

Language-specific encoding in multilingual corpora: Requirements and solutions

Jost GIPPERT

Only a century after Johannes Gutenberg had invented the art of employing movable types in printing, this new method of publishing had developed in such a way that it became possible to use it for all kinds of multilingual documents, the most striking examples being the so-called “polyglot” Bible editions where modern versions of the Holy Scriptures were arranged synoptically with their Latin, Greek, Hebrew, or Syriac ancestors, all printed in their original scripts (cf., e.g., HUTTER 1599-1643). And by 1650, it had become quite common to adopt this standard of printing to what we might call routine products such as a young theologian’s forty-page doctoral thesis; cp. the specimen taken from HARTUNG (1659) which shows the beautifully mixed arrangement of Latin, Greek, Hebrew, Syriac, and Arabic possible at that time.

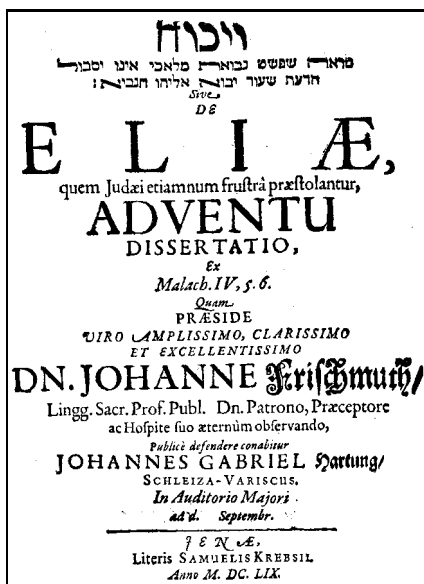


Figure 1: HARTUNG (1659), title page

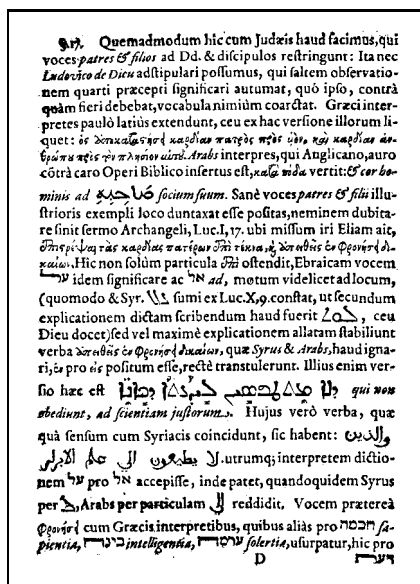
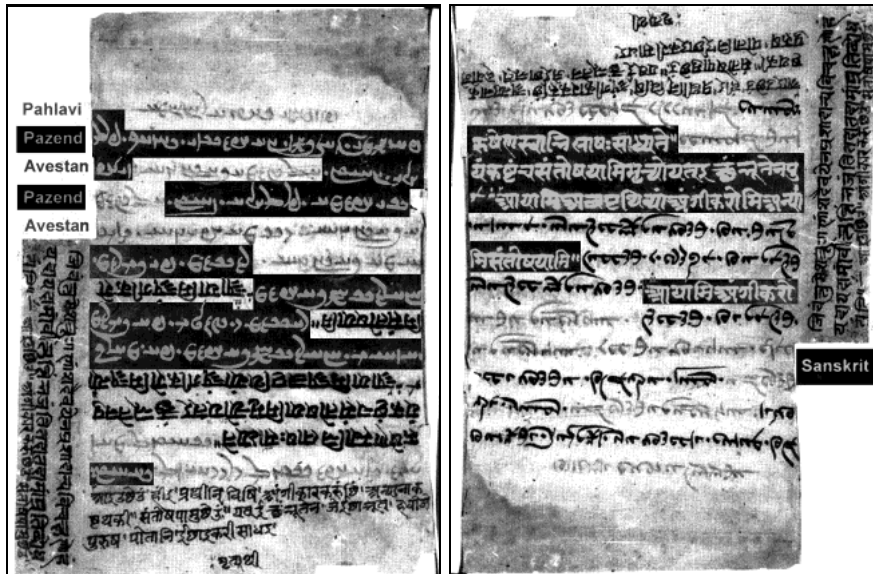


Figure 2: HARTUNG (1659), p. [23]

Today, more than 500 years after Gutenberg's invention, we are still striving to attain a comparable standard when working with digital equipments, the most challenging problem consisting in the mixed processing of different script directions. This is by no means a far-fetched task to be mastered by a fringe group of ivory-towered outsiders only, as some people might think, because the production of multilingual texts plays a steadily increasing rôle in the transfer of data via the World Wide Web, and different script directions are involved wherever European and Near Eastern languages have to be arranged side by side. And the problem is in fact not even a modern one: It had to be coped with, e.g., by the writers of Zoroastrian manuscripts such as the one reproduced in figures 3 and 4¹. Apart from its basic part, the Avestan (Old Iranian) text called *Aogamadaēcā*, this manuscript contains the Middle Persian version of the same text written in two different scripts, its translation into Sanskrit and a translation into Old Gujarātī, added secondarily *in margine*. Of the scripts involved, only the



Figures 3-4: Avestan, Pahlavī, Pāzēnd and Sanskrit versions of *Aogamadaēcā*

¹ Ms. K 42 of the Copenhagen Royal Library. Its first page reproduced here is taken from the facsimile edition printed in JAMASPASA (1982).

one used for Sanskrit and Gujarātī (the so-called “Devanāgarī”) is directed left-to-right while both the Avestan script (used for Avestan and Middle Persian in its so-called “Pāzend” style) and the Middle Persian script proper (so-called “Pahlavī”) are written from right to left. As the figures show, the solution applied by the writer is a simple one: Arriving at a position where he had to switch from Avestan to Sanskrit, he turned the paper by 180°, then continuing with Devanāgarī in its usual direction, but with lines arranged upwards; at the end of the Sanskrit passage, he turned the paper again in order to proceed with one of the other scripts, etc. etc.

It is not the question of processing and printing mixed scripts, however, that I intend to discuss in the following pages. Several software solutions have been developed for this problem in recent years (cp., e.g., figure 5 showing the “WordCruncher” screen output of Mt. 6,9, the beginning of *Our Father*, in a mixed arrangement of Georgian, Greek, Armenian, and Syriac Bible versions, both transliterated and in original scripts), and it is to be hoped (though by no means certain, cf. below) that after the adaptation of operating systems to real 16-bit encoding on the basis of Unicode, it will become more and more neglectable.

In the present paper, I shall deal with a different problem instead which is related to but not necessarily identical with the one discussed above, viz. the problem of preparing multilingual corpora for a language-specific retrieval. The synoptical arrangement of several Bible versions in one printed edition or in one electronic text file is indeed a multilingual corpus in its own right, just as the Zoroastrian manuscript we have seen

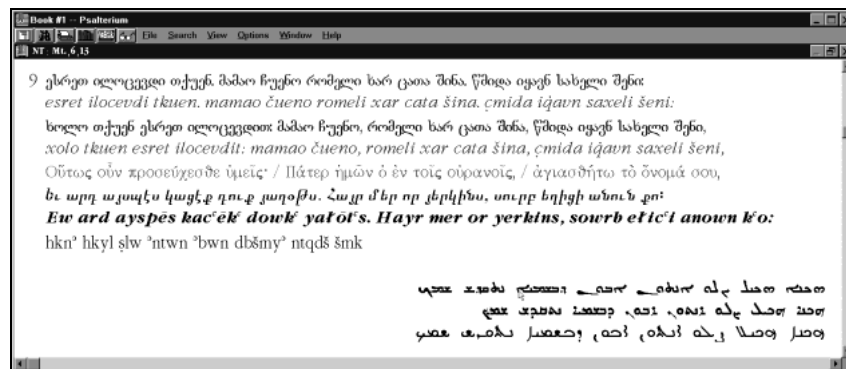


Figure 5: Mt. 6,9 in several versions (“WordCruncher” screen output)

before: The data they contain do not represent one text in the linguistic sense of this word, but several texts that are related to each other by their contents and their structure but differentiated from each other by their language. A language-specific retrieval to be undertaken under these conditions presupposes the languages involved to be clearly identifiable, i.e., separable from one another so that analyses concerning the individual languages can be performed as well as analyses concerning the mutual relationship of, e.g., linguistic units of a translated text and its model.

A software solution that matches these requirements to a certain extent by combining a reasonable amount of functions designed for linguistic analysis with a programmable language-specific interface, is the system provided by "WordCruncher". As I have discussed the facilities of this program recently in another context², I shall confine myself to a brief resumé here before trying to evaluate its efficiency.

In its present release (5.3) which is unfortunately available as an MS-Windows version only, the "WordCruncher" retrieval system³ allows to index and analyse multilingual text files of up to 2 GB size⁴. The number of languages represented in a given document should not exceed ten if they are to be analysed separately and if the data are to be accessed from a remote server; otherwise up to 31 languages can be kept distinguished within one text file. The retrieval presupposes a preindexation which consists in the adaptation of several file formats: The initial input requires a plain (8-bit ASCII/ANSI or DOS) text file containing the textual data with additional tags storing information about the inherent structure of the text (e.g., page and line numbers, chapter or paragraph headings etc.) and its items. This input file (a so-called ETA file) must first be converted into a proprietary format (ETB file) which is ready for the preindexing process. The resulting index file (ETX) serves as the basis for the retrieval which is extremely fast as it involves no further (sequential) searching.

² GIPPERT (1997), 75-93.

³ Developed by Brigham Young University (since 1985). The text viewer necessary for the retrieval of preindexed data (both locally and from a remote server) can be downloaded free of charge from the TITUS web pages (cf. <http://titus.uni-frankfurt.de/texte/tituswc2.htm>). For the full version cf. <http://www.wordcruncher.com> or contact johnston@wordcruncher.com.

⁴ The maximum file size depends on the operating system, not on the program.


```

...
TEXT_STYLE mge16
  FONT georgian-mcxeta
  INDEX_FLAG on
  FONT_POINTSIZE 16
  TEXT_COLOR [0 128 0]
                [255 255 255]
  END
TEXT_STYLE mge22
  FONT georgian-mcxeta
  INDEX_FLAG on
...
    
```

Table 2: TEXT_STYLE definition

```

...
FONT georgian-mcxeta
  FONT_NAME Titus Mxedruli
  FONT_FAMILY roman
  CHAR_SET ansi
  PITCH proportional
  DIRECTION left-to-right
  FONT_TYPE TrueType
  LANGUAGE Oldgeorgian
  END
FONT greek
...
    
```

Table 3: FONT declaration

```

...
LANGUAGE Oldgeorgian
  LANGUAGE_ID GEORGIAN
  CHARACTER_MAP Georgian
  TEXT_STYLE mge16
  LST_FILENAME GEORGICA.ETX
  END
LANGUAGE Greek
  LANGUAGE_ID GREEK
  CHARACTER_MAP Greek
  TEXT_STYLE grx16
...
    
```

Table 4: LANGUAGE definition

```

...
CHARACTER_MAP Georgian
  MAPS 201=144 230=145
198=146 244=147 246=148
242=149 251=150 249=151
255=152 214=153 220=154
162=155 163=156 165=157
199=128 252=129 233=130
226=131 228=132 224=133
229=134 197=143 131=159
225=160 237=161 243=162
250=163 241=164 ...
    
```

Table 5: Keyboard definition

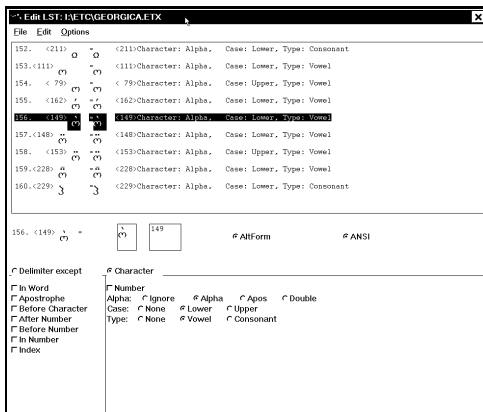


Figure 6: Definition of sorting order

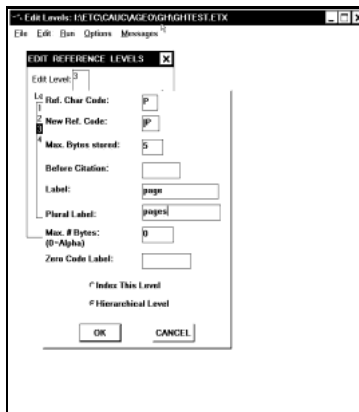


Figure 7: Definition of text levels

Once this information has been established, it can be used by the pre-indexing program for processing. The resulting information, consisting of language-specific indexes of word forms, is then available for immediate retrieval via a so-called “word wheel”, i.e. a look-up menu which displays the contents of the indexes in the respective script and alphabetic order of the language in question as defined in the LST file (cf. figures 8 to 12 showing the word-wheels for Georgian, Armenian, Syriac, and Greek, with a focus on the words meaning “Father”). This search engine meets further requirements of linguistic retrieval by enabling its user to extend the search to other preindexed texts collected in a “library”, to group (inflected) word forms pertaining to one lemma together, to investigate into grammatical features (provided these were prepared for indexing by entering appropriate tags) or to search for word forms that appear side by side within a certain distance and/or within certain levels of the text.

As to multilingual texts (or corpora, in the sense outlined above), the facilities provided by the WordCruncher system are not very elaborate yet. Even in the present release, it is possible, however, to formulate a cross-linguistic search such as the one indicated in figure 13. In this example, the search object is a context in which the Greek vocative *πάτερ*, the Armenian *hayr*, and the Georgian *mamao*, all meaning “father”, occur within the same verse. The result we expect to find is Mt. 6,9, but other contexts such as Mt. 26,42 will be found as well. Of course, this is not sufficient for an automatical generation of complete lists of the correspondences that exist between individual items of “parallel” texts, but it may give a first idea of where further investigation might be necessary.

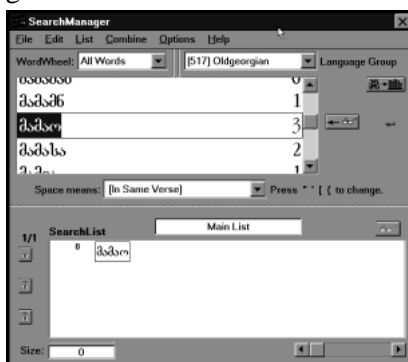


Figure 8: Georgian “word wheel”

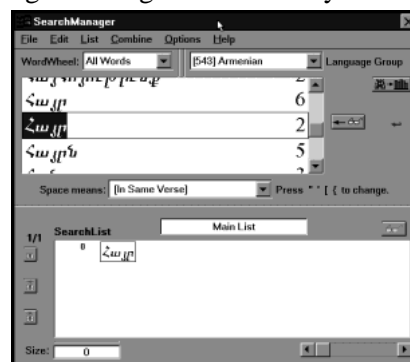
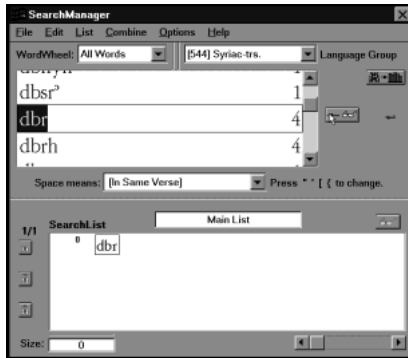


Figure 9: Armenian “word wheel”



Figures 10-11: Syriac “word wheels”:



Transcription vs. original script

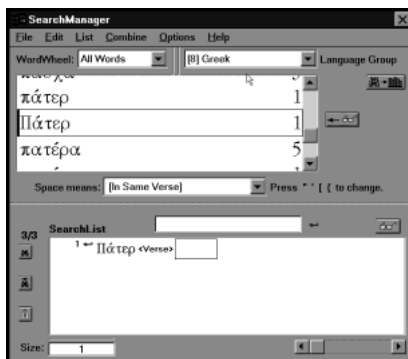


Figure 12: Greek “word wheel”

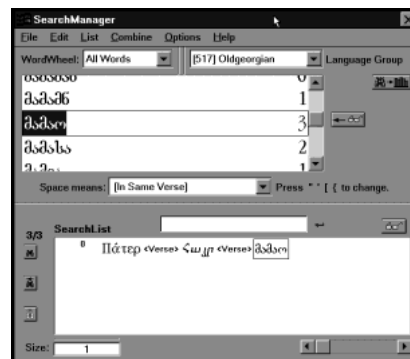


Figure 13: Mixed language retrieval

Returning to the question of encoding, it has to be stated that the system used by WordCruncher is rather disadvantageous with respect to multilingual corpora. It is at least two features that must be mentioned here. First, the system is clearly surface-oriented, with language tagging being unseparable from (and even dependent on) script tagging. Although this feature did not prove to be impedient when the WordCruncher system was adopted to the requirements of the TITUS project, a data base consisting of more than 1 GB of mono- and multilingual corpora in Indo-European and adjacent languages⁵, it will pose serious problems when the

⁵ Cf. GIPPERT (1996) for a detailed description and <http://titus.uni-frankfurt.de/texte/texte2.htm> for up-to-date information.

data are to be converted into another format such as an SGML or XML tagged structure. The same holds true for the second disadvantage which consists in the fact that the present 8-bit basis does not allow for a unique and consistent encoding of several scripts, let alone languages. A multilingual corpus such as the one illustrated above is, in its WordCruncher representation, characterised throughout by so-called “font mapping” which means that a given byte value (e.g. #97) is used for different purposes (e.g., Latin *a*, Greek α , Armenian ω , or Georgian ა), depending on the actual font (cp. table 1). As a result of this, it is hardly possible to recover the intended structure whenever portions of the text are to be exported. In my personal view, the principle of font mapping, typical for WYSIWYG applications, has proved to be the most disastrous circumstance of 8-bit encoding with respect to data exchange, and time is ripe indeed to get rid of it.

With the 16-bit standard of UNICODE approaching its completion, a radical solution for this problem seems to be close, all the more since the unique encoding of 65,536 characters it implies can already be used to a certain extent in an 8-bit based environment⁶. It remains doubtful, however, whether the requirements I have exemplified above will indeed be met by Unicode implementations in the near future.

Let us look at the *Aogəmadaēcā* manuscript again to see what problems we have to face when trying to use Unicode for the encoding of multilingual text documents written in several ancient scripts. The first problem consists in the fact that two of the scripts involved, viz. Avestan and Middle Persian-Pahlavī, are not and will not be part of the 16-bit Unicode standard proper (while Devanāgarī is). They are planned for storage in the so-called “surrogate area” instead which requires a 4-byte encoding and which is not accessible at present in any environment.

⁶ The two leading WWW browsers, Netscape Navigator / Communicator and Microsoft Internet Explorer, have implemented Unicode support in their latest versions (4.0 and higher). What they process is not plain Unicode encoding, however, but a 7-bit or 8-bit transformation of it (UTF-7 and UTF-8). The capabilities of handling the data further depend on the operating system: By now, only MS-Windows 95, 98 and NT (4.0 and higher) are fully Unicode-compatible. For details cf. <http://titus-uni-frankfurt.de/unicode/unitest2.htm>, which contains links to several sample pages; many texts of the TITUS data base are also available in UTF-8 format, cf. <http://titus-uni-frankfurt.de/texte/texte2.htm>.

Furthermore, there is no way yet to cope with the problem of script direction involved here: Although Arabic or Hebrew scripts have been included in the standard, the automatic handling of script directions which is declared to be the task of “rendering engines” that interpret and process the encoding⁷, has not yet been implemented. The same holds true for the switching between different variants of characters (isolated, initial, central, and final, depending on the context) which is a typical feature of the Arabic script but also of Pahlavī: Unicode, by its structure, provides an encoding for basic “character” values only, not for variants that are regarded as “glyphs”, leaving the task of selection to “intelligent fonts” or “rendering engines” again⁸.

A similar problem is met with when Devanāgarī is to be processed. Here, too, the encoding provided by Unicode is confined to basic characters, which in the case of Indic scripts means syllabic units like क, प, or म, i.e. *ka*, *pa*, and *b^ha* (cf. figure 14 showing the Unicode block in ques-

Table showing UNICODE characters in block U+0900 through U+09FF:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
UTF8: 224, 164, 128; UNICODE: 090	॥	॥	:	॥	अ	आ	इ	ई	उ	ऊ	ऋ	ॠ	ए	ऐ	॥	॥
UTF8: 224, 164, 144; UNICODE: 091	॥	॥	॥	॥	क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	॥	॥
UTF8: 224, 164, 160; UNICODE: 092	॥	॥	॥	॥	ट	ड	ढ	ण	त	थ	द	ध	न	प	फ	ब
UTF8: 224, 164, 176; UNICODE: 093	॥	॥	॥	॥	र	ॠ	ल	ळ	॥	व	श	ष	स	ह	॥	॥
UTF8: 224, 165, 128; UNICODE: 094	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥
UTF8: 224, 165, 144; UNICODE: 095	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥
UTF8: 224, 165, 160; UNICODE: 096	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥
UTF8: 224, 165, 176; UNICODE: 097	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥	॥

Figure 14: Devanāgarī encoding in Unicode

⁷ According to the Unicode standard, the encoding must be in left-to-right order in all cases. This means that right-to-left direction is a matter of the output only.

⁸ It is true that for Arabic, a large number of “glyphs” have been included as encodable units among the so-called “presentation forms” (Unicode blocks FB00 to FEFF). Their usage is not recommended, though, because it violates the basic encoding principle.

“precomposed characters”, in Unicode. In these cases, we have to employ the method of encoding basic characters and diacritics separately (i.e., as sequences such as $s + \checkmark + \acute{}$ for \acute{s}), which may pose serious problems for “rendering engines” or “intelligent fonts” again when the relative positioning of the items is concerned (cf. figures 15 and 16 showing a text passage from the Avestan *Hōm-Yašt*, *Yasna* 9,3, containing transcriptional Avestan, Middle Persian, Pazend and Sanskrit, in both 8-bit and 16-bit encoding).

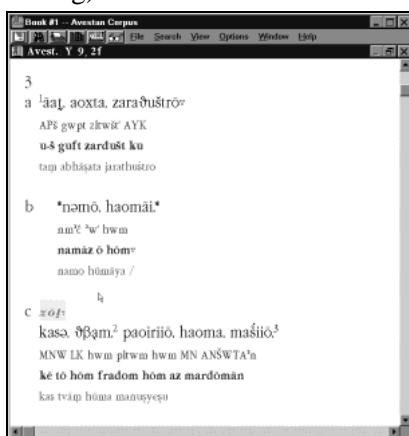
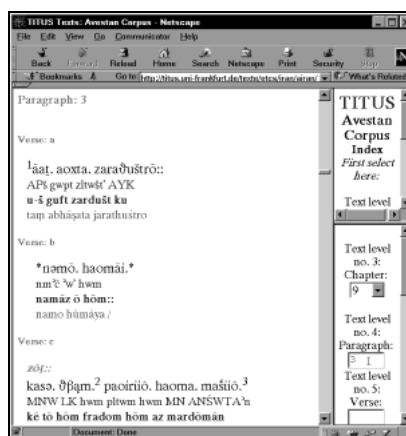


Figure 15-16: Y. 9,3 in 8-bit (WC)



and 16-bit encoding (UTF-8 HTML)

The treatment of diacritic combinations is indeed a weak point in the concept of Unicode. By offering more than one possibility of encoding for characters like \acute{a} or \acute{s} which can be treated both as “precomposed characters” and as combinations consisting of $a + \acute{}$ or $s + \acute{}$, Unicode paves the way for arbitrary decisions that are diametrically opposed to the principle of “unique encoding”. Furtheron, Unicode does not, at least in its present state, distinguish diacritics in any way according to their function, the outer shape being taken as their basic property only. Thus it is not possible to differentiate the subscript dot that appears in \acute{s} as rendering the Sanskrit retroflex [ʃ] sound (cf. Devanāgarī ष = $\acute{s}a$), from the dot below used in manuscript editing for denoting uncertain readings (cf. *sunufatarungoṣ* in the TITUS edition of the Old High German *Hildebrandslied*, figure 17). Moreover, the diacritics that have been included as such in Unicode (cf. figure 18 showing the block in question) were obviously intended not to be script-dependent; this means that one and the same

“diacresis” (¨) would have to be used for Latin-based *ä* or *ÿ*, Greek *ë*, Cyrillic *ë*, and maybe also for *ÿ* which would represent Pahlavī *y*. It goes without saying that this must lead to typographically unacceptable results, at least as long as special rendering engines are not available. Lastly, the stock of diacritics that have been incorporated is by no means complete for Latin-based transcription systems,

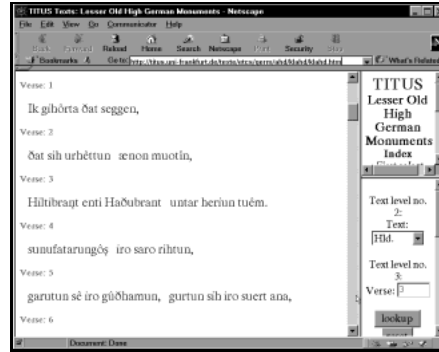


Figure 17: Passage from Hildebrandslied

let alone other uses. This may well be due to the fact that, at least for parts, Unicode encoding schemes were created not on the basis of thorough investigation into the necessities of the languages and scripts concerned but by mixing existing standards with accidental collections. To overcome this state, the TITUS project has started to build up a data base where diacritics and diacritic combinations appearing in scientific publications are stored so that they can be used for a documentation which will hopefully lead to an extension of the Unicode standard. At present, this data base is being prepared for interactive usage via the WWW (the URL will be <http://titus.uni-frankfurt.de/unicode/unicsel/unicsel.htm>; cp. figure 19 showing the provisional template).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
UTF8: 204, 128; UNICODE: 030	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ï	Ì	Í	Î
UTF8: 204, 144; UNICODE: 031	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï
UTF8: 204, 160; UNICODE: 032	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï
UTF8: 204, 176; UNICODE: 033	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï
UTF8: 205, 128; UNICODE: 034	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï
UTF8: 205, 144; UNICODE: 035	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï	Ï

Figure 18: Unicode block containing diacritics

Another problem which the use of Latin based transcriptions instead of original scripts brings about, consists in the fact that no inherent indication of the languages involved is available in this case if Unicode is applied. It is of course true that language encoding has never been an objective of Unicode, the target of which are scripts and characters, and it is merely accidental that some scripts are clearly, or even uniquely, related with certain languages. For a true language-specific encoding, it will therefore remain necessary to provide explicit tagging even in a Unicode environment.

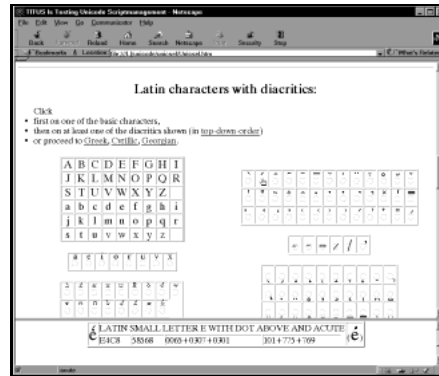


Figure 19: TITUS interactive data base of diacritic combinations

References

- GIPPERT, Jost (1996): TITUS – Alte und neue Perspektiven eines indogermanistischen Thesaurus. In: *Studia Iranica, Mesopotamica & Anatolica* 2, 1996 [1997], 49-76.
- (1997): Multilingual text retrieval: Requirements and solutions. In: *Studia Iranica, Mesopotamica & Anatolica* 3, 1997 [1998], 75-93. Preliminary version in: COOPER, Robert and GAMKRELIDZE, Thomas [Tamaz] (eds.), *Proceedings of the Second Tbilisi Symposium on Language, Logic and Computation*, Sept. 16-21, 1997, Tbilisi 1998, 106-118.
- HARTUNG, Johannes Gabriel (1659): *ויכוח מראה שפשוט נבואת מלאכי אינו יסבול הדעת: שער יבוא אליהו הנביא: sive de Eliæ, quem Judæi etiamnum frustrà præstolantur, adventu dissertatio, ex Malach. IV, 5.6. Jenæ: Literis Samuelis Krebsii.*
- HUTTER, Elias (1599-1643): *Novum testamentum Domini nostri Jesu Christi, Syriace, Ebraice, Graece, Latine, Germanice, Bohemice, Italice, Hispanice, Gallice, Anglice, Danice, Polonice.* Nürnberg.
- JAMASPASA, Kaikhusroo M. (1982): *Αογῶμαδαῖcā. A Zoroastrian Liturgy.* Wien: Verlag der Österreichischen Akademie der Wissenschaften (Sitzungsbericht, phil.-hist.Kl., 397 / Veröffentlichungen der Iranischen Kommission, 11).