Waldayu and Waldayu Mobile: Modern digital dictionary interfaces for endangered languages

Patrick Littell
Carnegie Mellon University
Language Technologies Institute
5000 Forbes Ave.
Pittsburgh PA 15213
plittell@cs.cmu.edu

Aidan Pine
University of British Columbia
Department of Linguistics
2613 West Mall
Vancouver, BC V6T 1Z4
apine@alumni.ubc.ca

Henry Davis
University of British Columbia
Department of Linguistics
2613 West Mall
Vancouver, BC V6T 1Z4
henry.davis@ubc.ca

Abstract

We introduce Waldayu and Waldayu Mobile, web and mobile front-ends for endangered language dictionaries. The Waldayu products are designed with the needs of novice users in mind – both novices in the language and technological novices – and work in tandem with existing lexicographic databases. We discuss some of the unique problems that endangered-language dictionary software products face, and detail the choices we made in addressing them.

1 Introduction

Lexicographers have noted that with the increase in access to digital technology, "lexicography is clearly at a turning point in its history" (Granger and Paquot, 2012). While the changes that technology presents to lexicography are relevant to non-endangered languages as well, there exists a unique set of challenges in developing lexicographic materials for endangered languages in particular. We identify and address two of these fundamental and perennial difficulties.

1. At least in the North American context (and likely elsewhere), there are relatively few potential users who are both fluent in the language and trained in a systematic orthography. Many users are students who have not yet achieved fluency in the language they are searching.

2. Lexicographic efforts have, in many communities, taken place over generations by various scholars, using a variety of formats, orthographies, and assumptions, leading to data sets that are often very heterogeneous.

To address these issues, we have developed *Waldayu* and *Waldayu Mobile*. Waldayu is an orthographically-aware dictionary front-end with built-in approximate search, and a plugin architecture to allow it to operate with a variety of dictionary formats, from the output of advanced back-ends like TLex (Joffe and de Schryver, 2004), to semi-structured HTML like online word lists, to simple word/definition spreadsheets. Waldayu Mobile is a mobile implementation of Waldayu which is compatible with both Android and iOS devices.

Waldayu and Waldayu Mobile were originally developed to provide online and mobile interfaces to a forthcoming Gitksan (Tsimshianic) e-dictionary, but are intended to be language-neutral and have since been expanded to St'at'imcets (Salish), Nuuchah-nulth (Wakashan), Sliammon (Salish),

Squamish (Salish), Thangmi (Sino-Tibetan), and Cayuga (Iroquoian).

In Section 3, we discuss some of the user experience principles we have adopted, and in Section 4 discuss our challenges and solutions in adapting these to mobile devices. In Section 5, we discuss our implementation of approximate search.

2 A reusable front-end for novice users

Waldayu is primarily intended as a front-end solution to the perennial "Dictionary problem": that dictionaries are fundamentally a language-learning tool but require a certain level – sometimes an advanced level – of language knowledge to use in the first place. This is particularly apparent in our field context, the North American Pacific Northwest, where the sheer phonological and morphological complexity of the languages makes traditional print dictionary use particularly difficult (Maxwell and Poser, 2004).

We can observe this when, to give a real-life example, a user is trying to look up the word for "coyote" (sn̓ḵ̓yep) in Thompson's monumental print dictionary of Nłeʔkepmxcín (Thompson, 1996). To successfully find this word, the user must know that it is alphabetized under *ḵ̓* rather than *s* or *n* (which are prefixes/proclitics) and that *ḵ̓* and *y̓* are distinct from *k* and *y* for the purposes of collation. With thousands of nouns starting with *s* and about ten phonemes that might be confused with *k* (both pan-Salishan traits), most words in the dictionary cannot easily be found by novice users.

This is not a specific criticism of Thompson's lexicographic choices; complex languages require the lexicographer to make difficult decisions about orthography, morphology, morphophonology, and collation, and any decision they make about these will pose difficulty for some segment of novice users.

However, modern technology – particularly approximate search – allows us a way to meet the user halfway, by letting the system itself be aware of complications (prefixes/proclitics, easily-confused sounds, orthographic difference, etc.) that novices have yet to master.

Waldayu is not, however, intended to be a *replacement* for a mature, collaborative lexicographic software solution such as TLex, Kamusi (Benjamin and Radetzky, 2014), or the Online Linguistic Database (Dunham, 2014). Most lexicographic teams we have encountered already have preferred back-ends, file formats, and workflows; solving the "dictionary problem" for users should not require teams to abandon solutions into which they have already invested time, effort, and resources. Rather, Waldayu and Waldayu Mobile are intended to serve as a lightweight, uncluttered online front-end for novice users, while expert users can make use of more advanced functionality offered by a mature lexicographic database.

3 User experience and interaction

There are five user experience principles that we attempted to make consistent throughout both products, so as to remove barriers for novice users who may not be familiar with online dictionaries, or online interfaces in general.

3.1 Consistent control behaviours

Each control (button, search box, link, etc.) has only a single function, and no controls change their behaviour depending on the settings of other controls. For example, there is no Gitksan-English/English-Gitksan toggle that changes the behaviour of the search box; this convention, although ubiquitous in online bilingual dictionaries, is a frequent source of user error even for experienced users.

Rather, Waldayu utilizes a double search bar, in which user input in the left search box searches the primary language (e.g., Gitksan) while the left search box searches in the secondary language (e.g., English).[1] This parallels the two most frequent user tasks, using English as a query language to find an entry in

[1] This is superficially similar in appearance to the two-box interface used by Google Translate, but both boxes accept user input, and the user cannot swap the boxes or otherwise change what languages they represent.

Figure 1: Screenshot of Waldayu as the Gitksan/English Online Dictionary

Figure 2: Reduced Waldayu control layout embedded at the top of each page

a target language dictionary, and using a target language as a query language to find an English definition.

3.2 Immediate responses

In addition to the above, we try to avoid requiring other "two step" user inputs (e.g., typing a query and then hitting "search"). Instead, we attempt to provide user inputs with immediate feedback à la Google Instant search or other "AJAX"-style client and server interfaces (although as noted in Section 3.5, Waldayu does not in fact have a client-server architecture).

3.3 Consistent visual metaphors

In both controls and presentation, consistent visual metaphors are maintained. We borrow the fundamental metaphor from the FirstVoices online word lists (First Peoples' Cultural Council, 2009) – the online interface most familiar to our users – that the primary language (e.g., Gitksan) is always in the left column, and the secondary language (e.g., English) is always in the right column. This metaphor is maintained in the search interface (as seen in Fig. 1, the left search box searches the primary language, the right search box searches the secondary language), in browsing interfaces like the random word page (Fig. 3), and in entry pages (with primary language keywords and examples on the left, definitions and commentary on the right).

This horizontal visual metaphor is augmented by a colour scheme that presents the target language as green-blue (#066) and English as dark gray (#333); like other metaphors, these colours are preserved throughout the site.

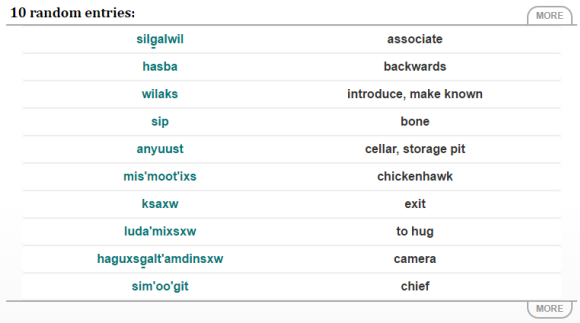

Figure 3: Position and colour metaphors maintained in a "browsing" page

3.4 Continued presence of controls

The major controls and links are present on every page, and work the same way on each page. The interface of the start page (a horizontal bar with links to free browsing, random browsing, etc., the page title, and the dual search bar) is embedded (with a reduced layout, as seen in Fig. 2) and fully functional at the top of every page.

This is intended to make it essentially impossible to "get lost" within the site; there is no question of how to return to the site's core functionality because every page has that functionality.

3.5 Connection-independence

Many of our user communities are located in remote regions of Canada, and some users do not have home internet connections or reliable mobile data access. While the initial connection to the Waldayu site, or the initial download of Waldayu Mobile, requires an internet connection, subsequent uses should not.[2]

While Waldayu appears to be a modern AJAX client-and-server web application, in actuality the Waldayu engine compiles a single HTML file, containing the dictionary itself in JSON format and JavaScript code that emulates a multi-page website. This way, the page can be downloaded and used offline on any platform; it has no installation steps, external files, or any prerequisites (save, of course, for a reasonably recent web browser).

4 The mobile user experience

Exporting Waldayu's user experience to mobile devices could not be done wholesale. While most laptop screens have a minimum of ˜700px in width, mobile devices are typically in the ˜300px range, with the result that maintaining the principles in Section 3 often required different interface decisions.

[2]Or, more precisely, as much functionality as possible should be retained even if the internet connection is lost. Some multimedia capability is lost when Waldayu loses internet access, but the basic search functionality remains.

4.1 Side menus

While a navigation bar that itemizes each individual page associated with the site was an appropriate organization for Waldayu, the navigation bar became too cluttered for mobile devices, especially considering Waldayu Mobile has additional pages such as a "Flashcards" page. For this reason, navigation for Waldayu Mobile was translated into a side-menu that is accessible by tapping the three-bar icon (what is sometimes called "hamburger" in mobile interface jargon) in the upper left corner of the screen, or swiping to the right on a touch screen. This is seen below in Figure 4.

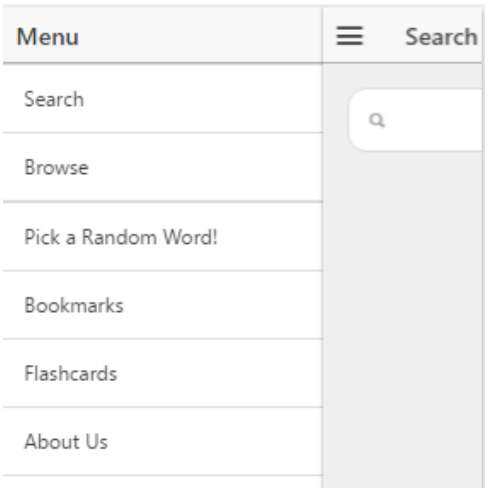

Figure 4: Side menus

4.2 A unified search bar

Similarly, maintaining Waldayu's horizontal metaphor for search bars in Waldayu Mobile would force search bars to be too small (i.e., less than 150px wide) for a positive mobile user experience. The option to force users to turn their devices and search with a 'landscape' (horizontal) orientation was also not preferred because it would result in forfeiting being able to visualize results as queries are searched. This would mean that users would have to search with a horizontal orientation, and view results either by scrolling or switching to a vertical orientation which would go against our UX principles in Section 3.

Initially, two solutions were created. The first kept both search bars and dynamically changed their widths depending on whether a

user tapped the left or right search bar. However, this still proved to limit the size of the search bar too severely.

The second solution was to use only a single search bar and a language selector; while this does not achieve the principle in Section 3.1, we hypothesized that increased attention to the visual metaphors in Section 3.3 might alleviate the difficulties this interface can cause. The language-selection radio buttons preserved the aforementioned horizontal metaphor (L1 on the left, L2 on the right) and dynamically changed the colour scheme of the search box to either the L1 or L2 colour.

While this solved the issues related to screen size, user testing revealed difficulties with the "two step" process, in which users first need to first consider what type of search they wished to perform and then select a button, which in turn changes the behaviour of the search bar. Likely, this is a result of the habituation to Google-style search bars where users can type any sort of query and expect accurate and relevant results.

Ultimately, this led us to develop a single search bar which searches the query term in both languages. The L1 and L2 results are presented together (still preserving the aforementioned visual metaphors), but with additional highlighting indicating, for each result, whether the match is on the L1 side or the L2 side. This is seen below in Figure 5.

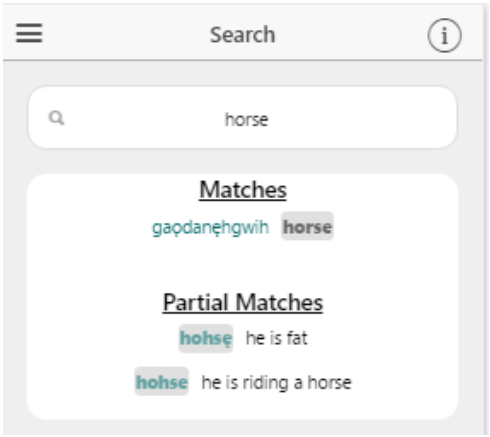

Figure 5: Highlighted search results in Cayuga (Ranjeet and Dyck, 2011) (left) and English (right)

5 Approximate Search

In this section, we consider the implementation of approximate search within Waldayu and Waldayu Mobile.

There are four primary reasons for why an approximate search is not simply a convenient feature for endangered language dictionaries, but a necessary one.

1. Some users, even if they can distinguish all the phonemes of the language (e.g., /*k*/ vs. /*q*/), do not always know the orthographic convention used to encode this difference.

2. Other users – particularly students – may know the orthographic convention but be unable to reliably discern certain phonological differences in their target language, resulting in systematic and predictable errors in spelling.

3. Many users will not have easy access to a keyboard which is able to type the required characters, whether for lack of a language-specific keyboard, difficulty with keyboard installation, or because they are using a mobile device.[3]

4. Many North American Indigenous languages have multiple orthographies, sometimes historical, but sometimes in current competition with each other. Different regions, generations, scholars, and schools have produced materials using different orthographic conventions, and users may have learned any of them, or may be attempting to enter data from material written in a different orthography.

Sometimes these issues are compounded, in that a user might type either "kl" or "tl" as an approximation of /ɬ/. Sometimes, though, the correct sequence of characters is perfectly valid using a standard English keyboard, but the user has difficulty hearing the distinction,

[3]FirstVoices (First Peoples' Cultural Council, 2009) recently developed an Android and iOS Keyboard app that allows users to type in over 100 endangered languages in Canada, the US and New Zealand.

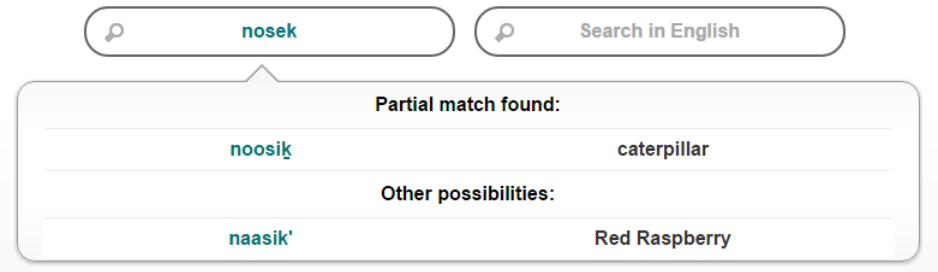

Figure 6: Gitksan-English Online Dictionary returning *noosik̲* and *naasik'* as possible matches to user query *nosek*.

as is the case with glottalized resonants (e.g. *'w* vs *w*) in many languages of the Pacific Northwest.

To address these issues, we implemented several approximate search algorithms. The most important criterion – aside from being fast enough to return results to users in a reasonable amount of time – was that this algorithm should either not require parameterization for a particular language, or should be able to be parameterized by a non-programmer (so that lexicographers can adapt the Waldayu software to a new language without hiring a programmer or having to contact the developers).[4] Ideally, a lexicographer should be able to adapt Waldayu using only familiar consumer software (like a spreadsheet program) rather than modifying the code or composing a structured data format directly.

5.1 Unweighted Levenshtein search

We began with a simple Levenshtein distance algorithm (Levenshtein, 1966), in which each word is compared to the user query and ranked according to the number of single-letter edits needed to change the query into the word. This comparison has the benefit that it can be expressed as a finite-state automaton (Schulz and Mihov, 2002) and, after construction of the automaton itself, run in linear time over the length of the query in characters, making search results nearly instantaneous regardless of the size of the dictionary.

Simple Levenshtein distance, however, is insufficiently discriminative to give useful results to users; for example, the Gitksan word for tree, *g̲an* (/Gan/), sounds similar to English "gun", but given the query *gun*, an unweighted Levenshtein algorithm has no reason to order the result *g̲an* above the result *din*, since these both differ from "gun" by two edits.

5.2 Weighted Levenshtein search

To address this, we then parameterized edit costs such that more similar and frequently-confused characters (*g̲* and *g*, *a* and *u*) accrue a lesser penalty, thereby ranking more-similar words higher, along the lines of Needleman and Wunsch (1970), Kondrak (2000), or Rytting et al. (2011).[5] Users could parameterize these penalties using a simple spreadsheet, in which they identified classes of similar sounds and chose [0.0-1.0] penalties within these classes.

While this approach proved adequate for our sample dictionaries, it did not scale well to larger dictionaries, leading to inappropriately long search times in practice, particularly on mobile devices.[6] Searching a five-letter word in a dictionary of about 9,000 words took up to 9 seconds on an iPhone 4.

[4]The third possibility is that the system learns appropriate parameters directly from the data. While this is promising for future work, we did not have, for any of our development languages, an appropriate corpus of user-transcribed data from which a system could learn orthographic correspondences.

[5]We also reduced the relative penalties for deletions and insertions at the beginnings and ends of words; this had the effect of allowing search for subword strings such as roots.

[6]As noted in Section 3.5, Waldayu ensures basic functionality even while offline. Search is therefore performed entirely on the client side, to ensure a consistent user experience online and offline.

5.3 Unweighted Levenshtein search on "comparison" forms

We therefore adopted a hybrid approach that leverages Waldayu's built-in orthographic conversion tools, in which we run an unweighted Levenshtein comparison between "orthographies" that purposely fail to represent easily-confused distinctions.

As background, Waldayu is designed to be orthography-independent, since it is designed to be able to take in heterogeneous resources that are not necessarily in the same orthography or the orthography that the user expects. Non-programmer lexicographers can specify orthographic correspondences using a simple table like that in Table 1, which Waldayu can read and compile into a finite state transducer.[7]

kw	k^w
k̲	q
k̲w	q^w
x̲	χ
x̲w	χ^w

Table 1: Sample orthographic correspondence table

In addition to specifying genuine orthographic transformations, the user can also specify transformations into one or more "comparison" orthographies, in which easily-confused (or difficult to enter on a keyboard) distinctions are not represented. Comparing words by collapsing similar sounds into a small number of equivalence classes is a technique of long standing in lexicography (Boas and Hunt, 1902)[8], information retrieval (Russell, 1918) and historical linguistics (Dogolpolsky, 1964).

In Waldayu, the user can specify a Soundex-like (Russell, 1918) transformation of entries and queries; by default Waldayu uses a transformation intended for North American Pacific Northwest languages ("PugetSoundex"), but it can straightforwardly be adapted to other languages. Table 2 illustrates a sample transformation.

k	KY
kw	KW
k̲	K
k̲w	KW
x	HY
xw	HW
x̲	H
x̲w	HW

Table 2: Sample correspondence table for approximate phonological comparison

For example, in the Gitksan development dictionary, the entry *noosik̲* ("caterpillar") undergoes a transformation into *NWSYK*, losing the distinctions between vowel length and height, the *u*/*w*/rounding and *i*/*y*/palatization distinctions, and the velar/uvular distinction. Meanwhile, a user input like *nosek* (Fig. 6) would undergo a similar transformation and likewise result in *NWSYK*. The results of these transformations are then compared using an unweighted Levenshtein distance; since the edit penalties are not continuous, we can implement this as a Levenshtein automaton using the `liblevenshtein` library (Edwards, 2014). As both orthographic transduction and Levenshtein automata operate in linear time, the resulting system returns results nearly instantaneously (dropping from the 9 seconds reported in Section 5.2 to less than 10 milliseconds).[9]

In practice, all entries in the development dictionary go through two transformations and comparisons, one to a very reduced form

[7] Programmer-lexicographers, on the other hand, can write orthographic transformation plugins if they require transformations more sophisticated than can be expressed in a tabular format.

[8] Presuming that he likely made mistakes in differentiating the similar sounds of Kwak'wala language, and that users of the dictionary would face similar confusions, Boas collated the glossary of the *Kwakiutl Texts* according to equivalence classes, so that, for example, all lateral fricatives and affricates were collated together.

[9] Qualitative evaluation of these approximate search algorithms (e.g, how often does a user query result in their intended word?) will require a larger collection of user-generated text than we currently have, and thus remains to be done.

like *NWSYK* and another to a more faithful (although still quite broad) phonological representation. The search algorithm ranks results according to a weighted average of these comparisons. This dual representation allows us to rank entries by both coarse and fine distinctions without using a continuous penalty function.

6 Conclusion and Future Research

Waldayu and Waldayu Mobile[10] are under continued development, although they are functional as-is and can be (and are being) adapted to additional languages.

Of the many features that remain to be implemented, several touch on unsolved (at least for this domain) research problems:

1. Incorporating algorithmic methods for determining the language of a given query term, along the lines of Cavnar and Trenkle (1994), could enhance the results of the "unified" search bar (Section 4.2, in order to dynamically prioritize L1 and L2 results according to which language the system decides the query is targeting. However, this is not a trivial task (Beesley, 1988), and the difficulties in Section 5 also pose difficulties for language identification, since many user queries will be attempts at L1 words but influenced by L2 phonology and orthography.

2. Combining phonological/orthographical approximate search with morphologically-aware subword search has proven problematic, when faced when long words composed of many relatively short morphemes. Take, for instance, the word for "telephone" in Gitksan:

   haluu'algyagamt'uuts'xw
   ha-luu-algyax̲-m-t'uuts'xw
   INSTR-in-speech-ATTR-wire
   "telephone" (Hindle and Rigsby, 1973)

   The results returned from combined phonological/orthographical/morphological search, although "relevant" in the sense that the entries contain sequences of morphemes that are phonologically/orthographically close to the query, can seem very counter-intuitive from a user point-of-view. It remains to be seen what level of morphological analysis and what notion of distance produces results that are intuitively "correct" from a user point of view.

3. We need to move our assumptions about the efficacy of our UX and algorithms beyond anecdote. For search algorithm efficiency, we would like to develop both statistical methods and analytics for determining how often users are given correct or relevant search results. For UX, we would like to conduct controlled experiments with users who possess varying levels of linguistic knowledge and target language competency.

4. Finally, the conflicting demands of web and mobile interfaces has led to a degree of codebase fragmentation. This fragmentation was in part due to Waldayu's origin as a web-based application – where there are fewer constraints on the interface – and subsequent adaptation to more-constrained mobile applications. The next version of Waldayu, now well into development, seeks to unify the interfaces and codebase as much as possible by taking a "mobile first" approach and implementing all three versions (web, Android, and iOS) in AngularJS, using the Ionic framework[11] for Waldayu Mobile. This change reworks Waldayu into a responsive Single Page Application (SPA), as seen in Figure 7, and brings features of Waldayu Mobile (like automatic flashcard generation) back into the web-based interface.

These problems suggest interesting future

[10] www.waldayu.org

[11] ionicframework.com

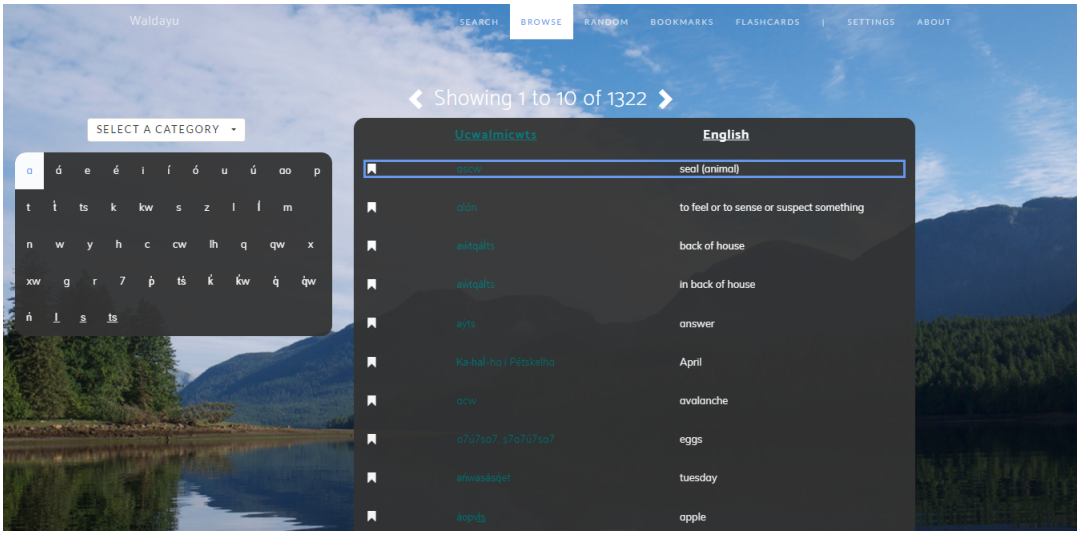

Figure 7: Browse state of Waldayu using AngularJS

research directions at the intersection of user experience, linguistics, and computation, and furthermore suggest that user-facing, novice-friendly interfaces may be valuable in generating novel data sets and research questions for endangered language research.

Acknowledgments

The development of Waldayu and Waldayu Mobile was supported by the SSHRC Insight Grant 435-2016-1694, 'Enhancing Lexical Resources for BC First Nations Languages'. Finally, we would also like to acknowledge our presence on unceded Coast Salish territory, where the majority of this work was inspired, created, and written about.

References

Kenneth Beesley. 1988. Language identifier: A computer program for automatic natural-language identification of on-line text.

Martin Benjamin and Paula Radetzky. 2014. Small languages, big data: Multilingual computational tools and techniques for the lexicography of endangered languages. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 15–23.

Franz Boas and George Hunt. 1902. Kwakiutl texts. *Memoirs of the American Museum of Natural History*, 5.

William B. Cavnar and John M. Trenkle. 1994. N-gram based text categorization. In *In Proc. of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175.

Aron B. Dogolpolsky. 1964. Gipoteza drevnejego rodstva jazykovych semej severnoj evrazii s verojatnostej toky zrenija. *Voprosy Jazykoznanija*, 2:53–63.

Joel Dunham. 2014. *The Online Linguistic Database: Software for Linguistic Fieldwork*. Ph.D. thesis, University of British Columbia.

Dylon Edwards. 2014. liblevenshtein: A library for generating finite state transducers based on Levenshtein automata. Retrieved from https://github.com/universal-automata/liblevenshtein.

First Peoples' Cultural Council. 2009. FirstVoices. Retrieved from http://www.firstvoices.com/.

Sylviane Granger and Magali Paquot. 2012. *Electronic lexicography*. Oxford University Press.

Lonnie Hindle and Bruce Joseph Rigsby. 1973. *A short practical dictionary of the Gitksan language*, volume 7. Department of Sociology/Anthropology, University of Idaho.

David Joffe and Gilles-Maurice de Schryver. 2004. Tshwanelex: A state-of-the-art dictionary compilation program. pages 99–104.

Grzegorz Kondrak. 2000. A new algorithm for the alignment of phonetic sequences. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 288–295. Association for Computational Linguistics.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710.

Michael Maxwell and William Poser. 2004. Morphological interfaces to dictionaries. In *Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries*, ElectricDict '04, pages 65–68, Stroudsburg, PA, USA. Association for Computational Linguistics.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–53.

Kumar Ranjeet and Carrie Dyck. 2011. Cayuga digital dictionary. Retrieved from https://cayugalanguage.ca.

Robert C. Russell. 1918. *U.S. Patent 1,261,167*. U.S. Patent and Trademark Office.

C. Anton Rytting, David M. Zajic, Paul Rodrigues, Sarah C. Wayland, Christian Hettick, Tim Buckwalter, and Charles C. Blake. 2011. Spelling correction for dialectal arabic dictionary lookup. 10(1):3:1–3:15, March.

Klaus U. Schulz and Stoyan Mihov. 2002. Fast string correction with Levenshtein-automata. *International Journal of Document Analysis and Recognition*, 5(1):67–85.

Laurence C. Thompson. 1996. *Thompson River Salish dictionary: Nłeʔkepmxcín*. University of Montana Occasional Papers in Linguistics.