**CHAPTER 9**

**On the Problem of the Epigraphic Interoperability of Digitized Texts of the Mediterranean and Near Eastern Regions from the First Millennium BCE**

*Doğu Kaan Eraslan*

**Introduction**

Recently there have been impressive efforts to digitalize ancient texts of the first millennium BCE. Their digitalization has not only facilitated access to the information contained in these documents, but it has also given us a chance to use computational methods to analyze them. The question of how projects that digitize these texts interact with each other, however, has not been a domain of interest for most researchers. Yet for those who are trying to understand the history of international relations in the ancient Mediterranean

---

1 The present communication has benefited tremendously from the suggestions and discussions that took place at the Caf'E.PHE (École Pratique des Hautes Études, Paris) during January and February of 2017 (Eraslan 2017a). I would like to express my gratitude to all of the Caf'E.PHE attendees, noting especially the very insightful questions of Daniel Stockholm and also Émilie Pagé-Perron’s suggestion to add the section on practical concerns. My remarks on the use of OCR were also incorporated following discussion with Émilie Pagé-Perron. Furthermore, I was fortunate to hear Terhi Nurmikko-Fuller’s 2017 paper at Caf´E.PHE on Linked Data technologies. This motivated my section on CIDOC CRM. Marine Béranger, one of the few people who dared to encode hundreds of Akkadian letters in TEI, laid the foundations for the observations that led to this paper. In addition, I would like to express my gratitude to Vanessa Bigot Juloux for her unwavering encouragement and faith in me as a colleague. I am also very grateful to Amy Rebecca Gansell for her English-editing suggestions. Her particular attention to technical details with regard to terminology and general phrasing increased considerably the overall comprehensibility of the paper. Of course, the content is my own responsibility. Lastly, I would like to thank my father, Eyüp Eraslan, for his constant support.

2 For example, a library such as Classical Language Toolkit (cltk) (<http://cltk.org/>) [accessed May 6, 2017]) would not have been possible without the digitalization of these texts. See Johnson (2014, <https://doi.org/10.5281/zenodo.60021> [accessed May 6, 2017]).

3 This was probably due to the isolated nature of the disciplines, but it has become increasingly clear that this practice of isolation cannot endure. For a notable exception to this phenomenon, see Gippert (1999).
region during this period, it is necessary to work with a multilingual corpus of inscriptions. Multilingual researchers are impeded by the incompatibility of the various digitalizations.

This paper demonstrates one of the incompatibilities that arises from the current state of the encoding schemes. After a brief survey of the encoding schemes used by the major corpora online, I review the primary trends and the differences among them and highlight the problem of how to facilitate their interoperability. In the interest of moving toward a solution, I propose a shift in the way we look at the encoding schemes. Until the present, we have read the ancient texts and deemed what we read as worthy of conservation; however, current technology makes it possible to analyze what we see, rather than what we have read. Although linguistic differences may make the encoding of what we have read incompatible, the unit we employ to encode what we see promises a solution of compatibility, since all 2D shapes used by the scripts of the first millennium BCE can be expressed in the vector spaces of linear algebra.

---

4 Throughout this paper, I use the terms “encodings,” “schemes,” and “encoding schemes” interchangeably. In the most general sense, an encoding scheme is a set of rules that standardizes the representation of an object in a digital environment.

5 I use the terms “epigraphic interoperability” and “interoperability” to refer to the same phenomenon. Epigraphic interoperability refers to the ability to transform the encoding scheme of an encoded epigraphic phenomenon into another encoding scheme without losing data.

6 Although the scripts under analysis are all attested in 3D environments, we can at least work with their 2D models in a vector space without data loss. This is necessary because not all texts have been digitized in 3D. With the continual advances in 3D scanning technologies, however, an upcoming task for digital epigraphists will be the development of standard encoding schemes for 3D shapes. Vector space: in linear algebra, an additive group that is associated with a field of real numbers and permits vector addition and scalar multiplication. For more detailed information, including formulas, see Weisstein (n.d., <http://mathworld.wolfram.com/VectorSpace.html> [accessed February 26, 2018]). For additional information, see in this volume, Monroe (271), who used vector space for processing a Document Term Matrix (DTM) for a bag-of-words method. Vector: Mathematically speaking, a vector is anything that represents something that has a size and a direction. Size is the distance between the origin and end points, and direction is represented by the angle that a line makes with an axis. For a more technical definition, see Lang (1986, 9–12), who uses the term “located vector” for the concept we have defined above, although this appellation does not represent a general consensus. Vector addition simply entails adding two or more vectors together. Scalar multiplication is a multiplication operation between a vector and a number that belongs to the same set of numbers contained in the vector.
Toward this end I present some tools to enable digital epigraphers to surmount the problem of the current lack of multilingual interoperability. The solution offered here consists of the projection of signs from a physical medium to a vector space where they are mapped to their semantic units, facilitating the querying of the shape in the vector space. For encoding technology, my choice of EpiDoc, svgs, and Unicode (which are discussed later in this paper) is motivated by purely practical concerns, but I discuss alternatives as well. Finally, this paper provides some practical insights on how different components of the model can interact and describes what would be necessary to implement the suggested solution for an ongoing project.

Encodings of the First Millennium BCE: A Brief Survey

C(anonical)-ATF

C(anonical)-ATF is the backbone of the texts displayed by the Cuneiform Digital Library Initiative (CDLI), which is currently the major online database for cuneiform texts. As the name implies, CDLI conserves epigraphical and semantic data of cuneiform texts, rather than for a specific language. It uses American Standard Code for Information Interchange (ASCII) characters. Special characters such as “_” or “$” are reserved for indicating sign functions and information about the state of conservation of the script and its physical medium.

7 These tools are part of a possible solution; certainly, there may be additional approaches with different tools.

8 Sign corresponds to an elementary semantic unit comprising alphabetical and non-alphabetical languages. The concept of elementary semantic unit is defined more precisely later on. Physical medium refers to the object on which we observe the signs. It can be papyrus, a vase, a stone surface, etc.

9 Querying: extracting information from a database based on the user’s input.

10 For details regarding the specific procedures used in the encoding process, see Tinney (2017, <http://oracc.museum.upenn.edu/doc/help/editinginatf/cdliatf/> [accessed January 26, 2017]).

11 CDLI was initially only for Sumero-Akkadian cuneiform (Anderson et al. 2000, <http://pages.jh.edu/~dighamm/ice/icereport.html> [accessed April 4, 2017]). Regarding the “c” in “C-ATF,” it can stand for either “canonical” or “CDLI,” since the latter uses ATF encoding. For further information, see in this volume, Pagé-Perron, 198–200, 2001.28. For the representation of cuneiform characters in unicode, see Everson, Feuerheim, and Tinney (2004, <http://std.dkuug.dk/jtc1/sc2/wg2/docs/n2786.pdf> [accessed April 7, 2017]).

12 ASCII characters are limited to those that are used in the writing of English words without accents. The characters include but are not limited to: “æ, b, R, a, m, k, [, ;”.

Doğu Kaan Eraslan - 9789004375088
Downloaded from Brill.com04/26/2019 02:37:59PM via free access
cal medium. The minimal semantic unit of the encoding corresponds to the language’s minimal semantic unit. In most cases, the minimal semantic unit is a cuneiform sign.

Table 9.1  Elamite text in C-ATF

| CDLI no: P383090$^a$ = PF 404$^b$ | Hallock transliteration: 20 ZÍD.DA.lg kur-min$^c$ |
|-----------------------------------|-------------------------------------------------|
| C-ATF:                            | 2(u) _zi3-da-mesz_ kur-min$^2$d                  |

$^a$ <http://cdli.ucla.edu/search/archival_view.php?ObjectID=P383090> (accessed January 25, 2017).
$^b$ Hallock 1969, 163.
$^c$ Author’s translation: “20 flour supplied by.”
$^d$ The differences, such as zi3=ZÍD, are related to different readings/interpretations of the signs and fall outside the scope of this paper. One should note, however, the typological differences between representations of the sign readings.

---

13 Special characters are exemplified by “+, _, =,” as opposed to alphanumerical characters such as “a, b, 1, 2,” etc.
14 I do not use the term “semantic unit” in its strictly linguistic sense. By minimal semantic unit, I mean the smallest distinctly perceived element. This corresponds to the smallest delimited element in the case of an encoding scheme, since the perception is done by the machine.
**Bavant XML-Elamite Standard**

XML-Elamite standard, developed by Marc Bavant, is tailored for conserving Elamite texts along with their transliterations and translations. It is based on Unicode characters, and it follows widely used conventions in Elamite scholarship, making it human readable.

| Table 9.2 | Elamite text in Bavant-XML |
|-----------|---------------------------|
| DNA 1§1a  |                           |

```xml
<inscription name="DNA"><metadata><king dates="522-486" name="Darius I" /></metadata><source></source><url>http://www.um.es/ipoa/cuneiforme/elamita/archivosreales/dario1/dario_i_dna.htm</url><metadata><notes/><section id='DNA:\$1'>
  <T1>(1) d.na-ap ir-šá-ir-ra</T1>
  <T2>(1) Ahuramazda es el gran dios
        </T2>
  <T3>d.na-ap ir-šá-ir-ra</T3>
  <T4>nap iršara</T4></section></inscription>
```

a “Great god (Auramazda)” (Schmidt 1970, pl. 32).

**M(anuel) d(e) C(odage)**

Manuel de Codage (Mdc) is the standard developed for processing ancient Egyptian hieroglyphs. Mdc, specifically its last version, Mdc-88, has seen a lot of variants, and it uses ASCII characters. The most frequently used special characters are reserved for expressing hieroglyph positions, such as “*” for

---

15 The 2014 version of Bavant’s Elamite-XML corpus is accessible through the University of Amsterdam’s Digital Academic Repository: [http://hdl.handle.net/11245/1.348271](http://hdl.handle.net/11245/1.348271) (accessed February 13, 2017).

16 Elamite: an ancient language used in Elam, a civilization existing from the third into the mid-first millennium BCE based in the southwest region of the Iranian plateau, just northeast of the Persian Gulf (in the modern provinces of Ilam and Khuzestan).

17 Human readable: a text that is destined to be read by humans, as opposed to machine readable, which means it is intended to be processed by machines.

18 Buurman et al. 1988. Note that, in theory Mdc can be extended to process Hieratic, an ancient Egyptian cursive script, as well. See also the critical overview by Nederhof (2013).
horizontal grouping and “:” for vertical grouping. Less frequently used special characters are employed for indicating syntactic interpretations, such as “_” for word ending.

Table 9.3 Ancient Egyptian text in \textit{mdc-88}

| Normal Transliteration: | \[—\] (.w) jn.n ḫr tw nḥ.t(?) pn rd(i) ḫr twa |
|------------------------|--------------------------------------------------|
| \textit{mdc}: | 4 -[-+b...+s-]_-.:3_-W25:N35_-Aa1:D21_-X1-G43_- |
| | G21” (?)”_-X1_-Q3:N35_-D21:D36*Z1_-Aa1:D21_-X1:G43_- ! |

a “One must bring this prayer, one must give ...” Here there is an irregularity between the demonstrative adjective and the object it governs. We would expect "tn" instead of "pn." The uncertainty of the surrounding "G21" also suggests a word other than "nḥ.t."

\textbf{CAL Code}

\textit{CAL} Code has been adopted by the Comprehensive Aramaic Lexicon (\textit{CAL}) project.\textsuperscript{19} It is used for storing Aramaic texts from the ninth century BCE to the thirteenth century CE.\textsuperscript{20} \textit{CAL} Code was developed in the 1980s. It has changed considerably with the arrival of the internet and Unicode, and it remains effective in giving a better understanding of displayed text.\textsuperscript{21} \textit{CAL} Code uses ASCII characters and was designed so that Aramaic scholars could use it without intensive training. Special characters do not solely express a particular semantic or syntactical field of the language. Some, for example, express phonetic values or diacritics.\textsuperscript{22}

\textsuperscript{19} <http://cal.huc.edu/> (accessed February 28, 2017).

\textsuperscript{20} Aramaic is an ancient Northwest Semitic language attested throughout eastern Turkey, northwest Iran, Iraq, Syria, the Levant, and parts of Arabia at the beginning of the first millennium BCE and onwards. Its descendants, such as Mandaic, are still spoken today.

\textsuperscript{21} Kaufman 1987, <http://cal.cn.huc.edu/pdfs/CalManualIntro.pdf> (accessed April 7, 2017).

\textsuperscript{22} On phonetic values, for example, "(" represents the phonetic value “ayin,” and "\$" represents the phonetic value “shin.” For examples of diacritics, "_" indicates a dot below a letter, and ";" indicates a double dot above a letter.
Table 9.4 Imperial Aramaic text in CAL Code

Hebrew Transliteration: יָשָׁב אֱלֹהִים אֱלֹהִים תִּשְׁבָּה קִרְיָת אֱלֹהִים [k]sP g?[br]y zy?[ (byd l byt mlk)

EpiDoc

Strictly speaking, EpiDoc is an XML-scheme for adding markup to ancient texts. Thus, it is applicable to all ancient languages, but it is predominantly used for Graeco-Roman corpora. It employs the Text Encoding Initiative (TEI) scheme as its backbone, and it accepts anything encoded with UTF-8. Since EpiDoc is fundamentally a markup language that tries to be applicable to a broad range of languages, it is fairly liberal when it comes to using values for attributes. A significant issue with EpiDoc, however, is that any text encoded with it is dependent on the project’s financial capacity to maintain itself long-term. Increased costs are due to the high overhead costs for the server and the need to retain a project staff person who knows how different versions of the schema map to each other in a particular project. One potential way to at

---

23 Elliot et al. 2017, <http://www.stoa.org/epidoc/gl/latest/> (accessed June 1, 2017).
24 Text Encoding Initiative (TEI): a consortium that maintains the scheme for evaluating XML-encoded documents. For the history of the consortium, see TEI Consortium (2018, <http://www.tei-c.org/release/doc/tei-p5-doc/en/Guidelines.pdf> [accessed May 15, 2017], xxv). For further explanation, see in this volume, Bigot Juloux, 164, 164n65.
25 "UTF-8" stands for Unicode Transformation Format, with “8” referring to the use of 8-bit (a numerical value that equals 0 or 1) sequences to represent a character. It is a method for encoding Unicode characters. There are also UTF-16 and UTF-32, which have, respectively, 16- and 32-bit character sequences.
26 Markup language: “a set of markup conventions used together for encoding texts. A markup language must specify how markup is to be distinguished from text, what markup is allowed, what markup is required, and what the markup means” (TEI Consortium 2018, <http://www.tei-c.org/release/doc/tei-p5-doc/en/Guidelines.pdf> [accessed May 15, 2017], xxv). For further explanation in this volume, see Bigot Juloux (163, 165–166, 186) regarding creating analytical taxonomies.
27 The cost is due to a specific situation with XML. During the encoding process, XML requires one to end a markup section with the tag that is used at the beginning of the section, making one repeat information in an inefficient way, since one would be using extra characters for information that does not necessarily need those characters. The gain is that this
least get around the server issue would be to make a detailed database of encoding projects, so that already encoded material can be scraped from the internet, which would eliminate the need to redo the whole process. The database should also provide an application programming interface (or API, a set of rules and tools that defines how computers can interact) for accessing it and scripts to add the markup to the source-text material based on the scraped material. The use of scripts in the automatic conversion of one database to another, however, could create what are known as cross-walk problems.

### Table 9.5  Ancient Greek text in EpiDoc-XML

| Απόλλω-νεϊ θεῷa | <div type="edition"><head>edition</head> <ab><lb xml:id="line-1" n="1"/> <persName type="divine"> <name nymRef="Ἀπόλλων">Ἀπόλλω-<lb xml:id="line-2" n="2"/> νεϊ</name> <w lemma="θεός">θεῷ</w> </ab> </div> |
|----------------|--------------------------------------------------------------------------------------------------|
| Normal EpiDoc  | EpiDoc                                                                                                                                 |

a The inscription reads “To the god Apollo;” see Thonemann et al. (2012, <http://mama.csad.ox.ac.uk/monuments/MAMA-XI-314.html> [accessed January 26, 2017]).

### Major Trends and Differences

There are two major trends in all of these encoding schemes:

- makes XML very human readable; the cost is the extra transactions between the client and the server.
- Scraping: a process of selectively retrieving online data using special software. For further discussion of “scraping,” see in this volume, Pagé-Perron 204, 204n41.
- See in this volume, Prosser, 318, who describes OCHRE API.
- Cross-walk problems occur when you want to move the content of a database to another database that is arranged in a different manner. One knows that the content is essentially the same, but different representations of it make it difficult to transfer content from one to another.
1. Minimal distinctly perceived units are mapped to intermediary elements.\(^{31}\)
2. Glyphs are treated as characters/letters.\(^{32}\)

The key difference among schemes is the nature of the intermediary elements. Cal Code, C-ATF, Bavant-XML, and EpiDoc use semantic elements in a manner that is similar to a transliteration, and MdC uses arbitrary codes. Two problems result:

1. Apart from MdC, the encoded texts are conceived so that the majority of the readers of the texts are humans.
2. Some schemes attribute glyphic properties to their elements, and some do not. Therefore, the resulting encoded document for multilingual corpora has the problem of a lack of interoperability.\(^{33}\)

The first problem concerns the design of the encoding schemes, and it probably cannot be remedied quickly. In sum, a document that is digitally conserved as it is observed by humans suffers from the imposition of a hierarchy by the human perception of the document. The semantic hierarchy imposed by the observer creates substantial differences in the retention rate of the data held by the original document. This means that since we wanted to conserve what the text said, we conceived systems that conserve in detail how that which is said is expressed but give less attention to the actual state of the document in

---

\(^{31}\) Minimal distinctly perceived units: units that could correspond to letters for alphabetical languages or signs for logo-syllabic languages. Intermediary elements: the vehicle between what we see in the physical medium and what we understand. This corresponds, for example, to the alpha-numeric characters that are employed to represent the glyph in the physical medium.

\(^{32}\) Glyphs: the most basic parts of the visual representation of an elementary semantic unit. In the case of an Egyptian hieroglyph, for example, a glyph could be a sitting man or a bird. For cuneiform, a glyph would be a horizontal, vertical, or diagonal line comprising a cuneiform sign, and for alphabetic languages, such as Greek and Aramaic, a glyph corresponds to a visual representation of a letter, such as an alpha.

\(^{33}\) For more technically oriented reader, the problem can be summarized as a class inheritance problem. If we take the elementary semantic unit as the basic class of architecture, which would be the encoding scheme, the above encoding schemes instantiate this first basic class with different attributes in their constructor, making it difficult do transformations later on. The logical solution would be to add the lacking attributes to the constructor, but this still would not enable us to transform the already encoded material, since what we want to express with the attributes is defined with methods later than at the current state of these encoding schemes.
question.\textsuperscript{34} Imposing the semantic hierarchy of the human observer therefore makes encoded documents less computable because it reduces our capacity to compute non-semantic data by presenting it, if at all, in a less precise and less organized manner. Thus, arranging encoding schemes on the premise that the majority of readers would be humans is a design flaw that cannot be addressed easily. The present paper deals with a subset of the problems that are generated by this flaw.

\textbf{The Problem of Achieving Interoperability}

The problem with achieving interoperability comes from the fact that the elementary unit of the above encoding schemes, corresponding to a minimal semantic unit in the relative language, does not have the same visual decomposability throughout the above-mentioned languages, making the units unreliable in representing the state of conservation of a multilingual document. This is the result of the human-oriented approach, which relies on characters rather than glyphs as the basis of encoding.

Consider, for example, the quadrilingual vase of Darius I.\textsuperscript{35} It bears four inscriptions, which are written with Elamite cuneiform, Akkadian cuneiform, Egyptian hieroglyphs, and Old Persian cuneiform. A fracture has damaged three of the four inscriptions. Each inscription uses more or less the same formula for venerating the king. In this case, to represent the texts on the vase, one would need to resort to at least three different encoding schemes.\textsuperscript{36} What if there were a thousand vases like this, and a researcher was trying to observe the extent to which the king’s name was preserved across all of the inscriptions on all of the vases?\textsuperscript{37} Such a scenario would require three steps:

1. extracting the king’s name from the documents
2. assessing whether the king’s name is damaged or not

\textsuperscript{34} For instance, a dot above an Aramaic word in \textit{cal} does not mean that there is a damage at that point of the sign; instead it means that the sign is damaged, so the reading can be doubtful.

\textsuperscript{35} Darius I ruled the Achaemenid Empire from 521 to 486 BCE. For the publication of the vase, see Stolper and Goodnick (2002).

\textsuperscript{36} In this case, the encoding schemes would be \textit{c-atf}, Unicode (for Old Persian), and \textit{mdc-88}.

\textsuperscript{37} We are assuming here that the texts address the king with the same name, but in different languages and thus with slightly different phonetic variations.
The first assumption holds that each encoded minimal distinct unit regularly refers to its correspondent (that is, there is a one-to-one correspondence between each encoded unit and its correspondent).
between the encoded distinct unit and its correspondent). The second assumption holds that all the encoded distinct units refer to their correspondents in the same manner. The first assumption is false. The second one is also not true, but it requires a little more attention.

42 The terms minimal distinctly perceived unit and minimal distinct unit are interchangeable.
43 The proof of this statement lies in the comparison of the encoded documents to their real correspondents; it is as if we are trying to prove that the ocean is made up of water. The following comparisons can also help the computer scientist understand the extent to which the specialist in ancient history considers the character of their work to be the same. From there, the computer scientist can get an idea about how much the encodings represent the actual documents. To observe the difference between the encoded document and its correspondent, the reader can refer to the following documents as examples: For MdC, compare the photograph of Caire O.25338 with the digital version distributed with Jsesh. Specifically, the first two signs of "V30" and "nb," in the eighth column from the right in the plate, and the seventh line from the top in the digital version. In the photograph the first one is a closed shape, whereas the second one is clearly not a closed shape. However, it is without a doubt that both signs represented the same semantic value; thus, they were encoded with the same character, "V30." See Daressy (1901, pl. 60, no. 25338). For C-ATF, compare the photograph of P480793 to its transliteration. At the second line of the first column, the last two "a"s are encoded in the same manner. In the photograph, one sees that the last "a" is actually slightly damaged on the superior side, but this does not change the reading of sign because one can count all of the elements of the sign. Nevertheless, it is clear that the "a"s do not represent the same thing here. For P480793, see <http://cdli.ucla.edu/search/archival_view.php?ObjectID=P480793> (accessed January 28, 2017). For CAL Code, compare the hand drawing of the TAD C3.7 with its encoding: <http://cal.cn.huc.edu/get_a_chapter.php?file=23350> (accessed November 4, 2016). See the "n" of the "ryqnn" in Column E., Recto Line 6, and the "n" of the "mnnt": the first one clearly has a longer superior part, but it is not visible in the encoding. Actually, the last "n" of the "ryqnn" is encoded as "ryqnnN," with the "N" corresponding to word final form of the "n" character. One can see that this does not go well with the hand drawing given in the publication, because, in that position, the hand drawing presents a normal form of the "n" character, as opposed to the word final form. For the hand drawing, see Yardeni (1994, fig. 1). For EpiDoc, compare the photography of the TM 25656 with its encoding. Compare the omegas of the Apollon, the omega of the toi in the first line, and the omega at the end of the fourteenth line. In terms of the encoding and interpretation, they are all omegas. However, the photograph shows that the "t" of the toi touches the omega, and that there is clearly a space in the middle of the sign at the final omega on the fourteenth line. For the encoding, see Duke Databank of Documentary Papyri (<http://papyri.info/ddbdp/bgu1;249> [accessed March 24, 2017]). For the photo, see the online catalogue of the Ägyptisches Museum und Papyrussammlung, Staatliche Museen zu Berlin (<http://berlpap.smb.museum/02131/> [accessed March 24, 2017]). Since the Bavant-XML standard directly follows the transliteration conventions in Elamite, there is no point in discussing
The second assumption claims that encoding schemes admit the same attributes for their elementary unit. On the contrary, there is not homogeneity among the types of the input value in the normalization process. For example, consider, hypothetically, that one wants to represent the damage on a certain “na” sign in an Elamite inscription. The bottom part of the vertical wedge is damaged. This information is impossible to express in C-ATF, EpiDoc, Bavant-XML, and CAL Code. MdC does, however, provide an indirect way to express this information with its "\shading" attribute, through which one can pinpoint which part of a sign is damaged. In its current stage of development, the "\shading" divides the sign to four sectors, but, with a little extension of its syntax, we can use it to pinpoint the damage. Therefore, we can say that there is a substantial difference between how the “na” sign is conceived throughout the encoding schemes. For C-ATF, EpiDoc, Bavant-XML, and CAL Code, the sign would be considered as a monolithic block, but for MdC, a sign can be considered in quadrants. Hence, there would not, and cannot, be homogeneity in the initial input types.

The empirical refutation of the first assumption, combined with the logical refutation of the second one, shows us that the premises required by the normalization process are indeed impossible, impeding the normalization process itself. Thus, the third step required by an epigraphically reasonable query of an ancient multilingual document is not possible. This proves effectively that the current status quo maintained by the providers of these encoding schemes, if it rests unchallenged, does not, cannot, and will not support a viable way to interact with multilingual documents on an epigraphic plane.
(A Possible Part of a) Solution

The only real challenge for ensuring epigraphic interoperability is to find a solution to the normalization problem discussed above. There are two possible solutions:

1. Extend the syntax of the encoding schemes so that they treat signs as a set of sectors, as they are treated by Mdc.
2. Encode visual phenomena rather than semantic units, then map the encoding to the semantic unit.

Both of these solutions would work for achieving homogeneity at the input level of the normalization process. However, due to the practical implications of the extension of the syntaxes, the second option—the encoding of the visual phenomena—is a more viable option in the long run.

What would the extension of the syntaxes of the encoding schemes imply? It would imply more precision regarding the state of conservation of the sign. How could one attribute a value to the extended syntax? Since such a value has not been generated before, one would have to go back to a photograph or facsimile to record the value in question. This would entail a lot of repetitive manual work, but it is possible to write a program so that the computer can do the heavy lifting.

Optical Character Recognition (OCR) has become a viable option to use in epigraphy largely due to the popularization of the deep-learning algorithms. The passage, however, between the raster image and the vector graphic, which can be stored in encodings, is actually quite tricky. The difference between the raster image and a vector graphic is that a raster image is a made up of a matrix of numerical data for the computer called pixels, whereas the vector image

---

49 Extending the syntax: understood here as adding additional features to the encoding scheme.

50 OCR: a subfield of computer vision that permits machines to extract text characters from a pixel-based image. The overall structure of the form recognition is concisely described by Cheriet et al. (2007, 6–7). See in this volume, Prosser, 322–323. For other practical applications of computer vision, see Dawson-Howe (2014), and, more recently Peters (2017).

51 Deep Learning: a subfield of machine learning. It is characterized by the breakdown of complex representations into a hierarchy of simple components during the learning process. For a concise introduction into its theory and practice, see Goodfellow, Bengio, and Courville (2016, 1–8).
comprises a group of lines passing through points defined in a coordinate system, that is, a set of equations instantiated by a set of points. Roughly speaking, the vectorization process of a raster image and the identification of it as a character address two related goals. The latter is relevant to our discussion because the languages of some of our texts are written without dividers between words. One cannot understand where one sign ends and another begins without recognizing both signs at first. Thus, in order to isolate the signs for the purpose of recording a value to the extended syntax, some sort of OCR-based method is necessary.

Since an OCR-based approach is inevitable, even in the simple case of extending the syntaxes, there is no reason to content ourselves only with recording values to an extended encoding syntax. This would neglect a lot of the OCR data, most notably the local visual features that are associated with the provenance of the script. The OCR process would actually generate a features’ vector that could be compared against the abstract vector of the sign. If the comparison gives a positive result, then the image from which the features’ vector is created is assigned (or mapped) to the sign in

---

52 The scope of the current paper does not permit further discussion of the application of OCR methods to ancient languages treated through the encoding schemes, but OCR methods remain vital for the study of previously encoded material.

53 Examples of local features include the handwriting of individual scribes or small, regional, or historical variants in the appearance of certain signs.

54 Since documents written in ancient languages are not reproducible products like fonts, we dismiss the matrix-based approach used in early OCR technologies. Features’ vector is a convenient term for the mathematical construct that contains the patterns of the image of the sign. Technically, depending on the architecture one is using, one can generate anything between a tensor and a vector as the output of the process. An abstract vector in this case refers to practically any representation of a sign that can be found in a sign list or a dictionary. These are called “abstract” because no sign in its native form would correspond to it 100%. The comparison would also implicitly require that the image of the sign in the sign list be treated by the same architecture that treats other images of the sign in order to generate the abstract vector. For a typical authoritative sign list, see Borger (2004).

55 In the context of a dictionary-data structure, to map is to assign a value to a key. Dictionary: a type of abstract data structure (also known as an associative array or symbol table) used in programming languages, implemented under the name dictionary in Python. A dictionary stores information as key value pairs. Use of dictionaries, at least in the context of Python, implies that we would have a non-negligible performance gain for operations in which we want to process a value by using its associated key. For a more detailed explanation, see the Python tutorial: <https://docs.python.org/3.6/tutorial/datastructures.html> (accessed March 24, 2017).
question.\textsuperscript{56} Thus, the comparison of the features’ vector derived from the input image can be used against the abstract vector of the sign in order to map the input image to the sign.

In the case of previously encoded documents, one could take the generated features’ vector that would be mapped to the encoded sign and store it along with the encoded documents. This summarizes the second option (to encode visual phenomena rather than semantic units, then map the encoding to the semantic unit). Its viability boils down to the following:

1. Can a features’ vector accurately represent the sign in the photograph?
2. Can a features’ vector be converted to an encoding scheme of the type mentioned here?
3. How does a previously encoded document’s scheme interact with the features’ vector?
4. What tools can one use to accomplish this task?

The answer to the first question is that it depends on the expected accuracy. Of course, 100\% accuracy would not be possible, but somewhere around 90–95\% may be expected.\textsuperscript{57} Although not all scholars agree that methods that provide less than 100\% accuracy are acceptable, we should also consider the sheer quantity of the inscriptions. Manual intervention, which may seem like it would guarantee 100\% accuracy, can also introduce errors. However, we strive for perfection in our manual and digital efforts, and certainly increasing the precision of the OCR results is an important objective.\textsuperscript{58}

The answer to the second question is EpiDoc. The problem is that the encoding schemes we have discussed up to this point were not designed to support machine-to-machine interactions; rather, except for Mdc, they were mostly concerned with human-to-machine interactions. This resulted in the

\textsuperscript{56} A note to the technical reader: I am simply describing very broadly a supervised learning process in which the ground truth is produced from the entries of sign lists.

\textsuperscript{57} Holley 2009, \url{http://www.dlib.org/dlib/march09/holley/03holley.html} (accessed April 4, 2017). This might be confusing to some. OCR technology, like all deep-learning- and machine-learning-based technology, reasons based on probabilities. Consequently, there is always a risk of having a false positive as the result of the process even if that risk is very low in terms of probability. Since the image from which the features’ vector is created might not be mapped to the right sign, there is, again, a very small risk that the features’ vector might not represent the sign contained in the image.

\textsuperscript{58} The most viable option for increasing precision may be to allow public users to edit and save OCR-results, see Holley (2009, \url{http://www.dlib.org/dlib/march09/holley/03holley.html}) [accessed April 4, 2017]).
use of encoding schemes that are not compatible with the standard data serialization formats, such as JSON,\textsuperscript{59} XDR,\textsuperscript{60} and XML, used in computer science. This, in turn, has hindered our computational capacity with regard to encoded documents. It is also true that at the time these encoding schemes were produced, the data serialization formats mentioned above did not exist,\textsuperscript{61} with the exception of XDR. EpiDoc is the only encoding scheme that uses a data serialization format as its basis.\textsuperscript{62} This means that if a features’ vector can be serialized into XML format, it can be integrated into EpiDoc with relative ease. Furthermore, it can be serialized.\textsuperscript{63}

The answer to the third question (How does a previously encoded document’s scheme interact with the features’ vector?) is that previously encoded documents can provide the semantic content that the features’ vector would be assigned. The real question is whether or not the features’ vector and the semantic content would be stored in the same document. For the reasons stated above, storing both in the same document is only possible for documents encoded with EpiDoc. A viable approach that would not change the structure of the encoding scheme itself, but would serve as a markup, would be to use

\textsuperscript{59} JSON stands for JavaScript Object Notation. It is a serialization format, meaning that it allows us to save the state of our data. JSON is well standardized and well supported as an interchange format between different platforms.

\textsuperscript{60} XDR stands for External Data Representation. Like JSON, it is a standardized serialization format.

\textsuperscript{61} Data serialization: the process that enables us to save the data we are working on so that it can be reconstructed just as we had saved it in different platforms.

\textsuperscript{62} My experience working with around 30,000 texts encoded in EpiDoc has revealed a number of shortcomings of the format, mostly due to the changes in the scheme. Nevertheless, EpiDoc is the only encoding scheme that is based on a data serialization format, so we must work with it.

\textsuperscript{63} As I have stated, the features’ vector is a mathematical construct, and representing mathematical constructs like matrices and tensors inside XML has a result that is quite verbose. JSON would be a much better choice here. The structure of the content of the serialized data, however, is not a trivial subject. I discuss this further in the “Serialization and Epigraphic Interoperability” section, below. Note that a features’ vector cannot be directly included in an EpiDoc document due to the restrictions of the schema. It can be included if one modifies its schema.
linked data technologies based on ontologies, such as CIDOC CRM. CIDOC CRM, for example, presents classes, such as: class Physical Man-Made Thing, to which would belong the medium of the inscription; class Visual Item, to which would belong the picture comprising the general view of the medium; class Mark, to which would belong a features’ vector of a particular sign; class Image, to which would belong the vector graphic of the sign; and finally class Inscription, to which would belong the sign “transcription/reading.” This leads to our next question: are there any tools we can use to encode visual phenomena rather than the semantic unit? Here I discuss Unicode and encoding schemes, EpiDoc, and Scalable Vector Graphics (SVG).

**Unicode and Encoding Schemes**

Unicode is a system to map characters to bytes through code points. Characters are treated as semantic units, and code points are arbitrary constructions. The mapping requires two different reversible procedures that make the system rather complex. Most modern platforms use a Unicode-based encoding

---

64 For “Linked Data” (LD) as “a set of techniques for the publication of data on the Web using standard formats and interfaces,” see Wood et al. (2014, 5). For further information in this volume, see Nurmikko-Fuller, 344, and see Matskevich and Sharon, 46–47, for practical examples in Archaeology. Ontology: a term borrowed from Philosophy referring to Conceptual Reference Models (CRM), which are models that explain what things are in the digital world and how an object relates itself to the digital world. For additional information, see in this volume, Nurmikko-Fuller, 343, 347–349.

65 CIDOC is the International Committee for Documentation of the International Council of Museums. The CIDOC Conceptual Reference Model (CRM) “provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in cultural heritage documentation” (<http://www.cidoc-crm.org/> [accessed March 30, 2017]). See also Le Boeuf et al. (2015, <http://cidoc-crm.org/Version/version-6.2> [accessed March 30, 2017]). An interesting epigraphic application of CIDOC CRM ontology is found in Felicetti et al. (2016, <https://pdfs.semanticscholar.org/5ac4/f10e5f46b52f6f4a0eb3d6480408017ca.pdf> [accessed March 30, 2017]). For further discussions and evaluations related to the use of CIDOC CRM, see in this volume, Matskevich and Sharon, 45, on material heritage and field archaeology recording; Nurmikko-Fuller, 353–355, on the web publication of Sumerian texts.

66 Since the empirical observations that led to this paper were based on the provided encoding schemes and sign drawings in svg format, not on the technologies that were recently developed to link them, the question of properties remains to be explored in a future project.

67 For more technical information, see Whistler, Davis, and Freytag (2008, <https://www.unicode.org/reports/tr17/> [accessed January 20, 2017]).
scheme since it was designed to support every character used in every language.\(^68\)

Both Unicode and the encoding schemes covered above can be used as stable class identifiers for the glyphs. With encoding schemes, we can use the already encoded semantic unit as an identifier. I, however, do not recommend this option because sometimes the same signs can have different readings, and they could acquire other readings in the future. Nonetheless, since Unicode code points do not actually cover all of the signs encoded by these languages,\(^69\) encoded readings can be used for class identifiers while waiting for more robust Unicode coverage.

**EpiDoc**

EpiDoc is, as stated above, the only encoding scheme that is based on a data-serialization format. The abstract nature of the markup language lends itself well to the multilingual document. With a little parsing, we can pursue syntactic influences among contemporary languages and extract phonetic data from different languages for a relative lemma,\(^70\) such as a king’s name. EpiDoc generally has all the advantages and disadvantages of using an XML-based encoding scheme.

The disadvantages of using EpiDoc include:

1. EpiDoc’s hierarchy of the elements is sometimes too strict to accurately express what is happening. This is especially troublesome when one is working on texts that are copies of another text in different language,

---

\(^68\) This includes languages such as UTF-8 and UTF-16.

\(^69\) This is most notable for ancient Egyptian, for example.

\(^70\) Syntactic influence: the observation of the order of linguistic elements in a sentence of one language in a sentence of another, linguistically unrelated, language. This is exemplified by the formula “x-š-a-y-š-i-y : x-š-a-y-š-i-y-a-n-a-m,” “king of kings,” in Old Persian royal inscriptions. Normally in Old Persian, the genitive precedes its object “(p-á-r-s-h-y-a : p-u-ç)” [Xph line 12], son [p-u-ç] of Persian [p-á-r-s-h-y-a], but in Akkadian the genitive follows its object, as in “šarrū māti,” meaning “kings of the land.” Hence, with the genitive “x-š-a-y-š-i-y-a-n-a-m” following its object “x-š-a-y-š-i-y,” as in the Akkadian example, the occurrence of the genitive “māti” after its object “šarrū” represents syntactic influence. Old Persian is an Indo-Iranian language descended from Old Iranian. It was used mostly, but not exclusively, in royal inscriptions in the Achaemenid Empire (550–330 BCE). Lemma: the dictionary form of an attested word.
such as those on the quadrilingual vase or Behistun inscription of Darius.\textsuperscript{71}

2. To ensure a reliable result, EpiDoc requires a lot of planning on the namespace and tag usage.\textsuperscript{72} Without sufficient planning on this level, especially regarding the values given to the attributes of the elements, encoding would present an inconsistent representation of the document.\textsuperscript{73}

3. As a scheme for recording epigraphic phenomena, EpiDoc is rather limited in terms of the number of attributes that can be used for recording linguistic phenomena. For example, consider that we want to encode the word “λυθήσομαι” in first person singular, future tense, indicative passive.\textsuperscript{74} To store the information, we would use the $<$\texttt{w}$>$ element in EpiDoc.\textsuperscript{75} TEI, the main scheme in which EpiDoc is based, supports more attributes

\textsuperscript{71} One case that truly exemplifies this dilemma is that of the Behistun inscription (522–486 BCE). The traditional division of the inscription follows the Old Persian text, but the Elamite version of the same inscription does not coincide exactly with the divisions of the Old Persian version. See King and Thompson (1907); Shahbazi (2012, <http://www.irancanonline.org/articles/darius-iii> [accessed April 2, 2017]).

\textsuperscript{72} Namespace refers to the appellation of the tags with relation to the information they convey. There could be several options in determining which tag should be used for a personal name, such as John Doe Smith. We could, for example, decide to use “name,” “personalName,” or both. Which appellation better frames the interpretation of the name depends on the context, and, thus, on the project, although it should be kept in mind that the more interpretation one provides, the more time one must invest in constructing a coherent namespace. Among other examples: 1) When one uses an element and attribute in TEI (see in this volume, Bigot Juloux, 165), one must refer to TEI namespace URL. 2) There are URIs that are namespace names in XML (see also in this volume, Nurmikko-Fuller, 338–339, who describes them). The appellation of tags filling the role of markups, as we have seen in the case of XML documents, must be decided in the process of developing the namespace. Deciding what a tag should be called, however, does not necessarily result in its coherent usage. Taking the example of the name John Doe Smith, we would not know whether the name belongs to a real person or a fictional character. If we were to distinguish this aspect of the names in a document, the distinction would not come from the domain of the namespace, since it is possible for the same name to belong to both real and fictional persons. Rather, the differentiation would need to be made through the manner in which we apply the tags. Therefore, it is essential that tag usage is a planned part of a project. For a further definition of tags, see in this volume, Bigot Juloux, 163.

\textsuperscript{73} I address this issue in a forthcoming article on encoding Elamite inscriptions.

\textsuperscript{74} “Λυθήσομαι” is Greek for “I’ll be set free” (author’s translation).

\textsuperscript{75} $<$\texttt{w}$>$ is the markup for a word in TEI. Elements: equivalent to markups in this context, they correspond to “w” in \texttt{<w ana='singular'>apple</w>}. For further information, see in this volume, Bigot Juloux 165n68.
than EpiDoc, but, even with its support, at most we would have something like: `<w type='verb' subtype='not-deponent' function='indicative' ana='future' n='1' lemma='λύω'>λυθήσομαι</w>`. We can also add the value “singular” to any of the attributes used above, but that would be encoding a semantically different feature along with another feature, which would create problems later if someone wanted to analyze the attestations of those features separately.

4. There is no Graphical User Interface (GUI). This can be remedied on project basis, but the lack of a GUI can be a disadvantage to colleagues with less technical backgrounds.

There are several advantages to using EpiDoc:

1. EpiDoc’s text-based format is easily exchangeable with peers and is platform independent.

2. EpiDoc’s XML-based format is easily exchangeable with outside parties, such as computer scientists, who may provide help with certain issues. Since XML is a mature technology, most computer scientists and engineers have experience with it. Thus, help for technical problems is readily available.

3. EpiDoc is human readable to some extent. This is important for sharing one’s work with less technically trained colleagues, who would be able to understand what is happening in the document with a little effort.

4. EpiDoc has style sheets that facilitate the creation of the digital editions of the conserved texts.

5. With EpiDoc, it is easy to reuse any already encoded material. This is tricky to notice, however, because most of the time the encodings are full of project-based decisions. However, once a document is encoded with a

---

76 Attributes: precisions indicating an aspect of an element, they correspond to “ana” in `<w ana='singular'>apple</w>`. For additional explanation, see in this volume, Bigot Juloux, 165n70.

77 Graphical User Interface (GUI) refers to the medium in which users interact with the functions of the computer program through visual signs and symbols as opposed to written commands, as in the case of command line interface. Programs with GUI include, but are not limited to, Microsoft Word, Open Office Writer, Internet Explorer, and Mozilla Firefox.

78 Platform independent means that it does not depend on the user’s operating system, such as Windows, Linux, or macOS.

79 As far as the conservation of the document in a digital environment and the computations that can follow, electronic editions are of secondary importance, although they can support a variety of important digital humanities research projects.
scheme, it becomes a source that can be harvested for future projects that could be encoded with the same scheme. This advantage is mostly due to the maturity of the XML technology. Since there are lots of parsers available for XML, an EpiDoc-encoded document can be parsed and harvested with most of the major programming languages.

6. Although the lack of a GUI can present a disadvantage to some users, it can also be an advantage to technically trained users. By not having a GUI, EpiDoc gives the knowledgeable user more control. The absence of a GUI forces this user to become familiar with the XML structure and the limits of the scheme, which ultimately motivates better decisions regarding the project’s goals.

**Scalable Vector Graphics (svg)**

SVG is an XML-based language for describing 2D graphics. I suggest that SVG presents the most viable option for storing local glyph representations. SVGs are a joint product of the web development and design communities, making them reliable for expressing any type of 2D graphics that require cross-platform exchange. There are also different types of cross-platform open-source software for visualizing SVGs.

The XML nature of SVGs implies a relatively simple integration with EpiDoc. For the purpose of encoding visual phenomena, rather than semantic units, SVGs provide two very important elements for making any reasonable normalization attempt concerning the above-mentioned encoding schemes possible:

1. the abstract vector graphic to which OCR algorithms can be compared to the processed raster image
2. a means of visualization for the generated features’ vectors of the OCR procedures

---

80 Dahlstörm et al. 2011, <https://www.w3.org/TR/SVG11/> (accessed 1 April 2017), 20.
81 Inkscape is a popular example of this software (<https://inkscape.org/en/> [accessed April 1, 2017]). See Bah (2011).
82 Most of these are available as fonts, so projects should not have difficulties finding materials to train their OCR algorithms. I thank Daniel Stockholm for bringing this to my attention through our discussion at Caf’E.PHE (January 27, 2017).
83 This is partially true, because feature extraction algorithms differ in an OCR process. Extracted features are especially sensitive to preprocessing of the image, so SVG cannot incorporate or visualize every feature that can be extracted with every available feature extraction algorithm. Nevertheless, if one could convert the features of a simple tracing algorithm, such as the Potrace engine, to SVG, it would be major step toward the normalization of the input required for reasonable epigraphic queries throughout the corpora.
On The Problem Of The Epigraphic Interoperability

Serialization and Epigraphic Interoperability

Up to now I have tried to avoid technical terms and issues with regard to my subject, which may have resulted in oversimplified statements on technical procedures. Now I would like to address the readers who have the necessary background to follow the discussion behind these technologies. A reader who has the sufficient technical background would notice that once vector space representation is adopted, a key problem is that of the difficulty of data serialization, since serialization ensures interoperability across systems.

The choice of format, i.e., the technology, e.g., JSON, XML, etc., is actually trivial, because transformation between major serialization formats is already extensively supported by programming languages. But the interior structure of the serialized data, that is, the representation of the shape contained within the vector space inside the serialization, is what matters the most. In more abstract terms, we should be asking ourselves what kind of graph the shape generates and how we can process it with computers. Since we are dealing with 2D structures here, the simple solution would be to represent the shape as a list of points. As the technical reader might have noticed by now, however, the general mistake of the presently analyzed encoding schemes was their lack of mathematical foundation, that is, they did not aspire to represent their data in the form of mathematical constructs that would have given them long-term flexibility and technological independence. This lack of mathematical foundation is something the new generation of research should avoid. The solution has to be extensible, especially to 3D shapes. In my humble opinion, the ideal solution would be along the lines of the following:

Let $G$ be the graph generated by the shape $\theta$,
let $V = \{v_1, v_2, \ldots, v_n\}$ be the vertex set of $G$,
let $E = \{e_1, e_2, \ldots, e_j\}$ be the edge set of $G$,
let $v_k, v_i \in V$, and let $U = \{u_1, u_2, \ldots, u_x\}$ where $\forall u \in U, u(v_k) = v_i$,
let $I = \{v_k, u_j(v_k) | v_k \in V, u_j \in U\}$,
then $I \subseteq E$.

Simply put, if $G$ is the graph generated by the shape, the relationship between the nodes of the graph is modeled as a set of functions that maps one node to

For the Potrace engine, see Selinger (2003, <http://potrace.sourceforge.net> [accessed April 1, 2017]).

Graph as in graph theory: $G = [\text{Vertices}, \text{Edges}]$. 

84
another node.\textsuperscript{85} The application of these functions to a node simply outputs another node. As the reader might have noticed by now, I am simply proposing to model each edge between two nodes as a function. In this way, adding new information with regard to the relationship between any two points on the graph would simply mean adding new functions to their relative set of functions. Notice that this way of representing shapes gives great flexibility.\textsuperscript{86} We can even represent the relationships within entire corpora of inscriptions if we consider them as components of the same graph. In terms of data structures used in programming languages, we simply have an array $S$, which contains an array $V$, an array $E$, and an associative array $T$.\textsuperscript{87} The keys of $T$ are pairs that contain a node as the first term and the function value of first term as the second term. These two terms, which are essentially two nodes, are mapped to an array of functions.\textsuperscript{88} Notice that we do not propose to store the result of the functions; we propose to store the functions themselves.\textsuperscript{89}

There are already some properties that we can infer if we model a shape in this way. For example, we would know that every vertex is related to every other vertex with a distance function, and that a damage is simply an edge cut creating different components within the shape graph, of which some are visible, and some are not, visibility being a simple coefficient between 0 and 1. Increasing the dimensionality of the vector space is also possible without changing the structure of the serialization.\textsuperscript{90}

\textsuperscript{85} I use the terms “node” and “vertex” interchangeably.

\textsuperscript{86} As usual, the flexibility comes at the price of difficulty in standardization. A project-based standardization can be obtained by implementing type definitions for functions, that is, defining which function maps which type to which type. Even some global standardization is possible by defining custom algebraic data types, as in Haskell programming language.

\textsuperscript{87} Notice that this is not the most efficient way to represent graphs in a computing environment. One efficient way would be to use an adjacency matrix for representing $E$ and $V$ together. An adjacency matrix is a matrix whose rows and columns are labeled with the nodes, and whose cell value represents whether an edge is present between the node labeled in the row and the node labeled in the column.

\textsuperscript{88} The signature of the data structure $S$ in Python would be: $S = [\text{set()}, \text{set()}, \{() : []\}]$, where each element of $S$ corresponds to $V$, $E$, and $T$, respectively.

\textsuperscript{89} This should give flexibility in implementing their computations, depending on practical circumstances.

\textsuperscript{90} An increase or decrease in the dimensionality of the vector space can have a non-trivial impact on the related function set of the node group. Adapting a given function to higher dimensionality, however, provides much more backward compatibility(!) than any encoding scheme we have seen so far.
Though this is the ideal solution, it would be unrealistic to expect that it would be adopted within the community of researchers in the humanities or ancient history apart from those who are particularly interested in mathematics, graph theory, and the like. Unfortunately for a potential solution, it would be difficult to convince colleagues that it is a solution that is worth spreading. Thus it would be more practical to use a technology that is easier to work with but also is capable of being transformed into the above-mentioned structure.

Here SVG technology, in my opinion, provides the middle ground, not as a serialization technology, but as the first step to transforming the physical shape into a graph. We can add other qualities of the shape with respect to its physical state by other means. Plus, there are already a lot of fonts dealing with ancient languages, including those that are supported by presently analyzed encoding schemes. Most of these fonts, if not all of them, use SVG to represent signs. Even in the worst-case scenario, in which reusing an SVG drawing from a font for modeling a physical sign being studied by an epigrapher is not possible, SVG encoded signs present a good resource as a training dataset if the dataset is well augmented.

Ultimately, what we want is a representation that contains everything we know about the shape, and one that is open to the adding of supplementary information once discoveries are made. Computationally, this representation would provide a lot of labeled data, which in turn can be used with supervised learning algorithms, opening up all the benefits of using artificial-intelligence technologies. Unfortunately, the current scope of this paper does not permit me to detail the application of these technologies to numerous aspects of epigraphy and ancient history in general.

Some Practical Notes

Generally speaking, the tools, or rather the existing elements, that can help us to deal with the problem of normalization have a very simple architecture. EpisDoc should hold the SVG graph objects, which are based on the results of augmentation.

---

91 Augmenting a dataset in this context means modifying the dataset in order to increase the invariance of the algorithms working on a classification task. This technique is mostly used where Convolutional Neural Networks (CNNs) are used as a classifier. CNNs are neural networks that have a special layer called a convolutional layer, which consists of convolutional filters. They provide the state-of-the-art performance in image-classification tasks. See also in this volume, Svärd, Jauhiainen, Sahala, and Lindén, 229n19.

92 For a recent application of AI to Akkadian which incorporated the ATF scheme, see Homburg and Chiarcos (2016).
feature extraction algorithms. First, each should have an identifier indicating its class, which can be either a Unicode code point or an encoded sign in the encoded document. Each should also have an identifier indicating the source features’ vector. The namespace of the identifier of the features’ vector should include: a stable identifier for the document; an arbitrary unique integer for instantiating this features’ vector so that if, in the future, we have better algorithms that can extract better features, we can cast this one aside and add the updated result with another arbitrary unique integer; and, lastly, the position of the sign over an arbitrary unique decimal that represents the total number of signs in the document in order to instantiate the current sign count for marking changes in the future. If the use of EpiDoc is impossible as a scheme, or if it is not really desired, then SVGs should be mapped to their relative objects at first again in a separate file. Then, the signs in the encoded document would be mapped to the SVGs with LD technologies as mentioned above.

Currently, no project uses the implementation described above, but there is an ongoing effort in that direction. In terms of staff, any project that wants to deploy the implementation described above would need someone experienced in OCR technologies, equipped with working knowledge of LD technologies, and possessed of computational experience with the encoding schemes mentioned above. This staff is of course a supplement to the actual encoding

---

93 Feature extraction algorithm: the process through which one extracts the factors of variance from a digital representation of an object. For example, it could be an algorithm that detects the corners and sides in a picture of a triangle. Factors of variance (also called factors of variation) refers to elements that are believed to be constitutive of the observed state of that which is observed. This could include things such as elements of form (e.g., lines, angles), degrees of arc or convexity, color, and light intensity. Such features make the object distinguishable in a representation.

94 This last abstraction might seem unnecessary, but since the counted signs can be changed if a new interpretation of the document arises, and since change in the counted signs does not necessarily imply a change in the total count of the signs but may imply a change in the sign position, it would be wiser to instantiate the total sign count. Total sign count should be based on the already encoded material. It should include damaged signs but not gaps presented as sign numbers. For example, a gap may be as wide as three signs.

95 The main source for the observation of the problem of achieving epigraphic interoperability that is described above, the quadrilingual vase of Darius 1, is being processed with the tools described above. Unfortunately, time and financial constraints have not yet permitted this research to be fully realized. However, at least the encoding work, with the kind help of Marine Béranger, continues.

96 Even if all of these qualities can be found in an individual, it is certain that that individual is an exception rather than the rule because OCR and LD technologies are themselves the
project, which could recruit as many encoders as needed, as well as expert epigraphers, philologists, and historians.

Conclusion

This paper began by presenting schemes used for encoding the texts from the first millennium BCE. The list of schemes is not exhaustive, and their descriptions are only summaries. This discussion was intended to give a brief presentation, after which it could be seen that these encoding schemes have some interesting similarities. All of them use an intermediary element for mapping signs to the encoded document. The major difference is the nature of the intermediary element. It might be semantic or non-semantic. Another revealed issue is the problem resulting from the design decisions of the encoding schemes. Apart from the Mdc, most of the encoding schemes were designed to work on a human-to-machine interaction level, meaning that the majority of the users of the encoded documents were thought to be humans, so the role of the machine was to provide and conserve the document for a human reader trained in the original language of the text. This was a design flaw, with many consequences, one of them being the problem of achieving epigraphic interoperability.

The problem of achieving epigraphic interoperability as discussed here is due to the different attributes of the elementary unit of the encoding schemes. Mdc and EpiDoc consider that a sign is made up of different parts, for which it is more or less possible to describe the state of physical preservation, while the others consider a sign as a monolithic block that is described either as damaged or intact. The difference in the attributes creates an incompatibility in the normalization process required by the queries on multilingual documents. The low-level nature of the problem prevents any working solutions from within the encoding schemes.

In the end, I described part of a possible solution to this problem. I have demonstrated that if normalization is to be attained on reasonable grounds, some form of OCR would be necessary. The requirements of this process and domains of specialized professional research. At the planning stage of a project, a team should not cut the budget in this area, because even if the team can find a talented individual who can do all of those things, it would be very difficult to maintain the code she or he has generated if said individual departs the project. A viable compromise would be to recruit two or three engineers to the institutional faculty and share them among the projects.
the data generated by it invite us to use a vector graph associated with a semantic unit. In light of this tool set, I recommended including some combination of EpiDoc, SVGs, Unicode, relative encoding schemes, and LD technologies. How they are integrated can be evaluated on a case-by-case basis. Some other practical concerns about the necessary staff for the integration models proposed here were also addressed. The effort to create an exemplary implementation of the EpiDoc model based on the quadrilingual vase of Darius I is ongoing.

References

Anderson, Lloyd, Karljuergen Feuerheim, John Jenkins, Rick McGowan, and Dean Snyder. 2000. *Initiative for Cuneiform Encoding (ICE): Report of the First Conference*. Initiative for Cuneiform Encoding, updated 3 November 2000. <http://pages.jh.edu/~dighamm/ice/ice1report.html>.

Bah, Tavmjong. 2011. *Inkscape: Guide to a Vector Drawing Program*. 4th ed. Boston: Pearson Education, Inc.

Bavant, Marc. 2014. *svg Cuneiform Tool (v4.3)*. <http://kursoj.pagesperso-orange.fr/cunei/>.

Borger, Rykle. 2004. *Mesopotamisches Zeichenlexikon*. AOAT 305. Münster: Ugarit-Verlag.

Buurman, Jan, Nicolas Grimal, Michael Hainsworth, Jochem Hallof, and Dirk van der Plas. 1988. *Manuel de codage des textes hiéroglyphiques en vue de leur saisie sur ordinateur*. Paris: Institut de France.

Chapman, Nigel. 1987. *LR Parsing: Theory and Practice*. Cambridge: Cambridge University Press.

Cheriet, Mohamed, Nawwaf Kharma, Liu Cheng-Lin, and Suen Ching. 2007. *Character Recognition Systems: A Guide for Students and Practioners*. Hoboken, NJ: John Wiley & Sons, Inc.

Copi, Irving M., Carl Cohen, and Kenneth McMahon. 2016. *Introduction to Logic*. 14th ed. London-New York: Routledge.

Dahlstörm, Erik, Patrick Dengler, Anthony Grasso, Chris Lilley, Cameron McCormack, Doug Schepers, Jonathan Watt, Jon Ferraiolo, Fujisawa Jun, and Dean Jackson. 2011. *Scalable Vector Graphics (SVG) 1.1*. 2nd ed. W3C. <https://www.w3.org/TR/SVG11/>.

Daressy, Georges. 1901. *Catalogue Générale des Antiquités Egyptiennes du Musée du Caire, 25001–25338: Ostraca*. Cairo: Le Caire Impr. de l’Institut français d’archéologie orientale.

Dawson-Howe, Kenneth. 2014. *A Practical Introduction to Computer Vision with Open CV*. West Sussex, UK: John Wiley & Sons, Ltd.
Elliot, Tom, Gabriel Bodard, Simona Stoyanova, Charlotte Tupman, Scott Vanderbilt, and Elli Mylonas. 2017. *EpiDoc Guidelines: Ancient Documents in TEI XML*. Last modified December 12, 2017. <http://www.stoa.org/epidoc/gl/latest/>.

Eraslan, Doğu Kaan. 2017a. “On the Use of Existing Resources for Ensuring Epigraphic Interoperability of Ancient Texts: What Do We Do? How Do We Do It? How Can We Make It Better?” Paper presented at the Caf’E.PHE, École Pratique des Hautes Études, Paris.

Eraslan, Doğu Kaan. 2017b. *PySesh: A Python NLP Complement to Jseq*. Last modified March 22, 2017. <https://github.com/D-K-E/PySesh>.

Eraslan, Doğu Kaan. 2017c. *C-ATF Feature Extractor*. Last modified May 29, 2017. <https://github.com/D-K-E/c-atf-feature-extractor>.

Everson, Michael, Karljuergen Feuerheim, and Steve Tinney. 2004. *Final Proposal to Encode the Cuneiform Script in the SMP of the UCS*. <http://std.dkuug.dk/jtc1/sc2/wg2/docs/n2786.pdf>.

Felicetti, Achille, Francesca Murano, Paola Ronzino, and Franco Niccolucci. 2016. “CICOD CRM and Epigraphy: A Hermeneutic Challenge.” In *Extending, Mapping and Focusing the CRM. Proceedings of the Workshop EMF-CRM 2015, Poznań, Poland, September 17, 2015*, edited by Paola Ronzion, 55–68. Prato, Italy: FIN, Servizi Didattici e Scientifici per l’Università di Firenze. <http://ceur-ws.org/Vol-1656/paper5.pdf>.

Gippert, Jost. 1999. “Language-Specific Encoding in Multilingual Corpora: Requirements and Solutions.” In *Multilinguale Corpora: Codierung, Strukturierung, Analyse. 11. Jahrestagung der Gesellschaft für Linguistische Datenverarbeitung*, edited by Jost Gippert, 371–384. Prague: Enigma Corporation, Inc.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. Cambridge, MA: MIT Press.

Hallock, Richard. 1969. *Persepolis Fortification Tablets*. OIP 92. Chicago: The Oriental Institute of the University of Chicago.

Holley, Rose. 2009. “How Good Can It Get? Analysing and Improving OCR Accuracy in Large Scale Historic Newspaper Digitisation Programs.” *D-Lib Magazine* 15 (3/4). <http://www.dlib.org/dlib/march09/holley/03holley.html>.

Homburg, Timo, and Christian Chiarcos. 2016. “Akkadian Word Segmentation.” In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris: European Language Resources Association.

Jansen-Winkeln, Karl. 2014. *Inschriften Der Spätzeit IV: Die 26. Dynastie. Vol. 1. Psametik I.–Psametik III*. Wiesbaden: Harrassowitz.

Johnson, Kyle P., Patrick J. Burns, Tyler Kirby, Luke Hollis, Sourav Singh, Chaitanya Sai Alaparthi, Bhupendra Singh Chauhan, et al. 2016. *CLTK: v0.1.41*. Last modified August 11, 2016. <https://doi.org/10.5281/zenodo.60021>.

Kaufman, Stephen A. 1987. “Coding Principles.” In *The Comprehensive Aramaic Lexicon: Text Entry and Format Manual*, 5–19. Baltimore: The Comprehensive Aramaic Lexicon. <http://calh.cn.huc.edu/pdfs/CalManualIntro.pdf>.
King, Leonard William, and Reginald Campbell Thompson. 1907. *The Sculptures and Inscriptions of Darius the Great on the Rock of Behistûn in Persia: A New Collation of the Persian, Susian and Babylonian Texts*. London: The British Museum.

Lang, Serge. 1986. *Introduction to Linear Algebra*. Undergraduate Texts in Mathematics. 2nd ed. New York: Springer.

Le Boeuf, Patrick, Martin Doerr, Christian-Emil Ore, and Stephen Stead, eds. 2015. *Definition of the CIDOC Conceptual Reference Model*, 6.2. <http://cidoc-crm.org/sites/default/files/cidoc_crm_version_6.2.pdf>.

Nederhof, Mark-Jan. 2013. “The Manuel de Codage Encoding of Hieroglyphs Impedes Development of Corpora.” In *Texts, Languages, Information Technology in Egyptology: Selected Papers from the Meeting of the Computer Working Group of the International Association of Egyptologists (Informatique & Égyptologie)*, Liège, 6–8 July 2010, edited by Jean Winand and Stéphane Polis, 103–110. *Ægyptiaca Leodiensia* 9. Liège: Presses universitaires de Liège.

Nurmikko-Fuller, Terhi. 2017. “From the Philology of Ancient Sumer to the Semantic Web.” Paper presented at the Caf’E.PHE, École Pratique des Hautes Études, Paris.

Peters, James F. 2017. *Foundations of Computer Vision: Computational Geometry, Visual Image Structures and Object Shape Detection*. Intelligent Systems Reference Library 124. Cham, Switzerland: Springer.

Schmidt, Erich. 1970. *Persepolis III: Royal Tombs and Other Monuments*. OIP 70. Chicago: The Oriental Institute of the University of Chicago.

Selinger, Peter. 2003. “Potrace.” <http://potrace.sourceforge.net>.

Shahbazi, Alireza Shapur. 2012. “Darius III. Darius I the Great.” *Encyclopaedia Iranica*. Last modified May 9, 2012. <http://www.iranicaonline.org/articles/darius-iii>.

Stolper, Matthew, and Joan Goodnick Westenholz. 2002. “A Stone Jar with Inscriptions of Darius I in Four Languages.” *Arta* 5: 1–13.

TEI Consortium. 2018. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*, revision f4d8439. The TEI Consortium. <http://www.tei-c.org/release/doc/tei-p5-doc/en/Guidelines.pdf>.

Thonemann, Peter, Charles Crowther, Edouard Chiricat, Maggy Sasanow, eds. 2012. “MAMA 314 (Perta): Votive Bomos Dedicated by Papias to Apollo.” *Monuments Asiae Minoris Antiqua XI*. University of Oxford and the Center for the Study of Ancient Documents. <http://mama.csad.ox.ac.uk/monuments/MAMA-XI-314.html>.

Tinney, Steve. 2017. “CDLI ATF Primer.” *Oracc: The Open Richly Annotated Cuneiform Corpus*. Oracc. <http://oracc.museum.upenn.edu/doc/help/editinginatf/cdliatf/>.

Weisstein, Eric W. n.d. “Vector Space.” *MathWorld*—A Wolfram Web Resource. <http://mathworld.wolfram.com/VectorSpace.html>.

Whistler, Ken, Mark Davis, and Asmus Freytag. 2008. *Unicode Character Encoding Model*. Unicode Technical Report 17. <http://www.unicode.org/reports/tr17/>.
Wood, David, Marsha Zaidman, Luke Ruth, and Michael Hausenblas. 2014. *Linked Data: Structured Data on the Web*. Shelter Island, NY: Manning Publications.

Yardeni, Ada. 1994. “Maritime Trade and Royal Accountancy in an Erased Customs Account from 475 BCE on the Aḥiqar Scroll from Elephantine.” *BASOR* 293: 67–78.