Towards the Integration of Cuneiform in the OntoLex-Lemon Framework

Timo Homburg & Thierry Declerck

Abstract. This publication shows our approach to adding representations of graphemes of the cuneiform script into the Ontolex-Lemon model. We define a new vocabulary that adds representations of graphemes and their variants, including etymology and their representations in character description languages. We describe how the ontology model can be generalized to describe graphemes of languages that do not rely on a written script for communication. We then interlink these representations to the Ontolex-Lemon model on one end and, for some instances, to the CIDOC-CRMtex model on the other hand and provide application examples in different scripts.

1. Introduction

The Ontolex-Lemon model (McCrae et al., 2017) is used by many big data repositories such as Wikidata (Vrandečić and Krötzsch, 2014) or Babelnet (Navigli and Ponzetto, 2012)[1] to represent lexical information about words, word forms, and their relation to semantic descriptions. Words are often depicted in some kind of writing system, the representations of which may give a researcher additional information about writing styles, different sign variants used to express certain characters and words, and their occurrences. This publication proposes a complementary ontology to the Ontolex-Lemon model, which can capture shapes of cuneiform characters in a semantic web vocabulary. This extension is to be thought of as an extension to represent signs and sign variants of the cuneiform script. Still, it should be understood as so general that it could be applied to other similar typed languages. We

Timo Homburg 0000-0002-9499-5840
i3mainz–Institute for Spatial Information & Surveying Technology, Mainz University of Applied Sciences, 55128 Mainz, Germany. E-mail: timo.homburg@hs-mainz.de
Thierry Declerck 0000-0002-9450-6648
DFKI GmbH, Multilinguality, and Language Technology Lab, Saarland University Campus D3 2, Germany. E-mail: declerck@dfki.de

1. See also https://babelnet.org/ for more details.

Y. Haralambous (Ed.), *Graphonlinguistics in the 21st Century 2022. Proceedings*
Graphonlinguistics and Its Applications (ISSN: 2681-8566, e-ISSN: 2534-5192), Vol. 9.
Fluxus Editions, Brest, 2024, pp. 265–297. https://doi.org/10.36824/2022-graf-homb
ISBN: 978-2-487055-04-9, e-ISBN: 978-2-487055-05-6

envision a second use case of this ontology model to represent sign languages, as described in (Declerck, 2022), but will, for brevity, mainly exemplify the primary use case of representing languages in the cuneiform script.

2. Foundations

Cuneiform signs are comprised of cuneiform wedges, which according to (Homburg, 2021) can be described using the following parameters:

- A wedge direction on the unit circle
- An optional wedge size identifier
- Indicators of their shape (e.g., broken, wedge head type, wedge stroke type)

While the cuneiform script itself is part of the Unicode standard and about 900 cuneiform signs[2] are attested, these cuneiform signs may appear in a variety of glyph shapes, which differ in the amount and positioning of the cuneiform wedges. The reasons for these changes in glyph shapes may be different writing styles of the same cuneiform sign in space and time, different habits of scribes of the cuneiform tablets, or possible other explanations concerning the adjacent signs of the respective cuneiform sign on given tablets. This situation is not uncommon in other scripts. For example, in Chinese, differences in the number of different stroke types per Chinese character exist not only traditional Chinese characters and Simplified Chinese characters but also between Chinese characters used in Japanese (Kanji) and in their usage over time (Galambos, 2021; Liang, 2021).

3. Related Work

This section discusses related work on linked data dictionaries, character encodings, and data formats common in cuneiform languages used for building the linked data-based character registry.

3.1. Linked Data Dictionaries

Linked data dictionaries (Gracia, Kernerman, and Bosque-Gil, 2017) provide, among other benefits, means of connecting words and word forms in written language to concepts in the semantic web, thus allowing natural language processing approaches to extract knowledge from a given textual context more accurately. Linked data dictionaries exist for many languages in well-known data repositories such as Wikidata or Babelnet. For cuneiform languages, the MTAAC (Baker et al., 2017) or ORACC (Tinney and Robson, 2014) corpora provide a suitable basis

2. https://www.unicode.org/charts/PDF/U12000.pdf

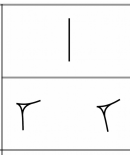
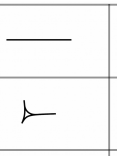
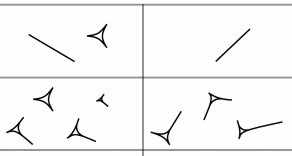
the elements					Sign EME 
designation	a	b	c	d	abc
parameters	sum of the designations and the indices				a3 b5 c1
category	number of elements				9 = 3+5+1

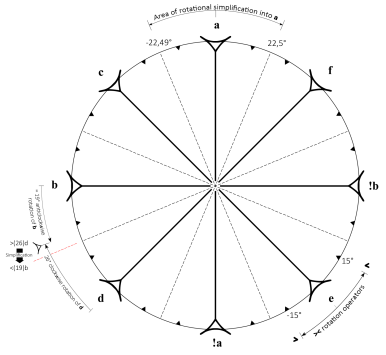
FIGURE 1. Gottstein System for Cuneiform signs from (Gottstein, 2013)

for the extraction of linked data dictionaries. However, such a process has not been attempted to the author's knowledge. In the future, we can expect linked data dictionaries to be present for each major language.

3.2. Character Description Languages

For many non-alphabetic languages composed out of strokes, such as Japanese or Chinese, encodings for the description of their character composition have been proposed. The Chinese character description language (Bishop and Cook, 2003a) can compose Chinese characters for font generation. Similar character description languages like KanjiVG[3] exist for Japanese. To the author's knowledge, fonts for cuneiform languages (Mousavi and Lyashenko, 2017; Píška, 1999; 2008) have been based on either SVG drawings or JPG images of cuneiform signs. Hence, unlike the Chinese character description languages, they have not relied on character description languages to describe their respective cuneiform characters. Images will give an accurate representation of the character in question but do not encode semantic information about the context of the character and its composition—something we deem necessary for a proper digital representation of structured scripts. Character descriptions for cuneiform languages have been attempted by (Panayotov, 2015) and (Homburg, 2019). The Gottstein system for describing cuneiform signs counts the number of wedge types in a cuneiform sign, whereas wedge types are distinguished into four different types, as shown in Figure 1. Sometimes, the Gottstein system is slightly adjusted to define the Winkelhaken wedge (w), i.e., the wedge type with only the wedge head as its distinct type, e.g., in (Homburg, Zwick, Mara, and Bruhn, 2022). PaleoCodage (cf. Figure 2a) aims to capture the structure of cuneiform signs, represent dif-

3. https://kanjivg.tagaini.net



(a) PaleoCodage encoding system: Wedge types are assigned to wedges on the unit circle. Operators allow for the modification of wedges for the representation of a certain degree on the unit circle (Homburg, 2021).

(b) Sign variants of the same cuneiform sign E in the same space and time and found in the same location and described with different PaleoCodes

FIGURE 2. PaleoCodage encoding system and sign variants of the same cuneiform sign E

ferent sizes of cuneiform wedges, and aims to capture repetitions of substructures of cuneiform signs. This enables PaleoCodage to accurately model cuneiform sign variants even in the same spatiotemporal context as shown in Figure 2b. Given two established character description languages for the cuneiform script, cuneiform characters can be described with two different goals in mind: To index them per cuneiform wedge types (Gottstein) and to describe their shape using PaleoCodage. Both representations may, to a certain extent, be convertible to RDF and, depending on the needs of respective scholars, can serve as a basis for querying different features of these abstracted representations of cuneiform signs.

3.3. ATF and JTF

To transliterate cuneiform tablets, two main transliteration formats exist. The ASCII Transliteration Format (ATF)[4] is the primary format of distribution of cuneiform transliterations for all cuneiform languages and exists in many different dialects and varieties which often differ per repository. JTF[5] is a JSON format (Bray, 2017) that includes the same elements as ATF but in a better machine-processable and extendable

4. http://oracc.museum.upenn.edu/doc/help/editinginatf/cdliatf/index.html
5. https://idcs.hypotheses.org/234

format. It is currently adopted by the Cuneiform Digital Library Initiative (CDLI)[6] and possibly other repositories as a storage format for cuneiform transliterations. Cuneiform transliterations can be rendered from JTF to ATF so that JTF does not provide a replacement format for ATF. Given these two common transliteration formats for cuneiform language transliterations, the JTF format seems to be suited to be extendable for linked data, as defining a JSON-LD context is an easy way to create compatibility with the ontology model we define. Both of the aforementioned transliteration formats do not provide support for paleographic descriptions in any way.

4. Extending the Ontolex-Lemon Model for Cuneiform Paleography

This section outlines our approach for integrating the cuneiform script into the Ontolex-Lemon model. At first, we introduce some terminology we use in our ontology model in Section 4.1, then describe the digital representation of a character in cuneiform languages in Sections 4.2 and 4.3 and how to represent its composition in Section 5.2. After discussing the relation of characters in the ontology model to Ontolex-Lemon Section 4.5, we focus on the description of relations, shape, and provenance of different graphemes by introducing a comprehensive paleographic description vocabulary Section 5. Finally we discuss the integration of etymology concepts in Section 6 and conclude the description of the ontology model by introducing terms to describe glyph occurrences Section 6.2.

4.1. Preliminary Definitions

In order to define a vocabulary for describing characters, we would first like to define certain terms that will be used throughout this publication. These definitions are intended to be so general that they may also be applicable to other languages with similar scripts.

Definition 4.1. – Glyph*: The physical manifestation of a grapheme on a written medium.*

This definition covers written glyphs on any medium and is equivalent to the concept `http://cidoc-crm.org/cidoc-crm/TX9_Glyph` in CIDOC (Doerr, 2005) CRMtex (Murano and Felicetti, 2021). This would be a single cuneiform sign depicted on a written medium (e.g., a clay tablet) for cuneiform. This cuneiform sign might be a non-standard variant. It

6. See `https://cdli.ucla.edu/` for more details.

might deviate from this standard variant because the glyph might be broken and have a different number of wedges or wedges not pointing in the expected directions. For non-written languages, such as sign languages, the ontology model provides a class `http://www.purl.org/graphemon#Movement` to represent, e.g., hand gestures.

Definition 4.2. – Grapheme*: Digital representation of relevant features of a representation of a glyph or equivalent non-written representation.*

A grapheme represents an idealized or canonical form of a set of glyphs, represented by a digital representation, i.e., abstraction of the set of glyphs describing the cuneiform sign and may be described by an identifier such as a Unicode code point or a dictionary entry number.

Definition 4.3. – GraphemeVariant*: A variant of a Grapheme that is associated with the same Unicode codepoint or a semantically equivalent identifier and other identifiers but differs in its normalized visual appearance.*

A `http://www.purl.org/graphemon#GraphemeVariant` is usually connected to a variety of Glyph instances that represent the respective Grapheme variant on physical artifacts in space and time.

Definition 4.4. – GraphemeManifestation*: The manifestation of a grapheme either on a written medium or using non-written means.*

We define a `http://www.purl.org/graphemon#GraphemeManifestation` as a more general concept for a Glyph. We would like to generalize the ontology model not to exclude, e.g., hand gestures of sign languages that may be represented using video media or representations of spatio-temporal descriptions of positions of movements. As a superclass of GrpahemeManifestation we define the class `http://www.purl.org/graphemon#SymbolicRepresentation` to represent all representations which created a symbolic value in any language.

Definition 4.5. – GraphemePart*: A representation of a grapheme that is found as a part of some other Graphemes in the same script.*

A `http://www.purl.org/graphemon#GraphemePart` definition relates to parts of characters found in other characters, but also to parts of, e.g., hand gestures that are part of another hand gestures to describe a particular concept. A grapheme part may constitute its own character. If so, it will be represented with its own Grapheme representation, i.e., also be an instance of Grapheme in the linked data graph.

Definition 4.6. – AtomicPart*: A representation of an atomic part out of which Graphemes are comprised.*

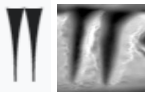
A `http://www.purl.org/graphemon#AtomicPart` may represent its own meaning, and Graphemes that consist of precisely one atomic part may exist. An example of an atomic part in cuneiform languages would be a single vertical cuneiform wedge which describes the number one in a

Grapheme. In Chinese, it would be a single stroke that describes a Chinese character (e.g., the horizontal stroke for the number one). However, in Chinese, a horizontal stroke alone might also describe the meaning of horizontal, even though it cannot be used as a Grapheme meaning in this language. In non-written languages, such as in sign language, an atomic part depicts a single unique movement that may be combined with other movements to describe a more sophisticated concept.

4.2. What Constitutes a Grapheme?

To describe what constitutes a grapheme in cuneiform languages, we define the following rules, which could also be implemented for automated classification. We assume a cuneiform sign variant to be the standard cuneiform sign variant to represent a particular meaning of a cuneiform sign across time and space. This standard cuneiform sign variant (i.e., its canonical form) might be the most occurring form that the respective linguistic community has agreed upon. It might also be defined per corpus, for example, the most occurring form in a certain corpus. This standard form could be linked to the grapheme data instance, that we define in our knowledge graph. If no such form exists, the grapheme instance in the knowledge graph will simply link to all known grapheme variant instances. For example, consider the cuneiform sign A[7], which

(a) The cuneiform sign A with its standard form once as grapheme and once as an actual occurrence in the cuneiform text HS 367, front side, column 1, line 3, sign 4



(b) The cuneiform sign A with an alternative form is more common in older cuneiform texts once as grapheme and as an actual representation in HS 1163, back side column 1, line 14, sign 4. This form also resembles the cuneiform sign for the number two 2(disz).

FIGURE 3

constitutes of three vertical cuneiform wedges with at least one attested meaning of water and is described with PaleoCode *a-a:a* shown in Figure 3a. We define a sign variant to A as a variant that differs in one of the following criteria:

7. https://en.wiktionary.org/wiki/\%F0\%92\%80\%80

C.1 Amount of cuneiform wedges per type
C.2 Positioning of cuneiform wedges towards each other
C.3 Changes in the type of cuneiform wedges at their respective positions

Figure 3b constitutes such a variant. This example also shows that sign variants may also have the shape of a different standard variant of a sign. In this case, the sign variant of A has the same shape and amount of vertical wedges as the standard variant of sign 2(disz) with the meaning of the number 2.

The definitions also mean that there are differences in cuneiform glyphs that we do not constitute as representing a new sign variant, i.e., a grapheme in the graph structure:

D.1 The writing order of wedges if known and not exposing a semantic of their own
D.2 The style of cuneiform wedges themselves (e.g., cuneiform head, cuneiform stroke)
D.3 The absolute sizes of cuneiform wedges, as long as their proportional size are the same
D.4 Changes in color or material on which the cuneiform wedges are imprinted unless they capture a semantic meaning

While we deem the latter characteristics not as relevant to distinguish between individual graphemes, they are essential information that should be added to the glyph description in cuneiform languages. Concerning the writing order of wedges, research has started some preliminary work (Taylor, 2014), but has not come to a definite conclusion. However, as long as the writing order of the wedges does not affect criteria C.1-C.3, it is of no relevance for the classification of glyphs as we define in this publication.

4.3. Representation of Graphemes in Linked Data

We propose encoding cuneiform graphemes in linked data with two different methods. The first method encodes graphemes using character description language representations like PaleoCodage, the Chinese character description language (Bishop and Cook, 2003b), or the American sign language (Liddell et al., 2003) transliteration as RDF text literals. When no character description languages are available, or as alternative means of definition, SVG literals (Ferraiolo, Jun, and Jackson, 2000) seem to be the natural choice because SVG literals may be displayed in a browser and may serve as the basis for a font generated from the given sign list. Alternative representations might include Open Type Font (Toledo and Rosenberg, 2003) Paths or other image formats such as PNG (Boutell, 1997), which can represent the respective

grapheme. For sign languages, videos or representations of spatiotemporal motions are also viable options. The former may be represented by a hyperlink, the second may use spatial text literals such as Well-Known Text (Herring et al., 2011) in combination with time point extensions. Our ontology model defines literal types for each of these representations.

A second method is to expose the elements that contribute to generating a grapheme directly in RDF. For cuneiform signs, this means that every cuneiform wedge present in a grapheme is represented by its own RDF instance. Hence, a grapheme consists of an RDF subgraph of interconnected AtomicParts. This representation might further semantic exploitation of the individual grapheme but is not practical if queries targeting the grapheme representation only should be answered.

We discuss how to encode a cuneiform sign in RDF using the example of the cuneiform sign A, which we introduced in Section 4.2. The PaleoCode for this grapheme is a-a:a, that is, a vertical wedge *next-to* a vertical wedge *over* a vertical wedge. We can represent this grapheme in RDF as shown in Figure 4. In this example, the grapheme is assigned

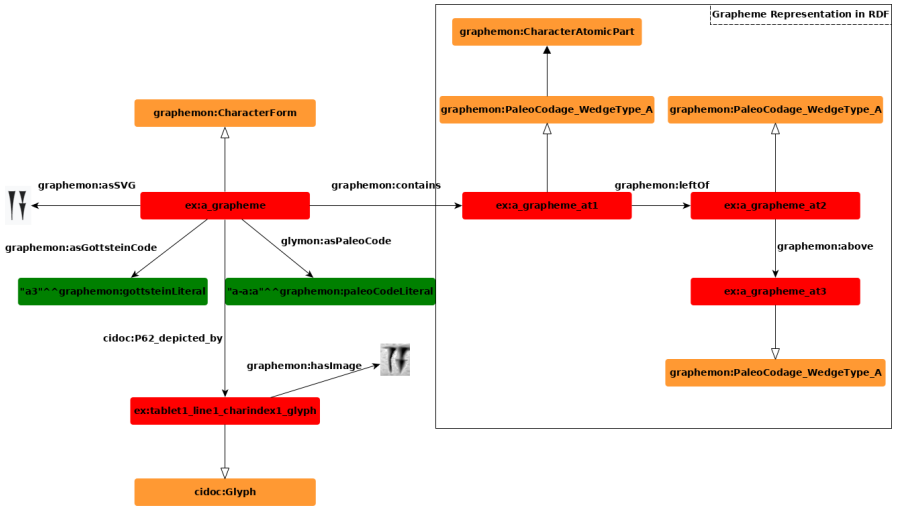


FIGURE 4. Representation of the grapheme structure of cuneiform sign A described with PaleoCode a-a:a

representations in SVG (`http://www.purl.org/graphemon#svgLiteral`), PaleoCodage (`http://www.purl.org/graphemon#paleocodageLiteral`), and in the Gottstein encoding (Panayotov, 2015) (`http://www.purl.org/graphemon#gottsteinLiteral`) and points to a glyph occurrence, while at

the same time, the glyph structure from the PaleoCode is extrapolated in an RDF representation on the right-hand side of the graph. In this RDF representation, even wedge types and atomic parts can be fleshed out in pure RDF. While the literal representations allow querying images of glyphs easily and ready for display by e.g., web browsers, RDF subgraphs may be used to query for clusters of similar representations and also to name such representations in the knowledge graph. Hence, they allow a comparison of shapes of glyphs using the SPARQL query language only, as sets of glyph atomic parts become similarly-shaped subgraphs.

4.4. Grapheme Atomic Parts

In the RDF representation features of single atomic parts, cuneiform wedges could be annotated. That is, each wedge could be annotated with its level of damage or be categorized into a writing style of a different area or scribe. Clearly, this example is only valid for the cuneiform script, and other scripts might include different elements of representation. However, we think these elements could be surmised under a common class structure, which groups similarly styled scripts. For example, Chinese, Japanese, and Cuneiform are all stroke-based scripts, for which the AtomicPart is a stroke of some kind. Figure 5 shows two atomic parts, strokes used in Chinese, the horizontal and the vertical stroke. Both strokes are integral parts of the character for 10, which in itself is included in the word for 11. As an atomic part, the horizontal stroke is also the character for 1, while the vertical stroke is not. Depending on the language, the atomic parts of characters often exhibit a certain order in which they are written. This order may be strict, for example, in Chinese, or it may be superimposed by the encoding used to describe the character variant, such as in the case of cuneiform. To represent a writing order of character atomic parts, these may be described in a `http://www.w3.org/1999/02/22-rdf-syntax-ns#List` or a position vocabulary that we introduce later on in this publication. Sometimes, it may also be sufficient to just state that certain atomic parts are available in a certain grapheme or grapheme part. In this case, a simple `http://www.purl.org/graphemon#partOf` relation is sufficient (cf. Figure 5).

Finally, one might want to capture how the atomic parts of characters are drawn to recreate the abstract character representation for a font. The list of atomic character parts to draw may be appended with positional information extracted from the individual character encoding.

4.5. Connection to Ontolex-Lemon

An important element of this model is its interconnectivity to the Ontolex-Lemon model for modeling semantic dictionaries. To link sign

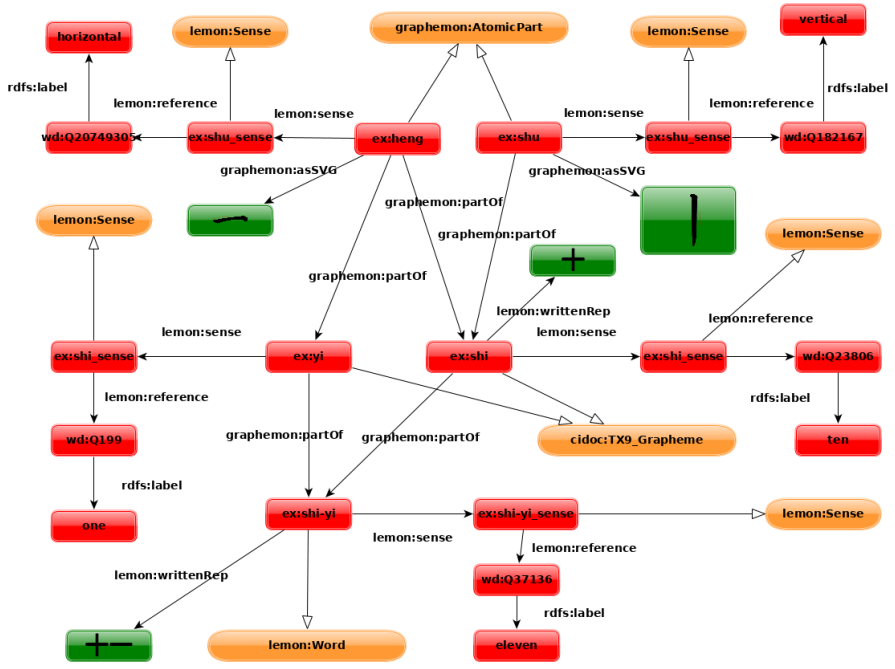


FIGURE 5. Combination of atomic parts: The to strokes, heng, and shu, which are used to build Chinese characters, are atomic parts and used in the character shi, which is used to build the word shi-yi, 11.

representations to Ontolex-Lemon word forms, we need to relate components of these word forms to grapheme representations. Unfortunately, an Ontolex-Lemon word form does not have a relation to link to individual graphemes. Instead, it is only possible to link to textual representations of words and word forms as transcriptions, transliterations, or written representations, in essence, represented as text literals. While we cannot change the Ontolex-Lemon model, we can link grapheme instances to instances of word forms described by the Ontolex-Lemon model. To do that, we need to define a new element called `http://www.purl.org/graphemon#WordformOccurrence`, which attests to the representation of a word form with assigned grapheme representations. Figure 6 shows one example connection of Ontolex-Lemon to our ontology model using the word "a," in its word form "a_form" and an occurrence of this word form being represented by the grapheme, represented by a variant of the grapheme for the cuneiform sign "A". In other words, this graph representation allows expressing that a word form can be represented with certain grapheme variant combinations.

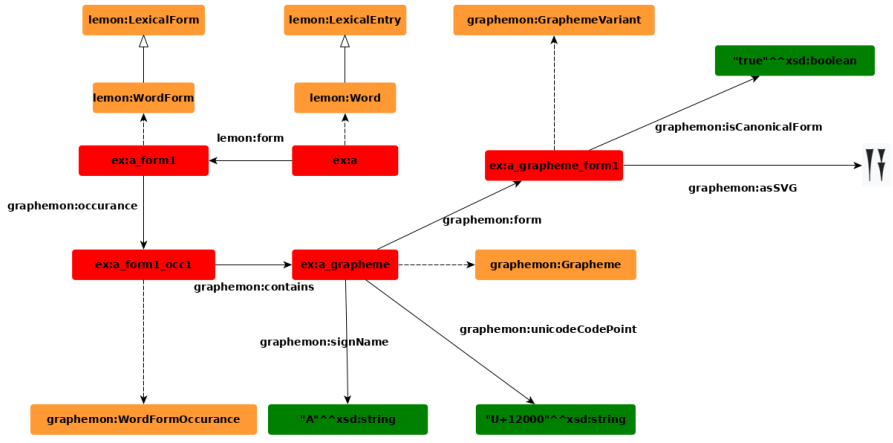


FIGURE 6. Connection Ontolex-Lemon model to grapheme representations: Word forms may be represented by a list of one or many grapheme variants

5. Paleographic Description Vocabulary

In the previous section, the possibility to express graphemes with character description languages and their serialization in an RDF subgraph was mentioned. To achieve the representation of the RDF subgraph, the elements of the respective character description languages need to be provided in RDF. We describe these elements in two different parts, one vocabulary to classify atomic parts of graphemes and one vocabulary representing the relations between these atomic parts in the form of directions.

5.1. Atomic Part Description Vocabulary

This part of Graphemon defines properties that describe features of individual cuneiform wedges, i.e., atomic parts in addition to an atomic part classification, as depicted by a subclass of `\glymonnsAtomicPart`. The reasoning behind a description of the grapheme representation is, that often grapheme representations in themselves contain semantic information that can be explicitly expressed in a knowledge graph.

Table 1 shows the kinds of attributes we can assign to a cuneiform wedge that might have an influence on its semantic meaning. We would like to stress that we are not discussing the shape or color of individual glyphs here but the shape of a derived grapheme. For example, a grapheme depiction with a filled wedge head often indicates, that the shape of the grapheme is meant to describe a cuneiform sign inscribed

TABLE 1. Atomic part description vocabulary for parts of a cuneiform wedge grapheme that represent a semantic meaning and therefore need to be represented in the knowledge graph

Relation	Description
graphemon:angle	Describes the angle by which the atomic part is rotated if applicable
graphemon:headColor	Describes the color of the head of the cuneiform wedge relative to a given scale
graphemon:headSize	Describes the size of the head of the cuneiform wedge relative to a given scale
graphemon:hasFilledHead	Describes whether the cuneiform wedge head is filled or empty
graphemon:strokeColor	Describes the color of the stroke of the cuneiform wedge relative to a given scale
graphemon:strokeSize	Describes the size of the head of the cuneiform wedge relative to a given scale
graphemon:partStyle	Describes the style of the cuneiform wedge in a style description language such as CSS

FIGURE 7. Two different grapheme styles which represent cuneiform signs. The grapheme style with the empty wedge head represents a sign variant present on clay tablets, and the style with the filled wedge head a variant present on stone inscriptions.

on stone rather than on clay (cf. Figure 7). Graphemes of cuneiform signs inscribed on clay usually depict an empty cuneiform wedge head. Therefore, the style in which the grapheme is depicted might in itself contain information about the circumstances in which the grapheme can be found, and useful information to be added to the knowledge graph.

5.2. A Vocabulary for Directions

PaleoCodage and further character description languages relate the different atomic parts of a character to each other by a set of operators and define or reuse an explicit or implicit order of atomic parts. To describe

a cuneiform sign but also further structured scripts in RDF, we formalize these relations in our RDF vocabulary as follows: Individual items may be connected using a set of positional relationships exhibited by the following vocabularies shown in Table 2 to represent the physical relation between atomic parts.

Table 2. Relationships between atomic parts: Atomic parts of cuneiform characters

Relation	Description
graphemon:above	indicates that the current atomic part is above the previous atomic part
graphemon:below	indicates that the current atomic part is below the previous atomic part
graphemon:downright	indicates that the current atomic part is on the lower right of the previous atomic part
graphemon:downleft	indicates that the current atomic part is on the lower left of the previous atomic part
graphemon:exactPosition	Describes the exact position of the atomic part in a fixed coordinate system
graphemon:left	indicates that the current atomic part is left of the previous atomic part
graphemon:right	indicates that the current atomic part is right of the previous atomic part
graphemon:upperright	indicates that the current atomic part is on the upper right of the previous atomic part
graphemon:upperleft	indicates that the current atomic part is on the upper left of the previous atomic part

Table 2 shows the sets of operators we defined to target the cuneiform script. Beginning with a first atomic part, the structure of the cuneiform script follows a subgraph of relations until no such relation can be found. In future work, it may be necessary to define further operators and relations to describe other script types.

5.3. Grapheme Relation Vocabulary

Within a script, such as cuneiform, one may encounter parts of individual graphemes reused in other parts of the script. An initial experiment on the representation of all cuneiform Unicode codepoints in one time period-specific font (Homburg, 2021) found that about two-thirds of all cuneiform signs had repeated components in them. Hence, it seems natural to encode these relations in our grapheme description vocabulary so that they can be correlated with, e.g., meanings of the single individual signs and possibly with etymology. When describing the cuneiform

script, we can derive part of individual graphemes from two different sources. The first source may be the definition of the cuneiform signs in standards such as Unicode. For example, the Unicode cuneiform sign AN/AN (AN over AN)[8] is defined by the cuneiform sign AN[9] over another instance of cuneiform sign AN. This makes AN/AN a Grapheme instance which is comprised of two GraphemeParts representing AN. While this definition is used in Unicode, we can generally assume that this definition is not valid for all Graphemes covering all time periods. The reason is that cuneiform signs developed from pictographs and will take the shape on which the Unicode definition is based only at a certain point in time. The second source to derive GraphemeParts from is the representation of Graphemes in a character description language or another structured format. This method was used in (ibid.) and has the distinct advantage that actual representations of GraphemeVariants can be compared by using established and reproducible similarity metric results such as Levenshtein Distance or Image Similarity metrics. In the cuneiform script, as in many other similar scripts, such as Chinese, there are parts of signs that repeat in other signs. This might mean that these signs are related semantically, e.g., that one sign extends a concept introduced by the first sign, that the meaning of two different signs is combined or that the inclusion of one sign in the other has been an artistic choice of the scribe. To model these relations, Table 3 describes properties to express the most occurring types of relations between graphemes. These definitions include two kinds of properties: Properties that derive their conclusions, e.g., from similarity metric calculations, and properties that describe assertions derived by other means. By other means we refer to e.g., the Unicode definition, a scholarly paper or any other external resource which is not readily available and therefore retraceable in the knowledge graph. While these definitions are enough to model relations between cuneiform characters, they might need to be extended for different other scripts.

6. Etymology Vocabulary for Graphemes

Etymology is an important concept that helps understand how words have evolved. The Etymological WordNet (De Melo, 2014) showed first how the etymology of words could be traced using a semantic web vocabulary and (Khan, 2018) suggested that the idea of tracing etymology could also be applied to the Ontolex-Lemon model. The resulting ontology model, Etymon (Etymology Model for Ontologies)

8. https://en.wiktionary.org/wiki/\%F0\%92\%80\%AE
9. https://en.wiktionary.org/wiki/\%F0\%92\%80\%AD

TABLE 3. Relation vocabulary between graphemes: Graphemes may be part of other graphemes, modified parts of other graphemes, a generalization, or a combination of other graphemes. Statements like these may stem from metrics or assertions.

Relation	Description
graphemon:isDescribedToBePartOf	The grapheme is described to be part of the target grapheme
graphemon:isDescribedAsMergedPartOf	The grapheme is described to be a modified part of the target grapheme
graphemon:isDescribedAsGeneralizationOf	The grapheme is described to be a generalization of the target grapheme
graphemon:isDescribedAsModifiedPartOf	The grapheme is described to be a modified part of the target grapheme
graphemon:isDescribedAsSimplificationOf	The grapheme is described to be a simplification of the target grapheme
graphemon:isGeneralizationOf	The grapheme is a generalized form of the target grapheme
graphemon:isModifiedPartOf	The grapheme is part of the target grapheme, but slightly altered
graphemon:isMergedPartOf	The grapheme is merged out of at least two different other graphemes
graphemon:isPartOf	Describes the subject grapheme as part of the target grapheme
graphemon:isSimplificationOf	The grapheme is a simplified form of the target grapheme

or lemonETY, describes essential relations and concepts for Etymology that we adjust for the representation of etymology in graphemes. In particular, the concepts `http://lari-datasets.ilc.cnr.it/lemonEty#Cognate`, `http://lari-datasets.ilc.cnr.it/lemonEty#Etymon`, and `http://lari-datasets.ilc.cnr.it/lemonEty#Derivative` are defined in this ontology model. We reuse these concepts in our ontology model but define them on a grapheme level to capture differences in graphemes. Figure 8 shows three examples of an etymological development of cuneiform signs over time. Similar to words, capturing these etymological relations can be of tremendous value for Assyriology research and appropriate machine learning classification tasks. Figure 9 shows how we represent etymology in our ontology model using the example of one cuneiform character in two stages of development. We must stress that the depiction of etymology is just one way to relate grapheme representations to each other. To be precise, etymology describes an inter-

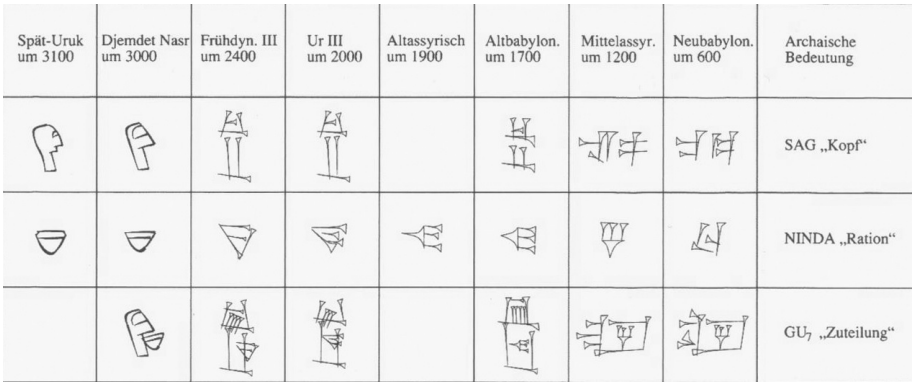


FIGURE 8. The etymology of cuneiform characters over time from a pictorial representation to a more abstract representation. Not all representations are depicted by cuneiform wedges. (Labat, 1995)

preted semantic relationship between grapheme representations, even if the semantic is only founded by the sign being a previous or following variant. Another way to represent the similarity between graphemes is to directly exploit their image representations or abstractions thereof.

6.1. Grapheme and Glyph Similarity

Grapheme similarity might be calculated by similarity measures based on either a String representation of the grapheme represented in a sign description language, i.e., a formal textual representation of the glyph depicted, or by a similarity metric based on the pictorial or other representations (e.g., 3D models) of the glyph itself. To enable these kinds of relations in the ontology model, we define one DatatypeProperty score and three base classes `http://www.purl.org/graphemon#SimilarityMetric`, `http://www.purl.org/graphemon#ImageBasedSimilarityMetric`, and `http://www.purl.org/graphemon#StringBasedSimilarityMetric`, from which we might derive script-specific subclasses to express relations between grapheme variants. Table 4.

We recommend using similarity metrics that can be normalized to a percentage range between 0-100 so that comparisons between different similarity metrics can be simplified. However, we do not want to restrict a user from defining arbitrary similarity metric definitions, as long as they are sufficiently documented in the knowledge graph. Given similarity metrics, etymological relationships and assertions about grapheme structures up until the atomic part level, we believe that the relation between graphemes have been sufficiently modeled.

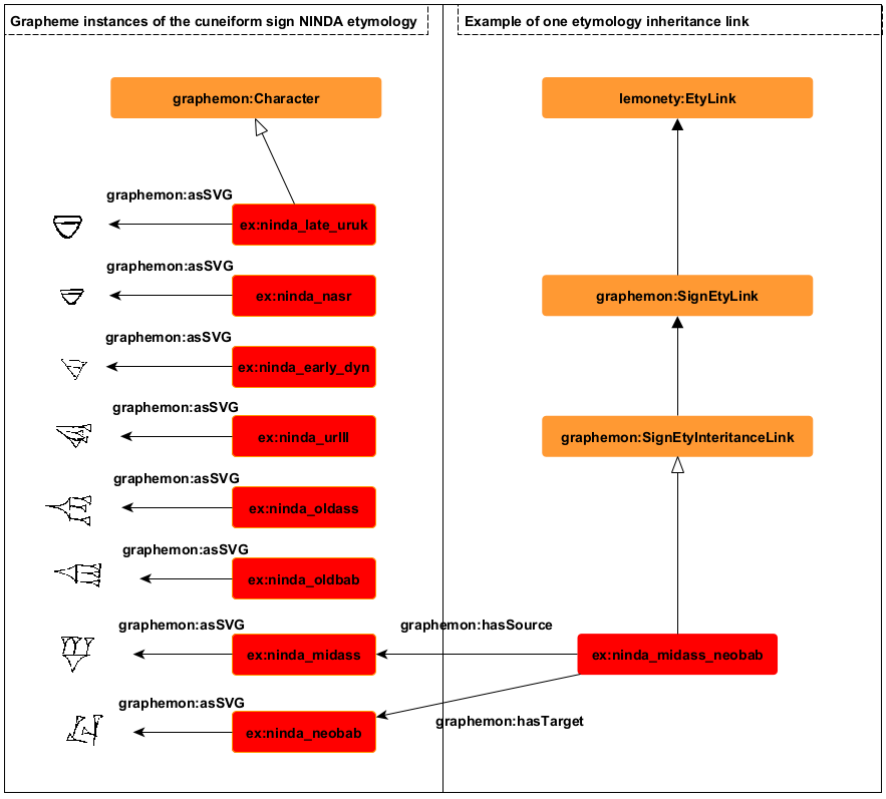


FIGURE 9. Etymology representation of graphemes in the ontology model (only one etymological relation is shown for brevity)

TABLE 4. Classes and properties describing superclasses for similarity metrics and results of similarity metric calculations between grapheme instances or glyph instances

graphemon:SimilarityMetric	Class
graphemon:SimilarityMetricResult	Class
graphemon:ImageBasedSimilarityMetric	Class
graphemon:StringBasedSimilarityMetric	Class
graphemon:score	DatatypeProperty

6.2. Glyph Description Vocabulary

This part of the vocabulary deals with describing visual features of glyph representations. On the example of a cuneiform glyph on a cuneiform

tablet, we will show the aspects of visual representation we deem necessary to be represented in our vocabulary:

- Color representation using the Color Ontology[10] or using CSS literals (`http://www.purl.org/graphemon#cssLiteral`)
- Indicators of damage either on the glyph itself or in its given encoding
- Indicators of the origin of the writers
- Material aspects of the material which was used to represent the glyph
- Metadata of the written script (time period, scribe, etc.)

These vocabulary extensions help identify glyphs by their visual features, another perspective that cuneiform researchers often apply. The Graphemon data model defines the aforementioned properties to be able to model rudimentary features of glyph representations. However, the authors believe that each of these features may be better fleshed out in other vocabularies specializing in the respective fields. Nevertheless, we found it to be a necessity for a researcher to be able to model glyph properties to be able to set them into relation to grapheme represenations. In this way, researchers may draw conclusions about the accuracy of the grapheme representations in relation to the given glyph representations.

7. Applicability of the Ontology Model for Other Languages

While the ontology model we have proposed is intended for the cuneiform script, we argue that the model also applies to a variety of similarly structured scripts and beyond written languages. We give two examples of written scripts that might benefit from the ontology model and one example of how sign languages, as representatives of non-written languages, can be described using the same or slightly varying terminology.

7.1. Egyptian Hieroglyphics and Hieratic Script

Recently, the paleography of Egyptian hieroglyphics and the hieratic written version of these have been digitally captured (Gülden, Krause, and Verhoeven, 2020) and published as a database at the university of Mainz[11]. Databases like these constitute an ideal application case for our ontology model, and this particular database even exposes part of its data as linked open data. As an example of further applicability,

10. `https://github.com/timhodson/colourphon-rdf`
11. `https://aku-pal.uni-mainz.de/graphemes`

we pick out grapheme A18[12] (child with a crown), which is attested as a hieroglyphic ideogram and in three hieratic written forms. Figure 10 shows the etymology relationship between two written grapheme variants, which have been attested in different dynasties. In this particular case, the Grapheme for a child with a crown is both the oldest attested Grapheme and the canonical Grapheme upon which the written graphemes are based. The ontology model can be applied to similar scripts and, if properly interlinked, enable comparison between graphemes across languages. For example, the shapes of graphemes could be compared across languages by connecting them through their attested meaning.

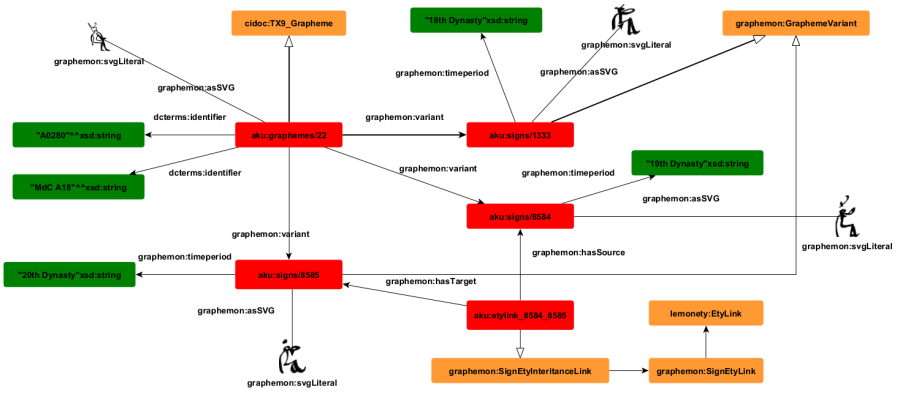


FIGURE 10. Application example of the Graphemon model using Egyptian hieroglyphics: Etymology of the grapheme A18 in hieratic written script

7.2. American Sign Language

A second application case can be seen in the American Sign language ASL. In the American sign language, gestures to describe a word may vary by location, even within the same sign language. As an example, we point to the sign language description for the term "school," as exemplified on `https://www.signasl.org/sign/school`. This site provides 10 video recordings of people performing the gesture denoting the word school in the American sign language. Most gestures describe the term school with two hands tapping together[13]. However, different dialects

12. `https://aku-pal.uni-mainz.de/graphemes/22`
13. `https://media.signbsl.com/videos/asl/elementalaslconcepts/mp4/school.mp4`

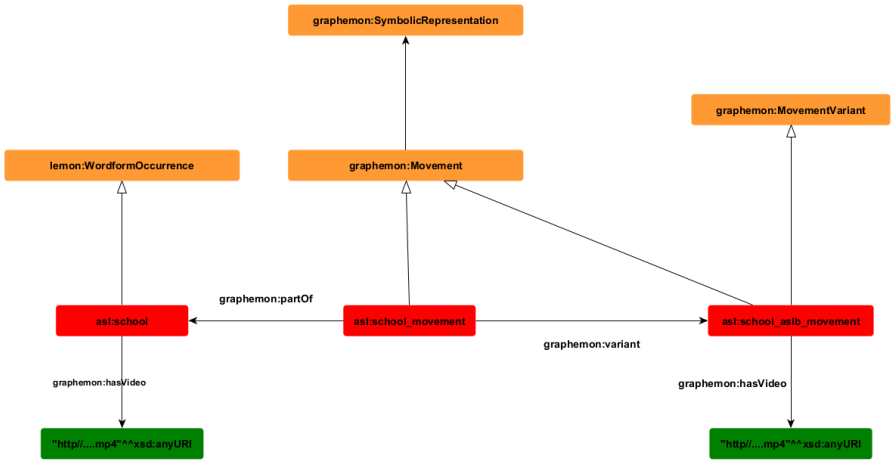


FIGURE 11. Application of the Graphememon model on a hypothetic variant of the American Sign Language (ASL)

of the American Sign language might employ different variants of the base hand gesture, which, in the Graphemon ontology model, would be treated as Grapheme variants.

Figure 11 shows how gestures may be modeled using the Graphemon ontology model. Each gesture becomes an instance of `http://www.purl.org/graphemon#Movement`, an abstract class for gestures. If a sign language like ASL is defined with a standard gesture vocabulary, variants of these gestures become de-facto variants of the initially defined gestures in ASL. As the main topic of this publication is the modeling of written scripts, especially cuneiform, we would like to point out that this part of the ontology model is likely to be fleshed out in future work, as gestures used in sign languages might depict other properties than the written script which will be needed to be modeled as properties in an extended ontology model. Therefore, extensions to the model might likely be developed in future work.

8. Application cases

This section discusses the implications of the definition of the ontology model we propose and shows applications in cuneiform studies which directly benefit from its modeling capabilities. In general, we believe that access to structured information about paleography and graphemes, as well as their variants constitutes a missing part in the documentation of primarily digital scholarly editions (Gabler, 2010) of

texts of a different kind. Research on paleography has been done in recent years (Stokes, 2015), and the need for a paleographic vocabulary specific for cuneiform has even been voiced in (Homburg, 2020), but systematic documentation of grapheme variants, their occurrences, and linkage to grammatical forms described by the Ontolex-Lemon model can provide a database to tackle research questions which combine questions of linguistics and paleographic research, an area which is sought to be better understood in a variety of languages (e.g., Maya language, hieratic script, cuneiform, Chinese). In the following, we exemplify immediate application cases enabled by the ontology model with a specific emphasis on cuneiform languages.

8.1. A Cuneiform Sign Variant Registry

A cuneiform sign variant registry is a web-based repository that allows the registration of grapheme variants of cuneiform signs, including its spatio-temporal context and further attributes. It attests these variants in different cuneiform transliterations, in different cuneiform languages, and on different cuneiform artifacts. The data structure we propose can be seen as the foundation of such a sign variant registry, which, apart from the functionality of encoding signs, might also help Assyriologists to search for a particular grapheme in its spatio-temporal contexts and find representations of this Grapheme as actual glyph image representations. In essence, the cuneiform sign registry needs to be able to store:

- GraphemeVariants described with unique identifiers and accompanied with metadata:
  - Spatiotemporal context
  - Attested cuneiform language
  - Etymology mappings
  - References to texts or URIs to annotations that describe the sign variant
- Sign definition as an image or in a character description language
- Search indices as similarity metric results between cuneiform signs

Figure 12 shows a screenshot of the JavaScript test tool for PaleoCodage. It can create cuneiform sign variants by entering the character description language code and stores already entered PaleoCodes in a git repository. The repository contents may be downloaded in an RDF representation. An extended version of this PaleoCode storage with support for etymology, cuneiform languages, textual references, and further metadata based on the Graphemon Ontology model is our vision for storage and good accessibility of cuneiform sign variants. The architecture of this repository already fulfills the criteria to represent cuneiform sign

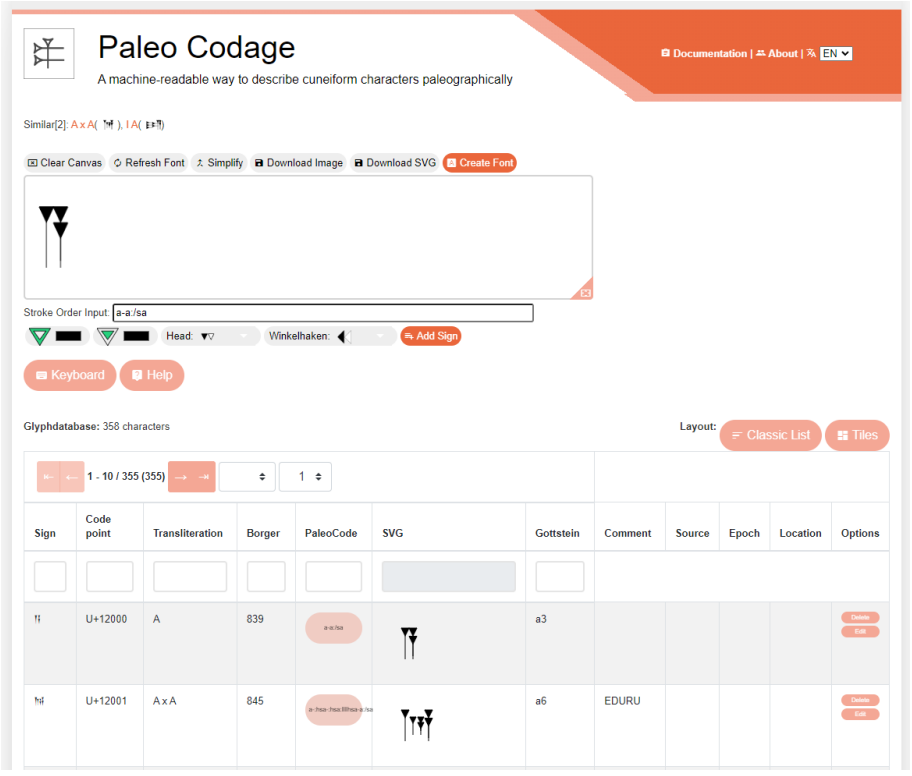


FIGURE 12. A precursor of a cuneiform sign registry which may be extended with the Graphemon ontology model as a backend

variants and can calculate similarity metrics results between its character representations. We believe it may be applicable to other language types as well.

8.2. Integration of Grapheme Information in Cuneiform Digital Editions

Cuneiform digital edition formats should be able to incorporate Graphemon data, as it is represented in this publication. We, therefore, investigate the suitability of data formats for this purpose and highlight what integration in these formats entails. We thereby have the following assumptions:

1. A cuneiform character variant registry as described in Section 8.1 exists so that graphemes may get their own identifiers (possibly also URIs)
2. The data format should aim to be a single file format for easier portability

The ATF format in any shape does not allow to add annotation information. It does not allow encoding information about character variants without defining yet another ATF dialect such as P-ATF (Homburg, 2021).

```
@tablet
@obverse
1. 3(u)_v1
```

Listing 1. Paleographic extension to the ATF format as suggested by (ibid.). This extension requires unique IDs of graphemes to be defined and used in the actual transliteration text.

Listing 1 shows the proposed P-ATF encoding of (ibid.). Each grapheme is assigned a unique ID used directly in the transliteration. The definition of such IDs is currently arbitrary, as the related work on cuneiform sign variants does not show a universally accepted identifier system for cuneiform signs. While such an identifier could be delivered with the URI or be part of a URI that describes a grapheme, the practicality of usage for the average Assyriologist would be to either use some kind of grapheme autocompletion system dependent on a centralized registry of graphemes or not use yet another dialect of ATF, but rather to treat grapheme variants as text annotations.

The situation differs for TEI/XML-based transliteration representations and JSON-based transliterations such as JTF. TEI/XML allows the representation of glyphs[14] so that links to graphemes and glyph representations as URIs could be drawn. The most promising format, in our opinion, would be a JSON-LD-based representation as an extension of the JTF format.

```
{"_class":"object", "type":"tablet", "children":[
....
{"_class":"chr", "type":"U+1200",  "value":"a",  "grapheme"
    :"GRAPHEMEID","glyphrep":"GLYPHREPRESENTATIONLINK"}
}
....
}
```

Listing 2. JTF format extended to link to grapheme representations

14. https://tei-c.org/release/doc/tei-p5-doc/de/html/ref-glyph.html

Listing 2 shows a hypothetical cuneiform tablet transliteration representation in JTF[15]. Somewhere in this transliteration, a character transliterated as *a* is attested and referenced to a Unicode code point. In our ontology model, the Unicode code point may identify the standard Grapheme, e.g., by resolving its URI using a SPARQL query. We add the keys "grapheme" and "glyphrep" to identify the grapheme variant via its URI and to identify a representation of the actual glyph on the cuneiform tablet in one of the literal representations we propose. A picture or another medium might represent this glyph. JTF even allows us to define our grapheme variants in the same file if needed and can easily be related to a JSON-LD context (Sporny et al., 2014) for conversion to a linked data representation. In this way, one could build applications that create new grapheme variants in JTF files, which are later synchronized with a cuneiform sign registry in a repository where the transliteration in JTF is supposed to be stored.

8.3. Annotation of Grapheme Variants With Annotorious

Annotorious[16] and Recogito[17] are two open-source annotation libraries in JavaScript which allow for annotations in the W3C Web Annotation Data model (Sanderson, Ciccarese, and Van de Sompel, 2013). The creation of annotations seems like the ideal place to use the Graphemon ontology model. Annotations in linked data are comprised of an annotation target, e.g., an area defined on an image resource and an annotation body. The annotation body describes the annotation information which is attested to the annotation target. Figure 13 shows a customized extension of Annotorious, which creates annotations on images of cuneiform tablet surfaces. The annotation objects created with this tool describe an image area with a PaleoCode and a transliteration string. With both information, a set of cuneiform grapheme variant URIs can be retrieved from the knowledge graph, which the user may confirm. The user may be asked to create and describe a new grapheme variant if no URI can be found. Either way, a URI to describe the selected image area is added to the annotation, making image annotations relatable to Graphemes. This way, an Assyriologist may easily document their Grapheme variants and, using the knowledge graph, find further occurrences of the same Grapheme in other texts for comparison.

15. https://github.com/cdli-gh/jtf-lib
16. https://github.com/recogito/annotorious
17. https://github.com/recogito/recogito-js

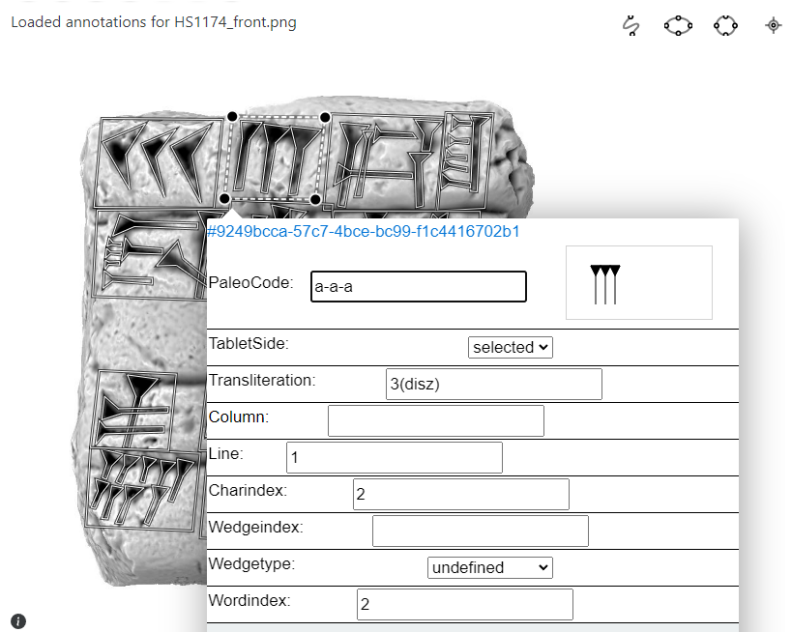


FIGURE 13. Creation of an annotation on a cuneiform 3D rendering using the software Cuneiform Annotator on the MaiCuBeDa dataset (Mara and Homburg, 2023). The marked area denotes the Glyph. The Grapheme is described using a PaleoCode and a Transliteration which can be mapped to a sign name (i.e., a Grapheme representation)

8.4. Sample Queries

This section presents sample queries that the new ontology model enables. We show typical applications which are relevant for Assyriology, computational linguistics, and the domain of machine learning.

8.4.1. *Find Graphemes With Similar Structures*

```
SELECT ?graphemevariant ?glyphimage WHERE {
    ?graphemevariant glymon:hasSimilarity ?graphemevariant_sim .
    ?graphemevariant_sim rdf:type ?PaleoCodeStringSimilarity .
    ?graphemevariant_sim rdf:value ?simvalue .
    glymon:hasImage ?glyph .
    FILTER(?simvalue>0.8)
}
```

LISTING 3. A sample query which allows to query cuneiform sign graphemes of similar structure

Listing 3 selects all graphemes above a given similarity threshold of a chosen similarity score. This allows Assyriologists to find similar grapheme variants of cuneiform signs for the sign currently examined and generate similarity statements within the respective text corpus they investigate.

8.4.2. *Etymology of Graphemes*

We can ask for the etymology of graphemes in two different ways and possibly at least two different motivations. The first motivation is to find out about different variants of a grapheme in a specific time period. For example, in Listing 4 we would like to retrieve every Grapheme, including its attested grapheme variants in the Old Babylonian period of cuneiform writing.

```
SELECT ?grapheme ?graphemevar ?graphemesvg ?timeperiod WHERE {
    ?grapheme rdf:type cidoc:TX9_Grapheme .
    ?grapheme graphemon:variant ?graphemevar .
    ?graphemevar graphemon:timeperiod ex:OldBabylonian .
    ?graphemevar graphemon:as\index{SVG}SVG ?graphemesvg .
}
```

LISTING 4. Example of querying for etymology relations of a given grapheme

The second motivation is to represent the etymology relations of a given cuneiform sign explicitly and to query similarities between them.

```
SELECT ?etymon ?grapheme ?graphemesrc WHERE {
    ?grapheme rdf:type cidoc:TX9_Grapheme .
    ?grapheme graphemon:variant ?graphemevar .
    ?etymon graphemon:hasTarget ?grapheme .
    ?etymon graphemon:hasSource ?graphemesrc .
}
```

LISTING 5. Example of querying for etymology relations of a given grapheme

Listing 5 queries all graphemes linked in an etymological chain as described in Section 8.4.2. The graphemes can be visualized for assessment by Assyriologists or for extraction by preparation scripts for machine learning analysis.

8.4.3. *Artifacts Including Special Graphemes*

As a third application, we would like to highlight the possibility of different visualizations of grapheme metadata. Similar to already existing approaches such as the cuneiform site index (Rattenborg, 2019), which display cuneiform tablet excavation locations, applications to display the occurrences of specific grapheme variants have not been

present in cuneiform studies. Considering a paleographic enrichment of cuneiform artifact data, one may use the metadata of cuneiform artifacts to create spatial distributions of grapheme occurrences. To achieve this, the Graphemon ontology model needs to be combined with the linked data representations describing the contents of a cuneiform tablet, which can be achieved with the JTF representation presented in Section 8.2. If the knowledge graph includes information on the glyphs on each individual tablet connected to its individual grapheme, each Glyph occurrence can be related to a specific location. Hence, it is possible to create a map representation of glyph occurrences by querying the ontology model.

```
SELECT ?grapheme ?graphemevar ?graphemesvg ?timeperiod WHERE {
    ?tablet rdf:type cunei:Tablet .
    ?tablet geo:hasGeometry ?tablet_geom .
    ?tablet_geom geo:asWKT ?tabgeo .
    ?tablet cunei:contains ?wordformocc .
    ?wordformocc cunei:contains ?grapheme .
    ?grapheme rdf:type cidoc:TX9_Grapheme .
    ?grapheme graphemon:variant ?graphemevar .
    ?graphemevar graphemon:as\index{SVG}SVG ?graphemesvg .
}
```

LISTING 6. Example of querying for etymology relations of a given grapheme

Listing 6 shows a query returning geocoordinates of findspots of the cuneiform tablets, including a specific GraphemeVariant according to the ontology model. The findspot points can be visualized on a map with an additional indicator as a color, e.g., the time period in which they were found.

A final application case can be discovering and identifying rare graphemes on cuneiform tablets. For this use case, we assume that a sufficiently large corpus of cuneiform tablets has been described using an extended JTF corpus, as described in Section 8.2. One information we can derive from this corpus is the frequency of usage of individual grapheme variants. Rare grapheme variants are graphemes that are not used very often compared to other grapheme variants describing the same grapheme.

```
SELECT DISTINCT ?grapheme ?graphemvar COUNT(DISTINCT ?wordformocc AS ?
    graphemvarcount) ?graphemesvg WHERE {
    ?grapheme rdf:type cidoc:TX9_Grapheme .
    ?grapheme graphemon:variant ?graphemevar .
    ?wordformocc graphemon:contains ?graphemevar .
}
```

LISTING 7. Example of querying for etymology relations of a given grapheme

Listing 7 states a SPARQL query to retrieve every grapheme with every grapheme variant and an occurrence count of each grapheme variant in the whole corpus. The result may be used as a ranking to retrieve common sign variants and may be combined with other metrics to get an accurate view of their distribution.

9. Conclusions

This publication presented a complimentary ontology model to the Ontolex-Lemon model, which can represent graphemes and grapheme variants. This model provides the opportunity to create and contribute to a linked open data cloud of graphemes, glyphs, and signs that can help researchers analyze and discover connections between different visual grapheme representations to classify and retrace the similarities and origins of paleography phenomena. Not only can graphemes be described, but they can also be related to words and actual occurrences of Glyphs, allowing the graph to be used for structured querying, e.g., to obtain instances for targeted machine learning systems. In this way, once enough data has been accumulated, a significant obstacle for machine learning tasks such as sign recognition or cuneiform tablet time period classification (Dencker, Klinkisch, Maul, and Ommer, 2020; Mara and Bogacz, 2019) can be overcome: The acquisition of relevant machine learning data for suitable automation tasks. For Assyriologists, integrating the Graphemon knowledge graph into repositories such as Wikidata, similar to the integration of Ontolex-Lemon for words, would help in documenting, classifying, and including paleographic information in emerging digital editions while at the same time being readily accessible for any data science approaches. We investigated how graphemes may be represented through different media, e.g., character description languages, images, or even videos, in the case of non-written gesture-based languages and how established similarity metrics may compare these different representations. This allows comparing and discovering similar graphemes using different characteristics, which can prove invaluable if sign registries for graphemes are created. Finally, we presented approaches to create, store, manage and query information from the gained knowledge base. The aforementioned components should lead to a better understanding and modeling grapheme variants and cuneiform signs.

9.1. Future Work

We see this specification's future work in exploring other scripts and grapheme representations in different languages and consolidating

these results in a working group such as W3C Ontolex[18]. The definition of a unified model for graphemes would allow repositories such as Wikidata to integrate word forms and their semantics in the form of glyph representations. Ideally, we would like to see Wikidata or a similar repository become the data backend of a sign variant registry for cuneiform or any other script that can be modeled in this way. For cuneiform studies, in particular, a formalized knowledge base of this kind is a precious resource for research and retraceability of grapheme variants, and we expect the adoption of these ideas by cuneiform repositories in the future—the adoption of which might pose further research questions and challenges which might need to be addressed.

Acknowledgements

This paper is based upon work from the COST Action NexusLinguarum—European network for Web-centered linguistic data science (CA18209), supported by COST (European Cooperation in Science and Technology). It is also supported by the Horizon 2020 research and innovation program with the project Prêt-à-LLOD (grant agreement no. 825182).

References

Allen, Julie D. et al. (2012). *The Unicode Standard*. Mountain View, CA: The Unicode Consortiume.

Auer, Sören et al. (2007). "Dbpedia: A nucleus for a web of open data." In: *The Semantic Web*. Springer, pp. 722–735.

Baker, Heather D et al. (2017). "Machine Translation and Automated Analysis of Cuneiform Languages (MTAAC)." In: *Humanities Commons*.

Bishop, Tom and Richard Cook (2003a). "A specification for CDL Character Description Language." In: *Glyph and Typesetting Workshop, Kyoto*. http://coe21.zinbun.kyoto-u.ac.jp/papers/ws-type-2003/098-cdl.pdf.

———— (2003b). "Character description language CDL: The set of basic CJK unified stroke types." https://unicode.org/L2/L2003/03420-cdl-strokes.pdf.

Boutell, Thomas (1997). *PNG (Portable Network Graphics) Specification Version 1.0*. RFC 2083.

Bray, Tim (2017). *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC 8259.

18. https://www.w3.org/community/ontolex/

Cidoc, Crm (2003). "The CIDOC Conceptual Reference Model." http://cidoc.ics.forth.gr.

Cyganiak, Richard, Markus Lanthaler, and David Wood (2014). *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation. W3C.

De Melo, Gerard (2014). "Etymological Wordnet: Tracing The History of Words." In: *LREC*, pp. 1148–1154.

Declerck, Thierry (2022). "Towards a new Ontology for Sign Languages." In: *Proceedings of the Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 3977–3983.

Dencker, Tobias et al. (2020). "Deep learning of cuneiform sign detection with weak supervision using transliteration alignment." In: *Plos one* 15.12, e0243039.

Doerr, Martin (2005). "The CIDOC CRM, an ontological approach to schema heterogeneity." In: *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Doerr, Martin, Francesca Murano, and Achille Felicetti (2017). "Definition of the CRMtex." In: https://www.cidoc-crm.org/crmtex/sites/default/files/CRMtex_v1.0_March_2020.pdf.

Eckle-Kohler, Judith, John Philip McCrae, and Christian Chiarcos (2015). "LemonUby–A large, interlinked, syntactically-rich lexical resource for ontologies." In: *Semantic Web* 6.4, pp. 371–378.

Ferraiolo, Jon, Fujisawa Jun, and Dean Jackson (2000). *Scalable vector graphics (SVG) 1.0 specification*. Bloomington: iuniverse.

Gabler, Hans Walter (2010). "Theorizing the digital scholarly edition." In: *Literature Compass* 7.2, pp. 43–56.

Galambos, Imre (2021). "Chinese Character Variants in Medieval Dictionaries and Manuscripts." In: ed. by Jörg B. Quenzer, pp. 491–512.

Gottstein, Norbert (2013). "Ein stringentes Identifikations- und Suchsystem für Keilschriftzeichen." In: *Mitteilungen der Deutschen Orient-Gesellschaft zu Berlin* 145.

Gracia, Jorge, Ilan Kernerman, and Julia Bosque-Gil (2017). "Toward linked data-native dictionaries." In: *Electronic Lexicography in the 21st Century: Lexicography from Scratch. Proceedings of the eLex 2017 conference*, pp. 19–21.

Greenwade, George D. (1993). "The Comprehensive TEX Archive Network (CTAN)." In: *TUGBoat* 14.3, pp. 342–351.

Gülden, Svenja A, Celia Krause, and Ursula Verhoeven (2020). "Digital palaeography of hieratic." In: *The Oxford Handbook of Egyptian Epigraphy and Paleography*. Oxford: Oxford University Press.

Herring, John et al. (2011). "Opengis® implementation standard for geographic information-simple feature access-part 1: Common architecture [corrigendum]." In.

Homburg, Timo (2019). "PaleoCodage—A machine-readable way to describe cuneiform characters paleographically." In: *Proceedings of the Dig-*

ital Humanities Conference 2019 (DH2019), Utrecht, the Netherlands 9-12 July, 2019. Utrecht, Netherlands.

Homburg, Timo (2020). "Towards Paleographic Linked Open Data (PLOD): A general vocabulary to describe paleographic features." In: *15th Annual International Conference of the Alliance of Digital Humanities Organizations, DH 2020, Ottawa, Canada, July 20-25, 2020, Conference Abstracts*. Ed. by Laura Estill and Jennifer Guiliano.

——— (2021). "PaleoCodage—Enhancing machine-readable cuneiform descriptions using a machine-readable paleographic encoding." In: *Digital Scholarship in the Humanities* 36, pp. ii127–ii154.

Homburg, Timo et al. (2022). "Annotated 3D-Models of Cuneiform Tablets." In: *Journal of Open Archaeology Data* 10.4. ISSN: 2049-1565.

Kamholz, David, Jonathan Pool, and Susan M Colowick (2014). "PanLex: Building a Resource for Panlingual Lexical Translation." In: *LREC*, pp. 3145–3150.

Khan, Anas Fahad (2018). "Towards the Representation of Etymological Data on the Semantic Web." In: *Information* 9.12.

Labat, René (1995). *Manuel d'épigraphie akkadienne. Signes. Syllabaire, Idéogrammes*. Paris: Geuthner.

Liang, Xiaohong (2021). "An exploratory survey of the graphic variants used in Japan: Part two." In: *Journal of Chinese Writing Systems* 5.2, pp. 115–124.

Liddell, Scott K. et al. (2003). *Grammar, gesture, and meaning in American Sign Language*. Cambridge University Press.

Mara, Hubert and Bartosz Bogacz (2019). "Breaking the code on broken tablets: The learning challenge for annotated cuneiform script in normalized 2d and 3d datasets." In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, pp. 148–153.

Mara, Hubert and Timo Homburg (2023). "MaiCuBeDa Hilprecht—Mainz Cuneiform Benchmark Dataset for the Hilprecht Collection." https://heidata.uni-heidelberg.de/dataset.xhtml?persistentId=doi:10.11588/data/QSNIQ2.

McCrae, John P et al. (2017). "The Ontolex-Lemon model: development and applications." In: *Proceedings of eLex 2017 conference*, pp. 19–21.

Mousavi, Seyed Muhammad Hossein and Vyacheslav Lyashenko (2017). "Extracting old Persian cuneiform font out of noisy images (handwritten or inscription)." In: *2017 10th Iranian Conference on Machine Vision and Image Processing (MVIP)*. IEEE, pp. 241–246.

Murano, Francesca and Achille Felicetti (2021). "CRMtex-An Ontological Model for Ancient Textual Entities." In: *Decimo convegno annuale dell'Associazione per l'Informatica Umanistica e la Cultura Digitale, Pisa*.

Navigli, Roberto and Simone Paolo Ponzetto (2012). "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network." In: *Artificial Intelligence* 193, pp. 217–250.

Panayotov, Strahil V. (2015). "The Gottstein System Implemented on a Digital Middle and Neo-Assyrian Palaeography." In: *CDLN, London*.

Píška, Karel (1999). "Fonts for Neo-Assyrian Cuneiform." In: *Proceedings of the EuroTEX'99 Conference*. GUST, pp. 20–24.

———— (2008). "Creating cuneiform fonts with MetaType1 and FontForge." In: *TUGboat* 29, pp. 421–425.

Rattenborg, Rune (2019). "Cuneiform Site Index (CSI): A Gazetteer of Findspots for Cuneiform Texts in the Eastern Mediterranean and the Middle East." https://ancientworldonline.blogspot.com/2019/12/cuneiform-site-index-csi-gazetteer-of.html.

Sanderson, Robert, Paolo Ciccarese, and Herbert Van de Sompel (2013). "Designing the W3C open annotation data model." In: *Proceedings of the 5th Annual ACM Web Science Conference*, pp. 366–375.

Seaborne, Andy and Steven Harris (2013). *SPARQL 1.1 Query Language*. W3C Recommendation. W3C.

Sporny, Manu et al. (2014). "W3C recommendation JSON-LD 1.0."

Stokes, Peter A. (2015). "Digital approaches to paleography and book history: some challenges, present and future." In: *Frontiers in Digital Humanities* 2, p. 5.

Taft, Marcus and Kevin Chung (1999). "Using radicals in teaching Chinese characters to second language learners." In: *Psychologia* 42.4, pp. 243–251.

Taylor, Jon (2014). "Wedge order in cuneiform: A preliminary survey." In: *Proceedings of the 60e Rencontre Assyriologique Internationale, Warsaw*, pp. 1–30.

Tinney, Steve and Eleanor Robson (2014). "Oracc: The open richly annotated cuneiform corpus." http://oracc.museum.upenn.edu/doc/search/index.html.

Toledo, Sivan and Zvika Rosenberg (2003). "Experience with OpenType Font Production." In: *TUGBoat* 24.3, pp. 557–568.

Vrandečić, Denny and Markus Krötzsch (2014). "Wikidata: a free collaborative knowledgebase." In: *Communications of the ACM* 57.10, pp. 78–85.

Yujian, Li and Liu Bo (2007). "A normalized Levenshtein distance metric." In: *IEEE transactions on pattern analysis and machine intelligence* 29.6, pp. 1091–1095.

Zesch, Torsten, Christof Müller, and Iryna Gurevych (2008). "Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary." In: *LREC*. Vol. 8. 2008, pp. 1646–1652.