Seeing the Heiltsuk orthography from font encoding through to Unicode: A case study using convertextract

Aidan Pine, Mark Turin

Seeing the Heiltsuk Orthography from Font Encoding through to Unicode: A Case Study Using Convertextract

Aidan Pine, Mark Turin
National Research Council Canada, University of British Columbia
aidan.pine@nrc-cnrc.gc.ca, mark.turin@ubc.ca

Abstract

Across the world's languages and cultures, most writing systems predate the use of computers. In the early years of ICT, standards and protocols for encoding and rendering the majority of the world's writing systems were not in place. The opportunity to deploy less-commonly used orthographies in cross-platform digital contexts has steadily increased since Unicode became the most widely used encoding on the web in late 2007 (Davis, 2008). But what happens to resources that were developed before Unicode standards became widespread? While many tools have been created to address this problem and other issues related to transliteration and character level substitutions,[1] this paper describes the process undertaken for the Indigenous and endangered Heiltsuk (Wakashan) language, and outlines a tool (*Convertextract*) that was designed to convert not only plain text, but also Microsoft Office (pptx, xlsx, docx) documents with the goals of updating and upgrading pre-existing digital textual resources to Unicode standards, and thus preserving the knowledge they contain for both the present and the future.

Keywords: language revitalization, Unicode, font encoding

1. Introduction

With a focus on sustaining knowledge diversity in the digital age, we introduce a tool that was developed to help ensure the preservation, proper rendering and future use of the Heiltsuk writing system as new standards for compatibility emerge. The tool was developed for rapid implementation to facilitate ease of use with other font-encoded writing systems. For textual knowledge to be encoded and viewed in a digital medium, the writing system in which it is composed must be digitally legible on a range of operating systems. For years, the orthographies of many languages were not easily supported by computers, and many continue to be constrained by requiring the installation of customized fonts and other proprietary tools. Despite these challenges, members of the Heiltsuk community of Bella Bella, Western Canada, have adapted and used a variety of techniques for communicating in written form using the established orthography for their language, from font-encoded writing systems to sending images of written text and the use of ad-hoc transliterations. A new generation of field linguists are being trained to be cognizant of the challenges of dealing with legacy data and language-specific font encodings (cf. Bowern, 2015).

Now that most of the world's written languages are supported by Unicode standards, tools that convert text from earlier non-Unicode systems to Unicode standards are increasingly important for preserving and sustaining the knowledge that is recorded in earlier digital file formats. Most of the existing digital language resources in the Heiltsuk community are stored in Microsoft Office file formats using specific styling and formatting. *Convertextract* was designed to minimize the possibility for human error associated with re-typing such documents and to reduce the time burden of using a plain text transliterator and reformatting documents one at a time. With specific reference to its application in the Heiltsuk community, we demonstrate three implementations of *Convertextract* that while computationally and metaphorically simple, hold exciting potential for community impact and widespread uptake.

2. Background

2.1 Bits & Bytes

A core factor that led to the flourishing of the digital age was the development of a standardized way for the 0s and 1s (binary code) that interacted with computer hardware to be encoded into something legible by humans. In the early 1960's, the American Standard Code for Information Interchange (ASCII) was developed to achieve just that (Gorn, Bemer, & Green, 1963). ASCII prescribed that 01100001 would represent the character "a", 01100010 would represent the character "b", and so on.

An eventual if rather self-evident problem with ASCII, at least to those familiar with the level of global linguistic diversity and variation, is that since a Bit is a binary value, and because ASCII is limited to 7 Bits, only 128 (2^7) possible characters exist in an ASCII-type encoding system. Considering all of the characters that exist in the world's writing systems, the limitation of 128 characters ensured that support for non-English characters was not readily available within the ASCII system.

UTF-8, developed in the 1990's, would later become the most widely used encoding on the web. UTF-8 encodes up to 4 Bytes instead of 1, and following a restriction on the total possible combinations set in 2003, allows for 2^{21} (2,097,152) possible characters (Yergeau, 2003). The first 128 characters in UTF-8 are identical to ASCII, meaning that 01100001 still indicates "a" in UTF-8, just like it did in ASCII. But UTF-8 can also prescribe sequences such as: 11000100 10011111 which will be rendered as "ğ".

[1] Including, but not limited to : URoman https://www.isi.edu/~ulf/uroman.html, Epitran (Mortensen et. al, 2016), Chatino transliteration http://ruphus.com/chatino_transliteration/, Inuktitut Transcoder http://www.inuktitutcomputing.ca/Transcoder/, Either/Orth http://orth.nfshost.com/, Digital Linguistics Transliterator https://tools.digitallinguistics.io/transliterator/.

2.2 Hiłzaqv Language and Culture Mobilization Partnership

Following the signing of a Memorandum of Understanding in 2016 between the Heiltsuk Cultural Education Centre, the Bella Bella Community School and the First Nations and Endangered Languages Program at the University of British Columbia, the Hiłzaqv Language and Culture Mobilization Partnership <https://heiltsuk.arts.ubc.ca/> was established. The Hiłzaqv language is spoken by the Heiltsuk Nation whose traditional territory includes the administrative centre of Bella Bella in Northern British Columbia, Canada.

Despite being critically endangered with only 4.7% of the population classified as either fluent or semi-fluent speakers, the language has a deep history of community-led documentation (Carpenter et al., 2016) and a vibrant community of dedicated and accomplished learners, which comprise 11% of the total population according to the First Peoples' Heritage and Language Council of Canada (First People's Cultural Council (FPCC), 2014).

2.3 Interim Strategies

The orthography for the Heiltsuk language, designed by linguist John Rath, uses many characters that are not part of the standard ASCII character set. For example, vowels may indicate high tone (through an acute accent, as in **á**), resonants may be vocalic (marked with an underdot), carry a high tone (marked by an acute accent), or be glottalized (marked by an apostrophe or even a combination of these as in **ḿ**, **m̓̀**, **n̓**, **ń** or **l̓́**). An **ɬ** is used to represent a voiceless lateral fricative and **ƛ** is used to represent a lateral stop. In total, the orthography uses 44 different characters that lie outside of the standard ASCII character set. Before Unicode was in widespread use, Heiltsuk language users still needed ways to display characters like **ǧ**. Heiltsuk is not alone in this dilemma, and many language communities and "linguists have devised a variety of ingenious solutions" (Bird & Simons, 2002) to overcome similar challenges. We are aware of three strategies used by members of the Heiltsuk community to achieve this: textual images, transliteration, and font-encodings.

2.3.1 Textual Images

When the Heiltsuk language could not be easily represented digitally, some individuals resorted to taking photographs of hand-written text and sharing the image file with others. Such an approach—while creative, immediate and effective—precludes the possibility of leveraging any digital text-processing tools, and makes resulting communication somewhat cumbersome. Indeed, this approach was used and can still be seen in the welcome greeting of the Heiltsuk Cultural Education Centre's website www.hcec.ca. Using textual images is time intensive, error prone, not easily scalable, and contingent on reliable and robust internet speeds. These techniques avoid both the power and pitfalls of text-editors, choosing instead to engage with the web and social media directly by assembling and curating image files of hand-written text.

2.3.2 Transliteration

In some cases, when an orthography has only relatively few characters that are not available in ASCII, ad-hoc transliterations designed to be used exclusively in digital contexts have been created, such as the one developed by linguist John Rath for Heiltsuk. For example, if the only required character outside of ASCII is schwa (ə) in a given language, then a community linguist might opt to simply use @ in place of schwa. While this can be quite an effective interim solution provided that not too many additional characters are required, it results in a symbol like @ having more than one meaning, which can confuse speakers, digital text processing tools as well as language-independent software and search engines. Along with being visually ad-hoc, requiring additional learning, and being potentially visually jarring and confusing, such an approach also burdens speakers with having to familiarize themselves with two distinct if related writing systems.

2.3.3 Font Encoding

Another approach to representing unsupported characters before Unicode support was available involved the development of language-specific fonts, usually referred to as "font-encodings", "font-hacks" or "font-encoded orthographies." The Heiltsuk orthography has two distinct font encodings: Heiltsuk Doulos and Heiltsuk Times. These fonts were specifically designed to render the 44 non-ASCII/ISO 8859-1 characters needed by the Heiltsuk orthography. For example, the Heiltsuk Doulos font was created to deliberately disregard 8-bit encoding ISO 8859-1's stipulation that 10101001 should render as ©, and instead render this sequence as **ǧ**. In order for the characters to be viewed, the customized font must be correctly installed. Such a work-around allows almost any character to be represented regardless of the underlying encoding. While this strategy works well if both the author and reader have the font installed, one result is that the language cannot be mobilized on social media or through web applications. Without the required font installed, 10101001 will appear as © and be illegible to users. With the development of UTF-8 encoding, it is now possible to type **ǧ** and expect that it will render correctly in most mainstream fonts. With this development, there is no longer a need for font encodings, despite their having played an essential interim role for Heiltsuk and for many other languages around the world (cf. Hall, Ghimire & Newton 2009 on the Preeti font for Nepali).

3. Implementation in the Heiltsuk Community

Within the Heiltsuk community in Bella Bella and beyond, many text documents (in both txt and Microsoft Word docx formats), Excel spreadsheets, PowerPoint presentations and lessons had been composed using the Heiltsuk Doulos and Heiltsuk Times fonts. Textual images were also fairly widely used on social media platforms.

The first goal of the Hiłzaqv Language and Culture Mobilization Partnership was to develop a cross-platform Unicode input system and keyboard to replace the font-

encoded fixes of Heiltsuk Doulos and Heiltsuk Times. Within a matter of minutes of the Heiltsuk Unicode

Figure 1 Rory Housty tweeting in Heiltsuk.

keyboard being released, community members began tweeting in the language, opening up a vibrant, online digital space where the language could be shared, as seen above in Fig. 1.

All earlier Heiltsuk digital materials, however, remained in the non-Unicode Heiltsuk Doulos and Heiltsuk Times fonts, making them effectively un-readable to users without the fonts installed and much harder to share. *Convertextract* was used to convert over 70 megabytes of text files from Heiltsuk Doulos and Heiltsuk Times to Unicode, including eight PowerPoint presentations, an Excel spreadsheet dictionary containing 10,005 entries, and 103 Microsoft Word files including several books used by school teachers in the Bella Bella Community School Native Language Program. In total, these files contained 103,056 characters of text which, assuming a typing rate of 100 characters/minute would have taken over 17 hours to retype, not including the time it would take to re-format the files.

4. Convertextract: Implementation

Given the context described in Section 2.2 above, it was apparent that the Heiltsuk community could make good use of a tool to perform a series of character conversions to upgrade text to Unicode standards from either the ad-hoc transliteration strategy described in Section 2.3.2 or from the font-encoding strategy described in Section 2.3.3. The following three sections describe implementations of *Convertextract* as of version 1.3.

4.1 Python CLI

The *Convertextract* command line tool is a MIT licensed Python library built from a fork of Dean Malmgren's Textract library.[2] Textract is a library that extracts text from a wide variety of different file formats. Leveraging this work, *Convertextract* performs a specific list of find/replace transformations on any source text, and saves a new converted file without altering the style formatting of the original document (font size, underlining, boldness etc).

As *Convertextract* expects to be converting from a 'hacked' font, it delivers the converted text in Times New Roman by default, although any other font may be specified. Out of the box, *Convertextract* currently supports three conversions: from Heiltsuk Doulos, Heiltsuk Times and Tsilhqot'in Doulos to Unicode. *Convertextract* also supports user-defined conversions which can be described in an Excel document passed as an argument to *Convertextract*. The correct ordering of each substitution is essential to producing the correct output. In order to prevent the incorrect sequencing of substitutions, they are ordered according to their length from longest to shortest.

The Python Library can be called either in a Python script or directly through the command line. Documentation on how to install and use the command line tool is available on the public repository[3]. As of version 1.3, plain text files (txt), Microsoft Word documents (docx), Excel spreadsheets (xlsx) and PowerPoint presentations (pptx) are all accepted file formats for *Convertextract*.

4.2 Web

As many potential users of *Convertextract* might not be familiar with command line interfaces, a web tool that accomplishes the same task was developed and released for Heiltsuk Doulos and Heiltsuk Times. The tool operates by allowing a user to either upload a file to be converted (and subsequently returning a converted file), or paste or type text directly into a text input box and then select a conversion. Text typed in the input box is directly and instantaneously transformed using the chosen conversion. The web UI has also been responsively designed for mobile use.

4.3 Chrome Extension

Convertextract has also been released as an extension for the Chrome web browser. This implementation allows the instantaneous conversion of published websites. While online publishing using font-encoded orthographies online is fairly uncommon (as font-encodings are not typically supported by web browsers), transliterations do sometimes appear online, as was common with the Preeti font for Nepali, and may need to be converted. The Chrome extension is likely to be most useful for converting text between multiple orthographies that are in circulation and use for the same language.[4]

4.4 Limitations

Convertextract has a number of requirements in order to work properly. First, conversions must be able to be applied independent of context. For example, consider a hypothetical orthography for a language in which the underlying vowel **ɔ** is represented as **ɔ** before rounded consonants, but as **ʌ** before unrounded consonants. As the

[2] https://github.com/deanmalmgren/textract.
[3] https://github.com/roedoejet/convertextract.
[4] As with the Inuktitut Web Page Transliterator http://www.inuktitutcomputing.ca/Transliteration/webpage/info.php?lang=en

distribution of ʌ is predictable, when designing a new orthography, a decision might be made to represent both as @. When, at a later date, both symbols become renderable in most fonts thanks to the widespread rollout of Unicode, the community wishes to use a tool like *Convertextract* to convert documents written using @ to the correct and original orthography that uses ʌ and ɔ. Unfortunately, in the current version of *Convertextract*, no functionality exists that would support context-dependent conversions. Instead of using methods like regular expressions which can provide this functionality, substitutions are defined in spreadsheets. This was done for the ease of adding additional languages as described in Section 4.5 and because context-dependent conversions were not required by Heiltsuk.

At present, *Convertextract* is not able to convert documents that contain multiple languages or writing systems within the same document. Depending on demand, these features could be developed and included in future releases.

4.5 Adding support for other font encodings

Convertextract was designed to be easily extended to other font encodings without requiring a significant investment of time, energy or computational expertise. If, for example, it is known that a font-encoding like Heiltsuk Doulos uses © to represent ğ, then that correspondence can be entered into a spreadsheet with © in one column and ğ in the other. By including a relative path to that spreadsheet as an argument, *Convertextract* will order and perform the same character substitutions using that spreadsheet. In that way, adding support for more font encodings can be as simple as preparing a spreadsheet. Indeed, this is how support was implemented by the Tsilhqot'in National Government for the Tsilhqot'in Doulos font encoding. The same lookup table that is used to inform the command line tool can also be used to build the web implementation described in Section 4.2 or the Chrome extension converters described in Section 4.3. Additional Excel lookup tables for font encodings are welcome and may be submitted either as pull requests or by email.

As illustrated in Section 3 above, we hope that the combination of supporting popular Microsoft Office formats and easy integration of custom lookup tables will result in *Convertextract* being used by more communities to save time preserving and mobilizing their digital language resources in years to come.

5. Conclusion

We have introduced a tool designed to convert text, Word, Excel and PowerPoint documents composed in non-Unicode compliant, customized and now legacy fonts into Unicode without altering existing file formatting and styles. This tool, while technically uncomplicated, has the potential to greatly reduce the hours previously required of communities to manually re-type and re-style these files in order to preserve their digital resources and broaden access in the information age.

6. Acknowledgements

We thank John Nenniger for his work developing the Chrome extension and Jennifer Carpenter for her invaluable edits and suggestions for improvements to this paper. We are grateful for funding from SSHRC Partnership Grant #895-2012-1029 (PI: Marianne Ignace) and SSHRC Knowledge Synthesis Grant #421-2015-2076 (PI: Mark Turin).

7. References

Bird, S., & Simons, G. (2003). Seven dimensions of portability for language documentation and description. *Language*, 79(3), 557-582.

Bowern, C. (2015). *Linguistic fieldwork: A practical guide*. Springer.

Carpenter, J., Guerin, A., Kaczmarek, M., Lawson, G., Lawson, K., Nathan, L. P., & Turin, M. (2016). *Digital Access for Language and Culture in First Nations Communities*. Vancouver, BC.

Coulmas, F. (2003). Writing systems. *An introduction to their linguistic analysis*, 249-268.

Davis, M. (2008). Moving to Unicode 5.1. Retrieved January 8, 2018, from https://googleblog.blogspot.com/2008/05/moving-to-unicode-51.html

First People's Cultural Council (FPCC). (2014). *2014 Report on the Status of B.C. First Nations Languages*. Brentwood Bay, B.C.

Gorn, S., Bemer, R. W., & Green, J. (1963). *American standard code for information interchange. Communications of the ACM*, 6(8): 422–426.

Hall, P., Ghimire, G., & Newton, M. (2009). Why don't people use Nepali language software?. *Information Technologies & International Development*, 5(1): 65-79.

Jones, M. C., & Mooney, D. (2017). *Creating orthographies for endangered languages*. Cambridge University Press.

Mortensen, D. R., Littell, P., Bharadwaj, A., Goyal, K., Dyer, C., & Levin, L. (2016). Panphon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 3475-3484).

Pine, A, & Turin, M. (2017). Language Revitalization. *Oxford Research Encyclopedia of Linguisitics*. Oxford University Press.

Rath. J. C. (1985). *Ways of Writing*. Heiltsuk Cultural Education Centre.

Rath, J. C. (1981). *A practical Heiltsuk-English dictionary with a grammatical introduction (Vol. 2)*. National Museums of Canada.

Rath, J. C. (n.d.). *Elements of Heiltsuk Grammar*. Bella Bella: Heiltsuk Cultural Education Centre.

Yergeau, F. (2003). *UTF-8, a transformation format of ISO 10646*.