Endangered Data for Endangered Languages: Digitizing Print dictionaries[*]

Michael Maxwell
Aric Bills
University of Maryland
{mmaxwell,abills}@umd.edu

1 Introduction

This paper describes on-going work in dictionary digitization, and in particular the processing of OCRed text into a structured lexicon. The description is at a conceptual level, without implementation details.

In decades of work on endangered languages, hundreds (or more) languages have been documented with print dictionaries. Into the 1980s, most such dictionaries were edited on paper media (such as 3x5 cards), then typeset by hand or on old computer systems (Bartholomew and Schoenhals. 1983; Grimes 1970). SIL International, for example, has nearly 100 lexicons that date from their work during the period 1937–1983 (Verna Stutzman, p.c.).

More recently, most dictionaries are prepared on computers, using tools like SIL's Shoebox (later Toolbox) or Fieldworks Language Explorer (FLEx). These born-digital dictionaries were all at one time on electronic media: tapes, floppy diskettes, hard disks or CDs. In some cases those media are no longer readable, and no backups were made onto more durable media; so the only readable version we have of these dictionaries may be a paper copy (cf. Bird and Simons 2003; Borghoff et al. 2006). And while paper copies preserve their information (barring rot, fire, and termites), that information is inaccessible to computers. For that, the paper dictionary must be digitized.

A great many other dictionaries of non-endangered languages are also available only in paper form.

It might seem that digitization is simple. It is not. There are two approaches to digitization: keying in the text by hand, and Optical Character Recognition (OCR). While each has advantages and disadvantages, in the end we are faced with three problems:

1. Typographic errors;
2. Conversion from the dictionary's visual layout into a lexicographically structured computer-readable format, such as XML; and
3. Converting each dictionary's idiosyncratic structure into some standard tagging system.

This paper deals mostly with the second issue (but see section 6 about the first issue, and section 7 about the last issue). The information structure in a print dictionary is represented mostly implicitly, by formatting: white space, order, font and font style, and occasionally by numbering, bullets, or markup (such as 'Ex.' for an example sentence). The task addressed in this paper is that of converting this implicit information into explicit tags for lexical entries, parts of speech, glosses, example sentences, and so forth.

At present, the tools described here are very much in the development phase, and must be run on the command line. Ideally, we would go on to create a Graphical User Interface (GUI) which would serve as a front end to these tools. Whether we will achieve that goal within the limits of our funding remains to be seen. But even if we do, we do not envision the task of converting paper dictionaries into electronic databases to be a job that most field linguists will want to, or even should, undertake. Indeed, there are a limited number of paper dictionaries that need to be converted (all modern dictionaries, we believe, are being created electronically). Instead, we envision the user community for this software as being composed of a small number of experts in lexicography, who can learn enough about a particular language to reliably interpret the format of a dictionary of that language, and can therefore act as a sort of conservation corps for legacy dictionaries.

2 Print Dictionary Format

Figure 1 shows two lexical entries in a Tzeltal-English dictionary (Cruz, Gerdel, and Slocum 1999; this is a more detailed version of Slocum and Gerdel 1976). In this dictionary, headwords of major entries appear at the left margin of a column, in bold font. They are followed by a part of speech, identifiable by position, italic font, and the fact that they come from a small class of tokens ('s', 'vt', etc.). The Spanish glosses follow in regular font; individual glosses are separated by commas (not shown in this snippet). Cross references to other entries in the dictionary are bolded and preceded by the word 'Véase' (Spanish for "see"). Multiple senses (not shown in this example) are indicated by Arabic numerals. The entire entry is formatted with hanging indent. Finally, subentries are further indented, and otherwise follow much the same formatting as main entries, with a subset of the information (e.g. subentries lack part of speech).

[*] This material is based upon work supported by the National Science Foundation under grant number BCS1644606.

Figure 2 shows a lexical entry in SIL's Muinane-Spanish dictionary (J. W. Walton, J. P. Walton, and Buenaventura 1997).[1] As in the Tzeltal dictionary, headwords of major entries appear at the left margin of a column, in bold font. They are followed by a part of speech, identifiable by position, by a lighter (non-bold) upright font, and again by the fact that they come from a small limited class of tokens ('s.', 'v.i.', etc.), all ending in a period. The glosses follow in the same font, and consist of Spanish words; individual glosses are separated by commas, and the list of glosses ends with a period. If the list of glosses extends beyond the first line, it is indented by a bit over an em relative to the headword. Example sentences and their translations appear on separate lines, indented by perhaps two ems; the Muinane text of example sentences is in italics, while the Spanish translation is in an upright font. Within the Muinane text, the inflected form of the headword is underlined. Finally, subentries (as in the right-hand column) are indented by about one em, and otherwise follow the same formatting as main entries. Irregular inflected forms (not shown in the figure) are given in italics, preceded by their inflectional category, with square brackets enclosing the grammatical information and the irregular form. Elsewhere in this dictionary, multiple senses are provided with Arabic numerals in a bold font.

Note that in both of these dictionaries, lexical entries are represented as paragraphs with hanging indent. Significantly, there is no extra vertical space between these paragraphs; this makes automatic inference of lexical entries difficult, a problem to which we now turn.

3 Inferring Lexical Entries

In our experiments, we are concentrating on how the OCR form of a dictionary can be converted into a lexical database. We are not concerned here with the OCR process itself, which we treat as more or less a black box. Our reason for doing so should be obvious; much work has been done on converting paper documents into electronic form as streams of text, searchable by matching strings of characters. While some work has been done on formatting the streams of text to correspond with the position of that text on paper, so that a human user can be shown the location on the page of their search terms, apart from the issues of processing tables, very little work has been done on converting the OCR output of visually structured documents into structured databases. This is particularly true for dictonaries, which vary greatly in their format. The post-OCR conversion process is thus ripe for exploration, and is our topic here.

In order, then, to concentrate our effort on the conversion from OCR output to lexical database, rather than cutting up paper dictionaries and feeding them through scanners, we are using for our experiments several dictionaries which are available on-line in image format:

- The Tzeltal-English dictionary mentioned above (Cruz, Gerdel, and Slocum 1999)
- The Muinane-Spanish dictionary mentioned above (J. W. Walton, J. P. Walton, and Buenaventura 1997)
- A Cubeo-Spanish dictionary (Morse, Jay K. Salser, and Salser 1999)

All three dictionaries use Latin script, although we did encounter issues with recognizing some characters (see section 6). Non-Latin scripts would of course introduce other issues, although most of those issues would have to do with training or adapting an OCR system to recognize those scripts, research which others have tackled (see e.g. Govindaraju and Setlur 2009; Smith, Antonova, and Lee 2009).

We converted the online images of these dictionaries into raster images, and then ran an OCR program on the result.

Most modern OCR systems include as one of their output formats the 'hOCR' form (`https://kba.github.io/hocr-spec/1.2/`. This XML format tags hypothesized words, lines, paragraphs, and columns, and provides additional information such as position on the page; some information on font and font style is also extractable. We have been using the open source program Tesseract (`https://github.com/tesseract-ocr`). An excerpt from the hOCR output of Tesseract showing the structure corresponding to the first line of figure 1 appears in figure 3.[2]

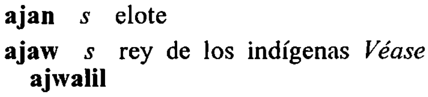
In this output, the 'bbox' indicates the coordinates of rectangles occupied by individual tokens, lines, paragraphs etc. Notice that this lexical entry has been correctly tokenized into the Tzeltal word 'ajan', the part of speech 's', and the Spanish gloss 'elote' (corncob), all as part of an `ocr_line`. Unfortunately, although lexical entries are represented as hanging indent paragraphs in this dictionary, neither the `<div>` (division) nor the `<p>` (paragraph) elements reliably parse this structure.[3] This is also true of the OCR output of the Cubeo dictionary which we have been working with (Morse, Jay K. Salser, and Salser 1999). It seems in general that we cannot rely on Tessearact's division into paragraphs, i.e. lexical entries.

Hence the first task for the user is to define the general shape of the page, including the approximate position of headers and/or footers, and columns. The user also needs to define the shape of a lexical entry, so that individual lexical entries can be parsed out from the

[1] A differently formatted version of this dictionary is available on-line: `http://www-01.sil.org/americas/colombia/pubs/MuinaneDictBil_49558-with-covers.pdf`.

[2] This has been simplified by removing some attributes, and formatted to clarify the structure.

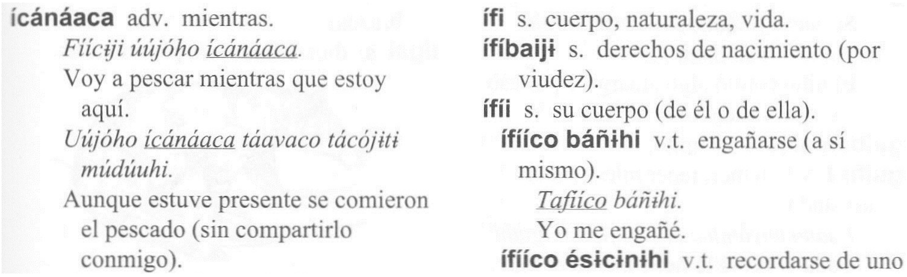
[3] Also, while the OCR correctly captured the italicization of the part of speech, it usually fails to detect bolding, as seen for the headword.



ajan *s* elote

ajaw *s* rey de los indígenas *Véase* **ajwalil**

Figure 1: Some lexical entries from Tzeltal print dictionary



ícánáaca adv. mientras.
Fíícɨji úújóho ícánáaca.
Voy a pescar mientras que estoy aquí.
Uújóho ícánáaca táavaco tácójɨtɨ múdúuhi.
Aunque estuve presente se comieron el pescado (sin compartirlo conmigo).

ífi s. cuerpo, naturaleza, vida.

ífíbaijɨ s. derechos de nacimiento (por viudez).

ífii s. su cuerpo (de él o de ella).
ífííco báñɨhi v.t. engañarse (a sí mismo).
Tafííco báñɨhi.
Yo me engañé.
ífííco ésɨcɨnɨhi v.t. recordarse de uno

Figure 2: Lexical entries from Muinane dictionary

```
<div class="ocr_carea"... title="bbox 526 577 1222 665">
  <p ... title="bbox 525 1459 1230 1655">
   <span class="ocr_line"... bbox 526 1459 794 1501...>
      <span class="ocrx_word" ... bbox 526 1460 607 1501...>
        ajan
      </span>
      <span class="ocrx_word" ... bbox 650 1471 667 1491...>
        <em>s</em>
      </span>
      <span class="ocrx_word" ... bbox 710 1459 794 1492...>
        elote
      </span>
   </span>
   ...
  </p>
</div>
```

Figure 3: Extract from hOCR output of Tzeltal dictionary (some details ellipted)

columns. A mechanism is therefore required to allow the user to define the shape of these entities, and then to apply this shape to the hOCR data so that successive lines (approximately) matching that shape can be either discarded (headers and footers) or grouped into lexical entries. This post-processing step is the first part of what we will call Stage 1.

There are several other steps needed to reliably group lines output by Tesseract into lexical entries. Tesseract does appear to correctly capture the division of lines between columns (without incorrectly joining lines across columns), but it does not create an actual column structure. This must be inferred in post-processing, so that lexical entries which are broken across columns can be re-joined. (Page breaks are of course correctly parsed.)

We have constructed a program (written in Python) to do this step of inferring the division of the OCRed text into components. A human supplies several parameters to the program, including:

- The number of columns on a page, and their left-hand margins relative to the page.[4]
- Any information that spans columns. These are typically headers and/or footers, but see below for other information which may fall into this category.
- Number of tab stops in lexical entries. Lexical entries are often formatted as hanging indent paragraphs; in such a case, the indent would be the first tab stop. For some dictionaries there may be additional tab stops (indents), as in the Muinane dictionary of figure 2. At present the user must also supply an approximate measure for each indent, but we hope to eliminate the need for that.

The output of Stage 1 is then an XML file whose structure below the root element consists of a sequence of inferred lexical entries. Within these lexical entries, the structure is a sequence of `<span class='ocr_line'...>` elements, unchanged from the original input.

Nevertheless, we expect there to be a need for manual intervention in this inference step. Figure 4 shows several potential instance of this in the Cubeo dictionary. First, while Tesseract is reasonably good at recognizing that images (such as the picture of the tree in the second column) are not text, and ignoring them, the captions of such figures are generally parsed as paragraphs, and must be omitted from the machine-readable dictionary.[5]

Second, observe that the letter 'B' constitutes a spanning element across both columns. That is, while elsewhere columns are usually continued at the top of the page, in this case, columns containing entries beginning with the letter 'A' appear above this spanning element, while the columns for words beginning with 'B' appear below this.

Finally, note that the upper first column, down to the spanning 'B', consists entirely of the continuation of a lexical entry on the previous page, and is thus indented to the first tab stop. In fact, this lexical entry continues in indented form onto the upper right-hand column. If our program incorrectly inferred that this indent is the level at which lexical entries start, instead of the level at which the second and following lines of a lexical entry are indented, then our program will infer that the upper columns consist of a number of entries, rather than being a single entry with some subentries.[6]

Our approach to manual correction is intended to allow the user to make corrections in a way that is preserved during processing, even during earlier stages of processing. The motivation for this is as follows: suppose that the user did not notice problems resulting from the layout of 4 until much additional processing and manual annotation had taken place. Insofar as this subsequent manual annotation is correct, we wish to preserve it, even if previous stages of automatic processing need to be re-applied. (Automatic annotation, on the other hand, can be easily and cheaply re-done, hence does not require preserving–indeed the reason for redoing previous automatic stages would presumably be to introduce changes in their functioning.) Our programs therefore preserve human-supplied annotations so that the annotated file can be re-run through earlier steps of processing without loss of the user's work.[7]

We must therefore provide a means for human correction of the output at this point in the pipeline, and do so in a way that does not lose any subsequent annotation, particularly where that subsequent annotation is done by humans.

4 Inferring Substructure

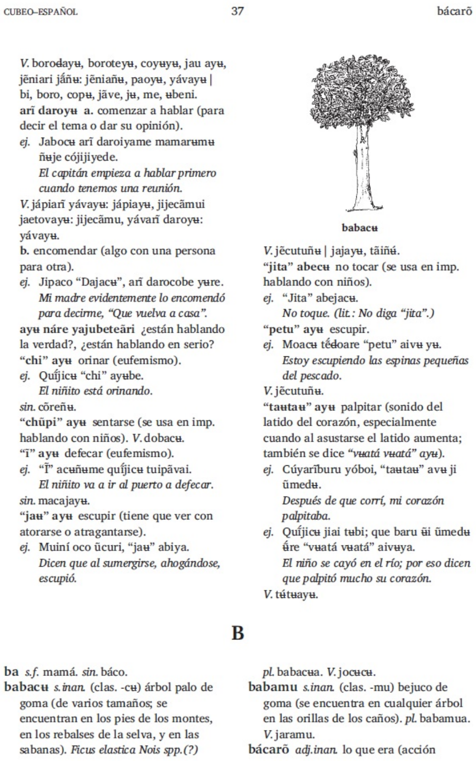
Once the boundaries between lexical entries have been tagged, the next step is to build a finite state grammar giving the order of elements in the dictionary (referred to by lexicographers as the dictionary's 'microstructure'). This can be written as a regular expression by observing a sampling of lexical entries. Conceptually,

[4] This may of course differ for left- and right-hand pages. For dictionaries whose headwords are written in right-to-left scripts, this will probably need to be the right-hand margin of the columns.

[5] If pictures are to be included in the machine-readable dictionary, we assume that the process of capturing those images and linking them to an appropriate entry will be separate from the process of converting the text portion of the dictionary into database form.

[6] In fact the program looks for indents relative to a bounding box around all the text of the page; so in this example, the indents would probably be correctly determined, since the lower left-hand column establishes the full width of the text. However, if the long lexical entry had appeared on a page that did not have a spanning letter, the page width might have been incorrectly inferred.

[7] Another way to preserve human work would be to use stand-off annotation; but that would require ensuring that the stand-off annotation pointed only to structures present in the earliest stages of processing.



CUBEO–ESPAÑOL 37 bácarõ

V. borođayʉ, boroteyʉ, coyʉyʉ, jau ayʉ, jẽniari jẵñʉ: jẽniañʉ, paoyʉ, yávayʉ | bi, boro, copʉ, jãve, jʉ, me, ʉbeni.

arĩ daroyʉ **a.** comenzar a hablar (para decir el tema o dar su opinión).

ej. Jabocʉ arĩ daroiyame mamarʉmʉ ñʉje cójijiyede.
El capitán empieza a hablar primero cuando tenemos una reunión.

V. jápiarĩ yávayʉ: jápiayʉ, jijecãmui jaetovayʉ: jijecãmu, yávarĩ daroyʉ: yávayʉ.

b. encomendar (algo con una persona para otra).

ej. Jipaco “Dajacʉ”, arĩ darocobe yʉre.
Mi madre evidentemente lo encomendó para decirme, “Que vuelva a casa”.

ayʉ náre yajubeteãri ¿están hablando la verdad?, ¿están hablando en serio?

“chi” ayʉ orinar (eufemismo).

ej. Quĩ́jicʉ “chi” ayʉbe.
El niñito está orinando.

sin. cõreñʉ.

“chũpi” ayʉ sentarse (se usa en imp. hablando con niños). *V.* dobacʉ.

“ĩ” ayʉ defecar (eufemismo).

ej. “Ĩ” acʉñʉme quĩ́jicʉ tuipãvai.
El niñito va a ir al puerto a defecar.

sin. macajayʉ.

“jaʉ” ayʉ escupir (tiene que ver con atorarse o atragantarse).

ej. Muiní oco ũcuri, “jaʉ” abiya.
Dicen que al sumergirse, ahogándose, escupió.

babacʉ

V. jẽcutuñʉ | jajayʉ, tãiñʉ́.

“jita” abecʉ no tocar (se usa en imp. hablando con niños).

ej. “Jita” abejacʉ.
No toque. (lit.: No diga “jita”.)

“petu” ayʉ escupir.

ej. Moacʉ tẽ́đoare “petu” aivʉ yʉ.
Estoy escupiendo las espinas pequeñas del pescado.

V. jẽcutuñʉ.

“taʉtaʉ” ayʉ palpitar (sonido del latido del corazón, especialmente cuando al asustarse el latido aumenta; también se dice “*vʉatá vʉatá” ayʉ*).

ej. Cúyarĩburu yóboi, “taʉtaʉ” avʉ ji ũmedʉ.
Después de que corrí, mi corazón palpitaba.

ej. Quĩ́jicʉ jiai tʉbi; que baru ʉ̃i ũmedʉ ʉ̃́re “vʉatá vʉatá” aivʉya.
El niño se cayó en el río; por eso dicen que palpitó mucho su corazón.

V. tʉ́tʉayʉ.

B

ba *s.f.* mamá. *sin.* báco.

babacʉ *s.inan.* (clas. -cʉ) árbol palo de goma (de varios tamaños; se encuentran en los pies de los montes, en los rebalses de la selva, y en las sabanas). *Ficus elastica Nois spp.(?)* *pl.* babacʉa. *V.* jocʉcʉ.

babamu *s.inan.* (clas. -mu) bejuco de goma (se encuentra en cualquier árbol en las orillas de los caños). *pl.* babamua. *V.* jaramu.

bácarõ *adj.inan.* lo que era (acción

Figure 4: A page of the Cubeo print dictionary

for example, in the two lexical entries of figure 1, there are two structures:[8]

```
Headword POS GlossWord
Headword POS GlossWord GlossWord \
   GlossWord GlossWord `Véase' XRefWord
```

Hence the task for the user is to create a finite state grammar representing these two lexical entries, and to incrementally expand the grammar to match additional lexical entries. A grammar (still conceptual) combining the above two structures would be the following:

```
Headword POS GlossWord+ \
    (`Véase' XRefWord)?
```

Here we have used Kleene plus to indicate that the GlossWord can be repeated one or more times, parentheses for grouping, and '?' to indicate optionality.

The finite state grammar described above can be thought of as a finite state acceptor (FSA). But in order to transmute the hOCR file into a lexicographically structured dictionary, a finite state transducer (FST) is used to convert the elements found in the hOCR file into the elements needed for the dictionary. For example, what we have called 'Headword' in the above conceptual grammar is represented in the input by `<span class='ocrx_word'...> ... <strong> <span>`; this hOCR structure must be converted into lexicographic elements like `<form>...</form>`.[9] Similarly, the transducer converts a sequence of conceptual glosswords, each represented by a `<span class='ocrx_word'...>...</span>`, into an element like

```
<def>
  <cit type="translation" xml:lang="sp">
       <quote>...</quote>
    </cit>
</def>
```

Further refinement of this grammar will capture the fact that the POS is one of a small number of possible words ('s', 'vt', etc.), and will take account of punctuation which (in some dictionaries) defines the end of certain elements.

The grammar is applied to the output from Stage 1 to parse as many lexical entries as it can; unparsed lexical entries are passed through as-is. Examination of unparsed lexical entries will reveal other patterns that need to be captured, which may include subentries, multiple senses, etc. semantic restrictions or explanations (such as 'trampa (compuesta de madera)' (= "trap (composed of wood)")), etc.

Recall that the hOCR representation tags each line of the original print dictionary. We have retained these tags in the output of Stage 1 because some line breaks may be relevant to parsing. (For example, subentries in the Tzeltal dictionary start on new lines.) However, line breaks can also be due to typesetting constraints. The FST therefore allows newlines between any two tokens, outputting an epsilon (nothing).

Application of the FST grammar to the output of Stage 1 produces a derived XML file in which (most) lexical entries have been transformed from a sequence of lines, each composed of a sequence of tokens, into lexicographically tagged entries. This constitutes the output of Stage 2.

The output of Stage 2 may contain a residue of lexical entries that do not parse. This may imply deficiencies in the grammar, but it may instead be a result of boundaries between lexical entries which have been incorrectly OCRed or inferred in Stage 1 processing. The input structure can be modified to correct such errors, and the grammar rules re-applied until some desired proportion of the lexical entries parse successfully.

Finally, the human may choose to parse some of these non-parsing entries–particularly complex ones–by hand, a process which we touch on in the next section.

5 Post-editing

Ideally, the grammar developed in section 4 will correctly account for all the lexical entries inferred in section 3. In practice, this may not always be the case. One source of noise which is likely to prevent full coverage is inconsistencies in the print dictionary; another is typos (treated in section 6, below) which are severe

[8] For purposes of this paper, we ignore the subentry.

[9] The `<strong>` tag represents bold font in this dictionary. For output, we use the Text Encoding Initiative (TEI) schema for dictionaries, see `http://www.tei-c.org/release/doc/tei-p5-doc/en/html/DI.html`.

enough to prevent correct parsing. But probably the main reason for incomplete coverage will be unique (or nearly unique) complex lexical entries, which are not erroneous per se, but which are enough unlike "normal" lexical entries that they resist parsing by normal means. Thus the output of Stage 2 may include some lexical entries in the line oriented format output by Stage 1 (figure 3), rather than as lexical entries parsed into lexical fields. At some point, it becomes more productive to convert such anomalous entries by hand, rather than further modifying the grammar to account for them.

We therefore allow post-editing of the output of Stage 2, so that the final output contains only those lexicographic elements appropriate to a dictionary database, without any of the formatting elements output by the OCR system. We are currently using a programmer's editor to do this post-editing; we may later substitute an XML editor specialized for dictionaries, which has been developed by David Zajic and others in our group.

6 Typo correction

As mentioned in section 1, OCR systems (and human typing) will produce typographic errors. This problem is particularly acute in dictionaries of minority languages, since it is unlikely that one can find an off-the-shelf OCR model tuned for such a language. Minority languages may also introduce unusual characters; Cubeo, for example, has a single (but frequent) verbal root that contains the only phonemically contrastive instance of a voiced interdental fricative, and the Cubeo dictionary writes it as a barred 'd', a letter not recognized by our OCR system.[10] The fact that such dictionaries are likely to be bilingual further exacerbates the problem, since at the time the OCR process runs, there is no indication of which words are in which language; so even if one did have a character-based language model to help discover typos, the system would not know which words to apply that to.

But of course the latter problem is solved once we have converted the OCR output into an XML dictionary file; we then know which fields encode which languages. At that point, various techniques can be employed to find possible typos, whether using a standard spell corrector for the glossing language, creating a character-based language model of the minority language, or manually searching in the minority language fields for particular sequences of characters (such as characters characterizing that one Cubeo verb root, using an ordinary 'd' for search in place of the desired barred-d). We therefore treat typo correction as a nearly last step, rather than an initial step.

7 Conversion to standard XML schemas

The structure of the output of the processing described above should match the *conceptual* structure of the original print dictionary. For reasons which will not be discussed here, this is probably not the optimum structure for electronic dictionaries. For example, while print dictionaries frequently treat phrasal entries as subentries of one of the words found in the phrase, it is more common in lexicographic databases to treat phrasal entries as separate lexical entries, linked to each of the words of which they are composed. A view of a phrasal entry as a subentry of these words can then be created on the fly.

The Lexical Markup Framework (LMF, Francopoulo 2013; ISO TC37 2008) is a meta-schema finding increasing use for electronic dictionaries. There is no comparable standard for print dictionaries; however, the Text Encoding Initiative (TEI) Guidelines (TEI Consortium 2016), particularly chapter 9, contain an ample set of tags that should be usable for virtually any dictionary. While we cannot treat here the conversion between a custom XML file derived from a particular print dictionary (which might conform to the TEI) and other standards (such as LMF), we do consider that conversion to be a recommended practice.

8 Related work

To the best of our knowledge, there has been little published about how to convert OCRed (or hand-typed) lexical data into a database format. From what we can ascertain from personal communication and examination of errors in converted dictionaries, what is usually done is to process such data using a chain of scripts, using regular expressions to convert whatever information is available into a target format, generally SGML or (more recently) XML.

There is one documented project that developed a more general means of importing OCRed dictionaries. This is the earlier University of Maryland project, BRIDGE (Karagol-Ayan, D. Doermann, and Dorr 2003; Ma et al. 2003). Unfortunately, the software developed under that project had multiple dependencies which are no longer readily available, so the tools are not working; our attempts to re-create the workflow using more modern, freely available software did not succeed. Also, while that project relied heavily on machine learning, we are creating a more manual process, which we expect to be easier to maintain and modify.

In a series of papers, Zajic and his collaborators at our institution have explored error detection in digital dictionaries (Zajic, Bloodgood, et al. 2015; Zajic, D. S. Doermann, Bloodgood, et al. 2012; Zajic, D. S. Doermann, Rodrigues, et al. 2013; Zajic, M. Maxwell, et al. 2011). This work will inform our work on the parsing of lexical entries (4); as mentioned in section 5, we may also incorporate the specialized XML editor that group has developed.

[10] Morse and M. B. Maxwell 1999, p. 5 treats this as virtually allophonic, but the Cubeo dictionary writes it in this one morpheme with a distinct alphabetic character.

9 Availability

We will make our software available as open source, although the exact license has not been determined.

References

Bartholomew, Doris A. and Louise C. Schoenhals. (1983). *Bilingual dictionaries for indigenous languages*. Mexico: Summer Institute of Linguistics.

Bird, Steven and Gary Simons (2003). "Seven dimensions of portability for language documentation and description." In: *Language* 79.3, pp. 557–582.

Borghoff, U.M., P. Rödig, J. Scheffczyk, and L. Schmitz (2006). *Long-Term Preservation of Digital Documents: Principles and Practices*. Berlin: Springer.

Cruz, Manuel A., Florence L. Gerdel, and Marianna C. Slocum (1999). *Diccionario tzeltal de Bachajón, Chiapas*. Serie de vocabularios y diccionarios indígenas "Mariano Silva y Aceves" 40. Coyoacán, D.F., Mexico: Instituto Lingüístico de Verano, A.C. URL: http://www.sil.org/system/files/reapdata/52/85/76/52857610164780871251544555610851968393/S040_DicTzeltalFacs_tzh.pdf.

Francopoulo, Gil, ed. (2013). *LMF: Lexical Markup Framework*. Hoboken, NJ: Wiley.

Govindaraju, Venu and Srirangaraj Setlur, eds. (2009). *Guide to OCR for Indic Scripts: Document Recognition and Retrieval*. London: Springer.

Grimes, Joseph E. (1970). "Computing in Lexicography." In: *The Linguistic Reporter* 12.5–6, pp. 1–5. URL: http://citeseerx.ist.psu.edu/viewdoc/download;?doi=10.1.1.620.9976&rep=rep1&type=pdf.

ISO TC37 (2008). *Language resource management — Lexical markup framework (LMF)*. Technical Report ISO 24613:2008.

Karagol-Ayan, Burcu, David Doermann, and Bonnie Dorr (2003). "Acquisition of Bilingual MT Lexicons from OCRed Dictionaries." In: *Machine Translation Summit IX*.

Ma, Huanfeng, Burcu Karagol-Ayan, David Doermann, Doug Oard, and Jianqiang Wang (2003). "Parsing and Tagging of Bilingual Dictionaries." In: *Traitement Automatique Des Langues* 44, pp. 125–150.

Morse, Nancy L., Jr. Jay K. Salser, and Neva de Salser (1999). *Diccionario Ilustrado Bilingüe: cubeo–español,español–cubeo*. Bogotá: Editorial Alberto Lleras Camargo. URL: https://www.sil.org/resources/publications/entry/19008.

Morse, Nancy L. and Michael B. Maxwell (1999). *Cubeo Grammar*. Studies in the Languages of Colombia 5. Dallas: Summer Institute of Linguistics.

Slocum, Marianna C. and Florence L. Gerdel (1976). *Diccionario tzeltal de Bachajón: castellano – tzeltal, tzeltal – castellano*. Serie de vocabularios y diccionarios indígenas "Mariano Silva y Aceves" 13. México, D.F.: Instituto Lingüístico de Verano.

Smith, Ray, Daria Antonova, and Dar-Shyang Lee (2009). "Adapting the Tesseract open source OCR engine for multilingual OCR." In: *MOCR '09: Proceedings of the International Workshop on Multilingual OCR*. New York: ACM. URL: http://doi.acm.org/10/1145/1577802.1577804.

TEI Consortium (2016). *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Technical Report. Charlottesville, Virginia. URL: http://www.tei-c.org/Guidelines/P5/.

Walton, James W., Janice P. Walton, and Clementina Pakky de Buenaventura (1997). *Diccionario Bilingüe muinane–español, español–muinane*. Bogotá: Editorial Alberto Lleras Camargo.

Zajic, David M., Michael Bloodgood, Benjamin Strauss, and Elena Zotkina (2015). *Faster, More Thorough Error Detection in Electronic Dictionaries*. Technical Report. University of Maryland: Center for Advanced Study of Language.

Zajic, David M., David S. Doermann, Michael Bloodgood, Paul Rodrigues, Peng Ye, Dustin Foley, and Elena Zotkina (2012). *A Hybrid System for Error Detection in Electronic Dictionaries*. Technical Report. University of Maryland: Center for Advanced Study of Language.

Zajic, David M., David S. Doermann, Paul Rodrigues, Peng Ye, and Elena Zotkina (2013). *Faster, More Accurate Repair of Electronic Dictionaries*. Technical Report. University of Maryland: Center for Advanced Study of Language.

Zajic, David M., Michael Maxwell, David S. Doermann, Paul Rodrigues, and Michael Bloodgood (2011). "Correcting Errors in Digital Lexicographic Resources Using a Dictionary Manipulation Language." In: *Proceedings of Electronic Lexicography in the 21st Century (eLex)*. Vol. abs/1410.7787. URL: http://arxiv.org/abs/1410.7787.