first removed all duplicates. Then we filtered out all lines shorter than three characters. After these operations, the smallest sets were those of Old Babylonian including 668 lines in the development set and 985 lines in the test set. As we wanted to make the development and the test sets equal in size between languages and dialects, we randomly selected the same number of lines from the other languages. Thus, in the CLI task, the development sets for each language consisted of 668 lines and the test sets of 985 lines.

[13] In the out-of-domain individual line identification test setting.

6.2 Baseline experiments

We used four of the methods described in the survey by Jauhiainen et al. (2018c) to implement baseline language identifiers for the CLI task. As features, we used sign *n*-grams of different lengths.

The first method is called simple scoring. In simple scoring, all the *n*-grams generated from the line to be identified M are compared to the language models and for each *n*-gram found in a language model $dom(O(C_g))$, the score R of the language g is increased by one. The language gaining the highest score is selected as the predicted language. Jauhiainen et al. (2018c) formulate the method as in Equation 2:

$$R_{simple}(g, M) = \sum_{i=1}^{l_{MF}} \begin{cases} 1 & \textbf{, if } f_i \in dom(O(C_g)) \\ 0 & \textbf{, otherwise} \end{cases} \quad (2)$$

where l_{MF} is the number of individual features in the line M and f_i is its ith feature.

The second method is the sum of relative frequencies where relative frequencies are added to the score of the language. Jauhiainen et al. (2018c) formulate the method as in Equation 3:

$$R_{sum}(g, M) = \sum_{i=1}^{l_{MF}} \frac{c(C_g, f_i)}{l_{C_g^F}} \quad (3)$$

where $c(C_g, f_i)$ is the count of the feature f_i in the training corpus.

The third method is the product of relative frequencies where the relative frequencies are multiplied together. Jauhiainen et al. (2018c) formulate the method as in Equation 4:

$$R_{prod}(g, M) = \prod_{i}^{l_{MF}} \frac{c(C_g, f_i)}{l_{C_g^F}} \quad (4)$$

The actual implementation adds together negative logarithms of the relative frequencies, which produces results with the same ordering. As a smoothing value, we used the negative logarithm of a comparably small relative frequency. The actual value was optimized using the development set. The product of relative frequencies differs from the HeLI method (Equation 1) in that it always uses the full range of available feature types (different length *n*-grams), as opposed to the HeLI

Method	*n*-gram range	F1 dev	F1 test
Prod. of rel. freq.	1–4	0.7263	0.7206
Voting Ensemble		0.7222	0.7163
HeLI	1–3 + lines	0.7171	0.7061
Simple scoring	1–10	0.6656	0.6554
Sum of rel. freq.	3–15	0.5984	0.6016

Table 5: The macro F1-scores attained by the baseline methods with the CLI dataset.

method, which uses only the longest length *n*-grams applicable.

The fourth method is a majority-voting-based ensemble of the three previous methods.

The parameters and the best possible language models are determined by training the identifier using the training set and evaluating its performance on the development set. Once the best parameters are decided, the texts in the development set can also be added to the training set for the final evaluation against the test set. We used the macro F1-score as the measure for language identification performance. For each of the methods, we evaluated all possible sign *n*-gram ranges from 1 to 15 using the development set. Table 5 shows the results for all the methods using parameters optimized with the development set. In the voting ensemble, we used the best parameters for the methods from the individual experiments, and in case of a tie, the result from the product of relative frequencies was used.

The product of relative frequencies method is clearly superior to the other two basic methods with an F1-score of 0.7206 using 2.0 as the smoothing value and sign *n*-grams from one to four. Adding the prediction information from the other two methods in the form of a voting ensemble also fails to improve the result. The F1-score achieved when using the HeLI method does not reach the one from the product of relative frequencies method either. The F1-score gained by the HeLI method is clearly higher than the score attained in the preliminary experiments, which was as expected, as we had filtered out some of the most difficult cases.

Table 5 is a confusion matrix displaying the exact number of identifications. The diagonal values represent correct identifications. Standard Babylonian and Neo-Babylonian were the most difficult varieties to distinguish, mostly being erroneously identified as each other. Late Babylonian was the easiest to identify, with a recall of over 96%.

Lang.	LTB	MPB	NEA	NEB	OLB	STB	SUX
LTB	***947***	6	9	34	13	25	8
MPB	3	***858***	51	94	84	69	55
NEA	6	26	***780***	185	26	148	26
NEB	4	19	81	***535***	30	160	30
OLB	3	22	12	16	***736***	47	110
STB	17	35	30	113	43	***491***	101
SUX	5	19	22	8	53	45	***655***

Table 6: Confusion matrix for the product of relative frequencies method. The rows indicate the actual languages and the columns indicate predicted languages. Correct identifications are emphasized.

7 Conclusions and future work

In this paper, we have shown that it is possible to perform language and dialect identification in cuneiform texts encoded in Unicode characters. We have created a dataset to be used in the VarDial Evaluation campaign and evaluated the performance of four baseline identifiers using the dataset.

Some sizeable Oracc projects were left out of the corpus, for example the "Royal Inscriptions of the Neo-Assyrian Period" project, as the exact dialect of the Akkadian language had not been annotated. Furthermore, for the same reason, only the lines in Sumerian could be used from the "Royal Inscriptions of Babylonia online (RiBo)" project. We believe that automatic dialect identification could be useful in making the annotations more detailed and are planning to provide this kind of automatically deduced information as part of the Korp version of Oracc.[14]

Some other avenues for further work are language set identification for the multilingual texts, as well as unsupervised clustering of data without any predefined languages.

Acknowledgments

The research was carried out in the context of the "Semantic Domains in Akkadian Texts" project and the Centre of Excellence in Ancient Near Eastern Empires at the University of Helsinki (ANEE), both funded by the Academy of Finland. This work has also been partly funded by the Kone Foundation and FIN-CLARIN.

We thank Johannes Bach, Mikko Luukko, Aleksi Sahala, and Niek Veldhuis for their valuable comments during the experiments and Raija

[14] http://urn.fi/urn:nbn:fi:lb-2018071121

Mattila and Saana Svärd for their continued support. We are grateful to Robert Whiting for revising the language of the text and for further insight into the Akkadian and Sumerian languages.

References

François Barthélemy. 1998. A Morphological Analyzer for Akkadian Verbal Forms with a Model of Phonetic Transformations. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, pages 73–81. Association for Computational Linguistics.

Christian Chiarcos, Ilya Khait, Émilie Pagé-Perron, Niko Schenk, Christian Fäth, Julius Steuer, William Mcgrath, and Jinyan Wang. 2018. Annotating a Low-Resource Language with LLOD Technology: Sumerian Morphology and Syntax. *Information*, 9.

Jonathan Cohen, Donald Duncan, Dean Snyder, Jerrold Cooper, Subodh Kumar, Daniel Hahn, Yuan Chen, Budirijanto Purnomo, and John Graettinger. 2004. iClay: Digitizing Cuneiform. In *The Proceedings of the 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST 2004)*, pages 135–143.

Timo Homburg and Christian Chiarcos. 2016. Akkadian Word Segmentation. In *Proceedings of the 10th International Conference on Language Resource Evaluation (LREC 2016)*, pages 4067–4074.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2018a. HeLI-based Experiments in Swiss German Dialect Identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 254–262, Santa Fe, NM.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2018b. Iterative Language Model Adaptation for Indo-Aryan Language Identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 66–75, Santa Fe, NM.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2016. HeLI, a Word-Based Backoff Method for Language Identification. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 153–162, Osaka, Japan.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2018c. Automatic Language Identification in Texts: A Survey. *arXiv preprint arXiv:1804.08186*.

Lauri Karttunen, Kimmo Koskenniemi, and Ronald Kaplan. 1987. A compiler for two-level phonological rules. Technical report, Stanford University, Center for the Study of Language and Information.

Laura Kataja and Kimmo Koskenniemi. 1988. Finite-state Description of Semitic Morphology: A Case Study of Ancient Akkadian. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING 1988)*, volume 1, pages 313–315, Budapest, Hungary.

Natalia Koslova and Peter Damerow. 2003. From Cuneiform Archives to Digital Libraries: The Hermitage Museum Joins the Cuneiform Digital Library Initiative. In *Proceedings of the 5th Russian Conference on Digital Libraries (RCDL 2003)*, St.-Petersburg, Russia.

Bert Kouwenberg. 2011. Akkadian in General. In *The Semitic Languages: An International Handbook*, pages 330–340. De Gruyter Mouton.

Yudong Liu, Clinton Burkhart, James Hearne, and Liang Luo. 2015. Enhancing Sumerian Lemmatization by Unsupervised Named-Entity Recognition. In *Proceedings of the Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2015)*, pages 1446–1451, Denver, Colorado.

Yudong Liu, James Hearne, and Bryan Conrad. 2016. Recognizing Proper Names in UR III Texts through Supervised Learning. In *Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference (Flairs 2016)*, pages 535–540.

Liang Luo, Yudong Liu, James Hearne, and Clinton Burkhart. 2015. Unsupervised Sumerian Personal Name Eecognition. In *Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference (Flairs 2015)*, pages 193–198.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jrg Tiedemann. 2016. Discriminating Between Similar Languages and Arabic Dialect Identification: A Report on the Third DSL Shared Task. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–14, Osaka, Japan.

Piotr Michalowski. 2004. Sumerian. In *The Cambridge Encyclopedia of the World's Ancient Languages*, pages 19–59. Cambridge University Press.

M Willis Monroe. 2018. Using Quantitative Methods for Measuring Inter-Textual Relations in Cuneiform. *Digital Biblical Studies*, pages 257–280.

Émilie Pagé-Perron, Maria Sukhareva, Ilya Khait, and Christian Chiarcos. 2017. Machine Translation and Automated Analysis of the Sumerian Language. In *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 10–16, Vancouver, BC. Association for Computational Linguistics.

Giovanni Ponti, Daniela Alderuccio, Giorgio Mencuccini, Alessio Rocchi, Silvio Migliori, Giovanni Bracco, and Paola Negri Scafa. 2013. Data Mining Tools and GRID Infrastructure for Assyriology Text Analysis (an Old-Babylonian Situation Studied Through Text Analysis and Data Mining tools). http://www.eneagrid.enea.it, Published 2017 in: R. De Boer and J. G. Dercksen, editors, *Private and State in the Ancient Near East - Proceedings of the 58th Rencontre Assyriologique Internationale at Leiden, 16–20 July 2012, Eisenbrauns*.

Rajesh P. N. Rao, Nisha Yadav, Mayank N. Vahia, Hrishikesh Joglekar, R. Adhikari, and Iravatham Mahadevan. 2009. Entropic Evidence for Linguistic Structure in the Indus script. *Science*, 324(5931):1165–1165.

Andrea Seri. 2010. Adaptation of Cuneiform to Write Akkadian. In Christopher Woods, editor, *Visible Language. Inventions of Writing in the Ancient middle East and Beyond*, volume 32 of *Oriental Institute Museum Publications*, pages 85–98. The Oriental Institute of the University of Chicago.

Saana Svärd, Heidi Jauhiainen, Aleksi Sahala, and Krister Lindén. 2018. Semantic Domains in Akkadian Texts. In Vanessa Bigot Juloux, Amy Rebecca Gansell, and Alessandro Di Ludovico, editors, *CyberResearch on the Ancient Near East and Neighboring Regions. Case Studies on Archaeological Data, Objects, Texts, and Digital Archiving*, volume 2 of *Digital Biblical Studies*, pages 224–256. Brill.

Valentin Tablan, Wim Peters, Diana Maynard, and Hamish Cunningham. 2006. Creating Tools for Morphological Analysis of Sumerian. In *Proceedings of the 5th Internation Conference on Language Resources and Evaluation (LREC 2006)*, pages 1762–1765, Genoa, Italy.

Steve Tinney and Eleanor Robson. 2018. Oracc Open Data: A brief introduction for programmers. *Oracc: The Open Richly Annotated Cuneiform Corpus*, Oracc.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubeic, Preslav Nakov, Ahmed Ali, Jrg Tiedemann, Yves Scherrer, and Nomi Aepli. 2017. Findings of the VarDial Evaluation Campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–15, Valencia, Spain.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language Identification and Morphosyntactic Tagging: The Second VarDial Evaluation Campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Santa Fe, USA.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 2014. A Report on the DSL Shared Task 2014. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 58–67, Dublin, Ireland.

Marcos Zampieri, Liling Tan, Nikola Ljubeić, Jrg Tiedemann, and Preslav Nakov. 2015. Overview of the DSL Shared Task 2015. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 1–9, Hissar, Bulgaria.