Encoding and special characters

Jonathan Hooper, Leeds. May 2022

Contents

In brief

- **The process uses UTF-8-encoding throughout** - for all inputs, processing and outputs
- For many search and comparison functions, where possible, special characters are converted to standard (ASCII-range) alphabetic or punctuation characters
- Characters in non-Roman scripts (e.g. Chinese) are preserved unchanged
- Some refinements might be possible and desirable if processing large quantities of non-Roman script

Source data

- All the source data (from Leganto, Alma and from Scopus, WoS and VIAF) may contain extended characters: punctuation elements, accented a-z characters, and characters from non-Roman alphabets (Russian, Thai, Chinese etc)
- UTF-8 encoding is used in all communication with external data sources
- In the scripts, citation data from Leganto and Alma has accented Roman characters converted to unaccented ones, and special punctuation characters (e.g. em-dash, curly double quotes) converted to standard ones (dash, double quote) before being saved. This standardisation is done by the function utils.php::standardise(). Standardisation leaves special characters with no ASCII-range equivalent unchanged (e.g. Chinese characters)

Internal processing

- Characters are UTF-8-encoded
- The PHP scripts process strings as bytes rather than (possibly multi-byte) characters. e.g. a regex looking for an en-dash (Unicode U+2013) looks for the sequence of bytes "\xe2\x80\x93"

- One known non-UTF-8-safe function in the code is **utils.php::cropto()** which is part of the **similarity** algorithm - this might break a multi-byte character in the middle.
  This does not matter in practice, because the cropped string is never fed back into stored data or shown to users

Output

- The intermediate JSON files are created using PHP's json_encode() function and this stores Unicode characters as "\uCODE". e.g. the Chinese charcter 긔 (Unicode U+ACE0, UTF-8 eab3a0) is stored as "\uace0" in the JSON file (although the PHP code will process it as "\xea\xb3\xa0").
- The CSV reports are UTF-8-encoded, and include a byte-order-mark (BOM) at the start of the file - modern versions of MS Office use the presence of the BOM to identify the encoding, and should correctly render extended characters. *The BOM (U+feff i.e. "\xef\xbb\xbf") is specified in the config.ini file in the form "\ufeff"*

Similarity scores

Author and title similarity scores give a measure of similarity between the data in Leganto/Alma, and the data in the external source

- Accented characters are converted to ASCII-range equivalents, where possible, before similarity is calculated. e.g. e-acute, e-grave and e are all treated as identical. This is done in the function **utils.php::standardise()** by passing the text through an HTML encode-decode loop, removing the accent from the encoded HTML:

```
$regex = '/&([a-
z]{1,2})(acute|cedil|circ|grave|lig|orn|ring|slash|th|tilde|uml|caron);/i';
$string = html_entity_decode(preg_replace($regex, '$1', htmlentities($string)));
```

- Most punctuation (including many extended punctuation characters like dashes and quotes) is removed before comparison - this is also done by the **standardise()** function
- Other extended characters are **not** removed before comparison. i.e. two Chinese-script titles will be compared correctly by the script.
  Although since comparison is byte-wise and extended characters are multi-byte, the resulting score may be slightly higher or lower than a genuinely character-wise similarity calculation.

Special characters in reading lists

- Accented characters and special punctuation are relatively common in Leeds reading lists
  Although these characters are converted to ASCII-range ones before the data is saved and processed
- Long sequences of non-Roman text (e.g. Chinese, Japanese, Thai, Russian) are rare in Leeds reading lists, but do occur in a few cases

Special characters in Alma

- For languages like Chinese, Japanese and Korean, in general Alma keeps both Romanised and original renditions of authors and titles etc - e.g. the Romanised title is in Marc **245** and the original Chinese title in Marc **880 $6245-1**

- The script **enhanceCitationsFromAlma.php** extracts the linked 880 fields as well as the Romanised ones - the integration with VIAF uses the (Romanised) 245 as the primary title in the search, but will also compare the Chinese renditions with titles held by VIAF - on some occasions, a better match may be obtained on the Chinese renditions than the Romanisations

Special characters in the reports

- The reports should correctly display any special characters in the various fields where they might occur
  Although for titles and authors taken from Leganto/Alma, accented and special punctuation will already have been converted to ASCII-range equivalents
- For Alma records with linked non-Roman 880 fields (e.g. Chinese-language records):
  - The long report (more than one row-per-citation) includes all renditions of the author and title present in Alma: both Romanised and Original
  - The short report (one row-per-citation) shows only the *Romanised* author and title in CIT-AUTHOR and CIT-TITLE
  - The short report includes for VIAF the "best matching title" which may be the Romanised rendition, but may be the original (e.g. Chinese) one
- Where Leganto contains non-Roman text in author/title which does not have an ASCII-range equivalent, this will display in the long and short reports
- Where the integrations return non-Roman text, this will display in the long and short reports