Language Documentation & Conservation

Volume
18 (2024)

Pages
109 - 131

Received
29 Dec. 2023

Handle
https://hdl.handle.net/10125/74788

Online
http://nflrc.hawaii.edu/ldc

Citation
Jones, Caroline, Jesse Tran, Eleanor Jorgensen, Romi Hill, Patricia Ellis, Jane Simpson, Felicity Meakins, Ben Foley, Marcel Reverter-Rambaldi, Gari Tudor-Smith, Paul Williams, Clair Hill, Mark Richards. 2024. WordSpinner: Developing a tool to convert plain-text lexicon files into dictionary webpages. Language Documentation & Conservation 18: 109-131.

WordSpinner: Developing a tool to convert plain-text lexicon files into dictionary webpages

Caroline Jones
The MARCS Institute for Brain, Behaviour & Development
Western Sydney University
ARC Centre of Excellence for the Dynamics of Language

Jesse Tran
The MARCS Institute for Brain, Behaviour & Development
Western Sydney University
ARC Centre of Excellence for the Dynamics of Language

Eleanor Jorgensen
Australian National University
ARC Centre of Excellence for the Dynamics of Language

Romi Hill
Australian National University
ARC Centre of Excellence for the Dynamics of Language

Patricia Ellis
Minga Aboriginal Cultural Services

Jane Simpson
Australian National University
ARC Centre of Excellence for the Dynamics of Language

Felicity Meakins
University of Queensland
ARC Centre of Excellence for the Dynamics of Language

Ben Foley
University of Queensland
ARC Centre of Excellence for the Dynamics of Language

Marcel Reverter-Rambaldi
University of Queensland
Australian National University
ARC Centre of Excellence for the Dynamics of Language

Gari Tudor-Smith
University of Queensland
ARC Centre of Excellence for the Dynamics of Language

Paul Williams
University of Queensland
ARC Centre of Excellence for the Dynamics of Language

Clair Hill
The MARCS Institute for Brain, Behaviour & Development
Western Sydney University
The University of New South Wales
ARC Centre of Excellence for the Dynamics of Language

Mark Richards
The MARCS Institute for Brain, Behaviour & Development
Western Sydney University
ARC Centre of Excellence for the Dynamics of Language

E-ISSN 1934-5275

Language Documentation & Conservation

Dictionaries for under-resourced languages are, like other dictionaries, traditionally in hard-copy form. Electronic online formats can improve access to dictionaries for under-resourced languages. To make it easier to create online dictionaries with linked sound and image files, starting from a Toolbox file or similar backslash-formatted dictionary text file, a new online tool, WordSpinner, was created. It was used and iteratively improved over several years in order to convert backslash lexicon text files into formatted dictionary output pages, with five Indigenous languages of Australia. In various stages of dictionary creation, checking, and formatting, the online tool was revised to meet user needs. The resulting version of WordSpinner is another free option now available to users for creating online dictionaries.

E-ISSN 1934-5275

1. Introduction[1]

For community members, linguists, and teachers engaged in language revitalisation, a dictionary can be a key resource. Dictionaries represent a 'reference' resource. Ideally, dictionaries enable learners and teachers to look up the spelling and pronunciation of words, as well as their meanings and uses. Digital dictionaries can allow users to hear the word and examples of word usage. Dictionaries can also function as learning resources for the right kind of learner or context. In language revitalisation or reclamation contexts, dictionaries often support adults in the self-study of their languages, sometimes because they are the only resource available (for further discussion, see Himmelmann 2016; Richards et al. 2019). Some dictionaries also include a grammatical sketch or provide enough examples for an adult to learn and/or teach grammatical aspects of the language. Dictionaries can also support the creation of many different learning and teaching resources, in classroom contexts for example, and often offer the first standardisation of spelling and grammar. Finally, and significantly, a dictionary also brings status and pride to a language and its community. As Lucy Pedwell, a Ngarinyman community leader, said about Indigenous language dictionaries in Australia at the 2019 launch of the dictionary for her language, "These books allow us to share our stories and have our voices heard on a much wider platform. They are a symbol of the strength of our culture and languages" (Archibald-Binge 2019).

Although dictionaries are a key resource, there are real accessibility and usability issues around dictionaries in many revitalisation or reclamation contexts. In terms of accessibility, an existing published dictionary, even if there is one, may be prohibitively expensive. It may be difficult to obtain copies, for example, if it is out of print or if it is held only in libraries outside the local area. Even if a copy can be obtained, durability contributes further to accessibility issues. Print-based language resources quickly deteriorate in extreme weather conditions (e.g., heat, humidity, dust) and with high usage, as when lovingly and frequently used by people who are teaching or learning their language. Hardbound books last longest, but many dictionaries are printed in cheaper paperback format. Many community members may just own a photocopy. Language community members also report the experience of lending their only copy to someone and finding it difficult to get it back due to the rarity of copies. It is common then for community members to be dependent on teachers or linguists to provide new copies, with access plagued by the intermittent nature of linguist visits and high turnover of teachers. For many languages in revitalisation or reclamation contexts unfortunately, there simply is no available published dictionary. For some languages, dictionaries may be scarce. There may just be a few historical manuscripts in a library, or there may be a manuscript typed up in recent years in circulation as a working draft lexicon.

In addition to accessibility concerns, usability issues are also rife. The dictionary may have been written some decades ago and contain obscure or dated academic terminology, which can be relatively opaque to people whether they are academically trained or not. Beyond these issues, previous research in Australian contexts points to usability issues that a modern and well-regarded dictionary can still present for many users. Such is the case when it comes to finding and interpreting information efficiently and being able to pronounce the spelling in an unfamiliar orthography (see Corris et al. 2004 for discussion of Warlpiri dictionary use).

In relation to usability, it is worth also noting that dictionaries in language revitalisation contexts have a more complex range of linguistic functions and cultural and social value than dictionaries in many other contexts (see Garrett 2018 on this point). They have language-learning functions, and they have implications for the success of language revitalisation (Garrett 2018).

The aim of this article is to describe the development of the new software WordSpinner. This software creates webpages for dictionaries for languages in revitalisation or reclamation contexts and targets some of the needs of speech communities in these contexts, with specific focus on Australian Aboriginal communities with whom they have been co-developed. This paper is structured as follows. In §2, background to electronic dictionaries in the Australian Aboriginal languages context is provided, then the motivation for WordSpinner is given in §3. In §4, the rationale for the input format and the architecture of the tool are described. §5 presents case studies (2017–2022) of how WordSpinner

[1] The authors acknowledge the following people for their participation as follows: Tula Wynyard (for feedback trialling WordSpinner for Dharug), Lucy Pedwell, Mikayla Friday-Shaw, Sharon Macmillan (for use of an early version for Ngarinyman), Lesley Woods, and Isabel O'Keeffe (for feedback on current use of WordSpinner for Ngiyampaa Wangaaypuwan). The research received human ethics approval from Western Sydney University (H11190, H14183). This research was supported by funding from the Australian Research Council Centre of Excellence for the Dynamics of Language – ARC CoEDL (CE140100041), an Australian Research Council Future Fellowship awarded to the first author (FT120100777), an Australian National University summer research program (2018–2019), and two University of Queensland Summer and Winter Research programs (2020–2021).

has supported dictionary creation by teams for five First Nations languages in Australia and how the use cases have prompted improvements in the software. §6 offers future directions, and §7 is a brief conclusion. Appendices A and B provide instructions for WordSpinner use.

2. Background to electronic dictionaries in the Australian-Pacific context

Projects for electronic dictionaries in the region began in 1966–1968 with the Nyungic project at the University of Hawaiʻi, which entered vocabularies of many Australian languages on punch cards (project led by Geoffrey O'Grady, funded by the National Science Foundation). For the transition of the Warlpiri dictionary through software systems from 1981 to the present, see Nash (2022). Since the late twentieth century, many such dictionary projects have produced plain-text files, where backslash codes are used to organise the contents of each lexical entry (e.g., \lx and \ps for the headword and part-of-speech fields, respectively). This approach has been popular among linguists and supported by software developed by SIL (Summer Institute of Linguistics), including Shoebox or Toolbox, Lexique Pro, and, more recently, FLEx and WeSay (for more details, see the Multi-Dictionary Formatter (MDF) guide by Coward & Grimes 1995). Some resulting lexicon files have been archived as well-structured, plain-text files (e.g., in Australia at ASEDA, AIATSIS, and PARADISEC[2]). Others have been converted into Word documents and printed out as working-draft dictionaries, to be used, edited, and improved, often over the decades of collaborative work that it can take to produce a dictionary.

The first online dictionary of an Australian language was launched in 1996 in Hyper Text Markup Language (HTML) format – *Kamilaroi/Gamilaraay Web Dictionary*[3], which is still available online. A notable and innovative project was the development of the KirrKirr platform (http://www-nlp.stanford.edu/kirrkirr/)[4] for the Warlpiri dictionary. This platform works with dictionary files in Extensible Markup Language (XML) format, which allows for active links to sound files, pictures, and visualisation of semantic relationships (Manning et al. 2001). In 1999, an online dictionary of Wagiman was created using a Microsoft Access database (Wilson et al. 2001). It is still available but not through an archive.[5] By 2008, Miromaa[6] was available as another dictionary utility (for Windows), which provides easy entry and upload to a database structure; its disadvantages include some limitations on form and type of output. There are also Lexique Pro dictionaries produced by SIL that are online; some date from the 1990s and 2000s.[7] Other online dictionaries of Australian languages, such as the influential Oykangand and Olkola dictionary (prepared by Philip Hamilton and the Kowanyama Aboriginal Community Council, ca. 1997), are either no longer available or only available through website archives such as Trove. This shows the importance of archiving online dictionaries and creating them in archivable formats.

Increased dissemination prospects provided by growing use of mobile phones have also been taken up. In the mid 2000s, Wunderkammer (McElvenny 2008; McElvenny & Wilson 2009) arose from a project to create a pronunciation guide for Kaurna, the language of the Adelaide Plains. Wunderkammer was a software program developed within the Project for Free Electronic Dictionaries[8] at the University of Sydney with the primary aim to make First Language dictionaries available with sound files to speakers and learners via mobile phone (see Hill 2012 for a review). The Wunderkammer approach responded to community requests for multimedia language materials so that it was possible to hear the words pronounced (Simpson 2003). The Wunderkammer system (McElvenny & Wilson 2009) involved a display program, which ran on the Java ME (Micro Edition) platform on non-smartphones, and an import program (wkimport), which converted a master backslash file format for storing the dictionary data in Wunderkammer. This approach was

[2] ASEDA = Aboriginal Studies Electronic Data Archive; AIATSIS = Australian Institute of Aboriginal and Torres Strait Islander Studies; PARADISEC = The Pacific and Regional Archive for Digital Sources in Endangered Cultures.

[3] Austin, Peter & David Nathan. 1996. *Kamilaroi/Gamilaraay web dictionary.* (https://www.dnathan.com/language/gamilaraay/dictionary/index.html) (Accessed 2023-12-15.)

[4] (Accessed 2023-12-15.)

[5] https://aphasialab.org/wagiman/ (Accessed 2023-12-15.)

[6] https://www.miromaa.org.au/miromaa-software (Accessed 2023-12-15.)

[7] https://ausil.org.au/dictionaries/ (Accessed 2023-12-15.)

[8] The link for Wunderkammer is now defunct, though reference to it is found in the PARADISEC blog at https://www.paradisec.org.au/blog/2009/01/wunderkammer/ (accessed 2023-12-15).

designed to capitalise on existing records and on the then-recent growth in mobile phone ownership in many parts of the world. In particular, the 2000s saw rapid uptake of (non-smartphone) mobile phones by First Nations people across Australia. For Whitesands, another focus language for the project, the context was Vanuatu, where the community's mobile phone tower was turned on in August 2008 (McElvenny 2010: 1506). The resources could be distributed within the community via Bluetooth. Wunderkammer was used for several languages, including Wagiman, Dalabon, and Umpithamu. A view of an example entry in Wunderkammer is shown in Figure 1 (reproduced from Figure 5 in McElvenny & Wilson 2009).

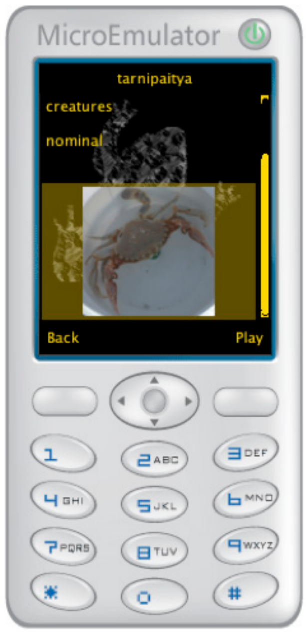


Figure 1. View of an entry in Wunderkammer

Due to hardware and software developments (advances in mobile phones and their operating systems), Wunderkammer is no longer functioning. There have been other mobile-phone dictionary projects in the Australian context, such as Ma! Iwaidja, developed for the Arnhem Land language Iwaidja and launched in 2012 with a strong focus on crowdsourcing to support app development (Birch 2013). Like Wunderkammer, it is no longer functioning. The longevity risk presented by rapid obsolescence of hardware and software highlights the importance of using open-source software (so the software can be updated as long as possible), of being able to extract or convert data from a dictionary project without loss of information or of links between data, and of having reliable long-term hosts for online dictionaries.

Instead of mobile phone apps, software exists to publish electronic dictionaries on the internet. One option available is Lexique Pro,[9] freeware for viewing (and editing) electronic dictionaries. The Lexique Pro editor, however, has to be installed on the user's computer, requires a Windows computer (or on a Mac under emulation or with Wine), and demands time to learn to use. Lexique Pro creates a viewer on a local computer, limiting it only to users with access to the computer. A webpage export function exists but does not include audio or images.

A newer option for creating mobile phone dictionaries is Dictionary App Builder[10] from SIL. This creates a dictionary app for Android and iOS smartphones and tablets from existing lexicon files in a range of SIL tools (e.g., FLEx, We-

[9] http://lexiquepro.com (Accessed 2023-12-15.)

[10] https://software.sil.org/dictionaryappbuilder/ (Accessed 2023-12-15.)

Say, and Lexique Pro). In doing so, it maintains a partition between the dictionary content and the form, which helps to overcome issues with app longevity. The Kullilli Ngulkana 'learning Kullilli' dictionary app for the south-western Queensland language Kullilli used this software.[11]

Another current dictionary-building system, Mother Tongues Dictionaries,[12] provides a way for people to create webpages (as well as mobile apps) with their own dictionary data. Mother Tongues has sophisticated approximate ('fuzzy') search, can be deployed on web/Android/iOS platforms, and is well documented as an open-source technology (Littell et al. 2017). To our knowledge, this software has not been used in the Australian context to date. Other current dictionary-building systems designed for small endangered languages include the Word Oriented Linguistic Framework,[13] which can import MDF-marked-up files and produce output as webpages or documents (Ransdell-Green 2021). It is not yet open source and not currently used in the Australian context.

3. Developing WordSpinner

This section describes the development of a free online tool system,[14] WordSpinner, to create webpages for dictionaries for languages in revitalisation or reclamation contexts. We provide an overview of key issues in the form of answers to frequently asked questions about the tool.

3.1 What is WordSpinner?

This software is a free online tool system.[15] It is intended for temporary use by teams wishing to develop dictionary webpages and host them elsewhere. It is not a publishing platform or an archive or repository. Instructions for users detail how to upload an input file and download zip files of webpages (see Appendices A and B). These webpages can then be post-processed by a web developer (for font type, size, colour, etc. and optimised for mobile phone use) and hosted with associated multimedia files (audio, images) through a URL.

3.2 What kind of input files does WordSpinner accept?

WordSpinner imports an MDF backslash file (e.g., a Toolbox file), plus image and sound files, and produces a set of formatted webpages. It is also possible to start with a Standard Format Marker (SFM) format exported from FLEx,[16] but the system was designed to create webpages from Toolbox files because many language lexicon files were created in this format.

3.3 Are there IP issues with files uploaded onto WordSpinner?

The intellectual and cultural property rights in any dictionary are unaffected by the use of the WordSpinner utility.

[11] https://kullillingulkana.wordpress.com/ (Accessed 2023-12-15.)

[12] https://mothertongues.org/ (Accessed 2023-12-15.)

[13] https://knowledgebase.arts.ubc.ca/wolf/ (Accessed 2023-12-15.)

[14] https://dictionaryresearch.org is the system home page; see Appendix A for login details.

[15] https://dictionaryresearch.org (Accessed 2023-12-23.)

[16] One issue that arises in using an SFM format as input to WordSpinner is that the export does not standardly include \so or \rf fields. These fields are typically important fields when working in a 'language reclamation/revival' type context, to demonstrate historical authenticity and explain the sources of words or examples. As a workaround, it is possible to create two custom fields under the main entry (e.g., \zo and \z1 for \so and \rf, respectively). We thank Tula Wynyard for pointing this out to us.

Files uploaded to WordSpinner are not shared with any individuals or organisations or used for any purpose beyond the creation of webpages within each user's private log-in area. WordSpinner is not intended as a place to store files long-term. The user is encouraged to use the button to clear all dictionary files from the server once the webpages are created and the downloaded zip files are saved. The server is also cleared of all files once every three months as a housekeeping and data security measure.

3.4 Where can the webpages be put?

The webpages created by WordSpinner can be hosted wherever the relevant community has decided to host them, with appropriate access permissions decided by the language community. Importantly, however, the zipped webpages are not left on WordSpinner long-term; it is not a publishing platform. The webpages can be hosted online or offline (e.g., on an unnetworked computer). Putting the webpages online means they are accessible to the increasing numbers of people who have an internet connection and access to a device with a browser (e.g., a smartphone, tablet, or computer).

3.5 Do we need technical skills to use WordSpinner?

Yes. One member of the dictionary team needs to be able to edit MDF dictionary files accurately and consistently. Once WordSpinner has been used to create zip files of webpages, the team also needs access to someone who has technical expertise in HTML, CSS (Cascading Style Sheet), and JavaScript to post-process the WordSpinner output.[17] For instance, having these skills allowed the Dhurga project team to implement colour coding of parts of speech after generating the HTML using WordSpinner. Having a team member who was able to upload media files to a webserver also meant the Bilinarra, Gurindji, Mudburra, and Ngarinyman dictionary teams could format the WordSpinner output for publication as public dictionaries.

3.6 Does the dictionary need to be fully finalised before using WordSpinner?

No. While you can certainly use WordSpinner to create a web version of an existing published paper dictionary, WordSpinner is designed for use by teams that are still in the process of compiling their dictionary. WordSpinner allows for webpages to be easily updated (by recreating the webpages or, that is, re-running the software with an edited text file).

This makes WordSpinner a tool that is suitable for collaborative use during the process of dictionary checking and editing (e.g., before a dictionary is finalised and printed). Once a dictionary is fully checked, edited, and ready for wider delivery beyond the dictionary creation team, WordSpinner can also be used again to make webpages from the master dictionary text file. This increases the appeal of the dictionary, boosts dictionary access, and allows the resource to remain up to date.

WordSpinner is designed as a utility to facilitate viewing and sharing dictionary content online, rather than restricting the resources to the computer on which they were made. This opens up collaborative and decentralised teamwork possibilities. (This type of usage is described in the first case study below in §5.) Team members in different parts of the world are able to view the current version of the dictionary in a readable format (i.e., without backslashes, with entries sorted by semantic domain or alphabetically) and discuss changes via Zoom, email, or phone, with one team member using the WordSpinner tool to remake the pages quickly for subsequent rounds of checking. While it is true that WordSpinner requires internet access for this type of online teamwork, the webpages from WordSpinner can be posted on an intranet or viewed on an unnetworked computer, using a browser. The webpage outputs of WordSpinner can also be printed out and shared by mail if necessary.

[17] WordSpinner could feasibly be modified to produce output that reduces the need for such post-processing.

3.7 What kinds of dictionary files can be handled? Are multiple senses and English-to-Language lookups possible? Can we include sound or image files?

WordSpinner is built to handle lexical entries that contain multiple senses. WordSpinner can also use the MDF reversal field (\re) to display entries in English-to-Language format, as well as in Language-to-English. This is helpful for learners who want to find the language word starting with an English meaning, such as beginner learners or learners/teachers in revitalisation/reclamation contexts where English is the dominant language of literacy and education. WordSpinner allows for the addition of sound clips to assist with pronunciation and image files to support understanding and to break up the text for readers with lower levels of literacy (e.g., young children).

§3 has set out key information: what WordSpinner is, what it is not, and ways it can be used. To assist potential users of the WordSpinner tool, the next section expands on §3.5 by explaining MDF: the kind of text file that needs to be used as the input to create dictionary webpages using WordSpinner.

4. A tool to create formatted webpages from plain-text lexicon files in MDF format

4.1 Why MDF format?

MDF format is a simple way of formatting a plain-text file for dictionaries so that for each lexical entry (i.e., each headword or main entry), each associated piece of information is categorised into one of a relatively small set of information types or kinds. An example of MDF format for one lexical entry in the Ngarinyman dictionary is shown in Figure 2. Key backslash codes are indicated with arrows. Each backslash code represents a category or type of information; what follows the backslash code (after a single space) is the specific information that corresponds to that category for each lexical entry. For example, in Figure 2, the headword is *gumulyurrany*, the part of speech is *n* (for noun), the English translation is 'budgerigar', and the semantic domain is "Birds."

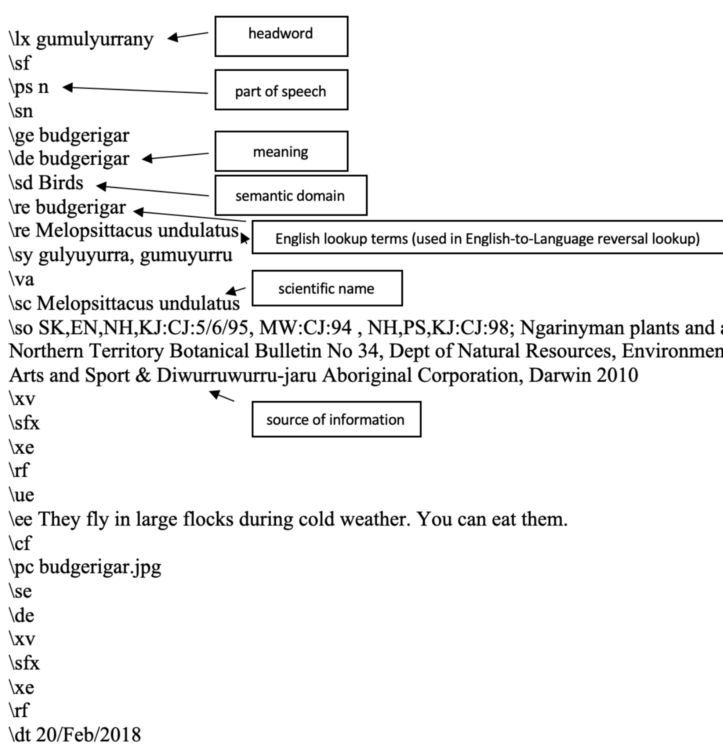


Figure 2. The lexical entry for *gumulyurrany* in MDF format

In addition to the key backslash codes indicated with labels in Figure 2, there are codes for sound file (\sf), gloss (\ge), usage (\ue), encyclopaedic information (\ee), cross-referencing (\cf), image file (\pc), any synonym (\sy) or variant pronunciation (\va), the options to number and include more than one sense of a word (\sn), one or more subentries (\se) (for derived forms), as well as an example sentence (\xv), its sound file (\sfx), a translation into the user's reference language, say, English (\xe), the source or reference for that example sentence (\rf), and date last modified (\dt). Having more than one sense (\sn) or more than one subentry (\se) means that a sequence of backslash codes repeats. (MDF involves a hierarchical structure like this in more complex lexical entries.)

MDF format has several advantages as an input file type in a webpage creation tool for dictionaries. MDF format requires well-structured data and is a suitable archive format for master dictionary text files. The individual components of a lexical entry, such as headword, part of speech, definition, and example sentences, need to be assigned consistently to backslash-coded fields. In addition to the distribution of information to separate fields, MDF format encourages a consistent order for fields within a lexical entry. These two aspects of MDF make the input data more predictable so a computer program can assign a set of formatting styles (e.g., font type, font size, punctuation), making the information types (e.g., headword, definition) within a lexical entry easier to scan. For example, the lexical entry for *gumulyurrany*, as shown in Figure 2, has the visual appearance shown in Figure 3 when converted to a webpage by WordSpinner. Finally, MDF format is designed for use in plain-text files, which are expected to remain readable by computers for some years, and are suitable for archiving. In practice, however, MDF format is still vulnerable to inconsistencies and human error (e.g., in the inclusion or order of fields; see Maxwell 2004).

Figure 3. The lexical entry for *gumulyurrany* in the webpage output from WordSpinner, in Ngarinyman-to-English (left) and English-to-Ngarinyman (right)

Other advantages of the MDF format are that it can contain multimedia fields, which can be capitalised on to produce an online dictionary. MDF can contain fields for associated image and sound files for each lexical entry (e.g., \pc for image, \sf for sound file, \sfx for example-sentence sound file). These audiovisual additions can make dictionary webpages more accessible and enjoyable for learners of all ages and background levels, particularly learners who are new to the spelling system. In our experience, there is always a demand from teachers and learners for audio examples. Audiovisual links potentially make the resources usable in school programs, which may not have access to a speaker or where the speakers are now too elderly to be present in the classroom. The resulting resource also tends be viewed positively by Elders, who see it as a priority to interest children in learning and view computers as appealing and engaging for children. These features are also especially crucial in language revitalisation or reclamation contexts where hearing the pronunciation of words is a key aid to adult learning or self-teaching.

MDF format also includes a source field (the main one being \so). Although printed dictionaries for small languages often do not include details of the source of the information in each lexical entry, the inclusion of this information in a public version of the dictionary is particularly appealing for community members whose dictionary is built from archived manuscript sources. It allows the user to track the authenticity and evidence quality (see, e.g., the dictionaries of New South Wales languages Dhurga and Darkinyung, among others). It is also the case that in languages that are still spoken by elderly adults (or were into the twentieth century), certain sources are often closely associated with one or two senior individuals. Younger community members, in our experience, are interested to know this family and community heritage and to know how this may relate to dialectal variants. In an online format, presentation space is not as expensive as in a printed dictionary, so it becomes possible to display more of the MDF file content/capability, including extra images and source information.

4.2 The functionality and architecture of the tool

The main aim of the tool is to allow people to create formatted webpages in the case that there is already a plain-text dictionary file in backslash format and permission for dictionary content to be shared online. These users visit a URL, upload the dictionary file and associated image and sound files, and then download webpages onto their device.

Each webpage generated by WordSpinner corresponds to a semantic domain (e.g., people, animals, plants, action types), to provide some context for the reader rather than the experience of "jumping" topics and to make it potentially easier for the dictionary to be a resource for theme-based teaching or for learners to browse. Within each semantic domain, the entries are organised in alphabetical order and can be searched using the browser's search function. (A WordSpinner dictionary can consist of an alphabetical list of all words; in other words, it does not have to be presented in semantic domains. This can be achieved by using one entry in all semantic domains fields (e.g., \sd All).)

The tool was built over a six-year period between 2015 and 2021 and has had two major versions: Version 1 (2015–2018) and Version 2 (2018–2021). The development journey is detailed in §5 below. The following section provides more technical detail on the current system (Version 2).

4.3 Architecture of Version 2 (as of August 2021)

WordSpinner is currently available online as a website using client and server architecture (see Figure 4). People from around the world using a variety of devices such as desktop computers to mobile devices are able to utilise it to collaboratively generate dictionary webpages.

The server side of our system was implemented using the PHP (Hypertext Preprocessor) programming language.[18] A PHP module was written to translate static-text dictionary data into a webpage format. WordSpinner is hosted on an Apache server. The WordSpinner interface is created with HTML and CSS.

[18] PHP is a popular programming language, commonly used for developing webpages with dynamic content.

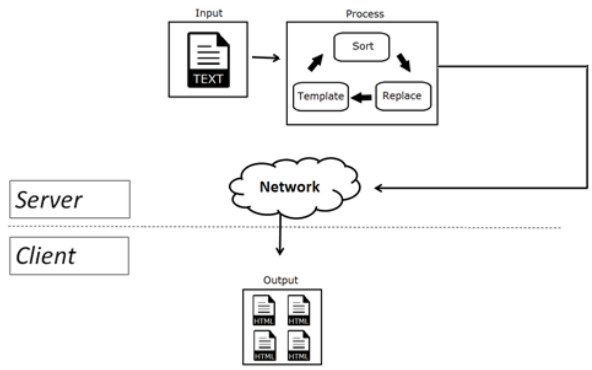


Figure 4. Framework of the system. A backslash text file is uploaded via a web form. The system processes the backslash data by sorting and formatting the fields into semantic-domain templates.

Three features are presented on the website's start page. First is a management feature that generates a hash key (a kind of unique ID) to identify the current session. A session refers to a unique state of the website in which the user can input, process, and view their own data. This unique ID, acting like a password, allows the user to access the same session on another device if the current session gets cleared or to give access to a third party such as a collaborator. The unique ID consists of arrangements of characters and numbers (e.g., 6163d17ea8005).

The advantages of this type of session management are that it bypasses the need for each user to create and manage their own account with a username and password, and it can be shared by a team. Its security is the equivalent of having a password. Many websites and applications use this technique to save the user from constantly having to log in again each time they use the system. In WordSpinner, the unique IDs are given to the user, whereas other information systems may save the unique IDs into the URL address or "HTTP cookie" files.

The second feature is a form that allows the user to upload a formatted text file (see the "Upload" button in Figure 5). This text file can contain multiple word entries with a set of predefined backslash fields that are repeated for every entry. The file must be in UTF-8 encoding with Windows CRLF (Carriage Return Line Feed) line endings.

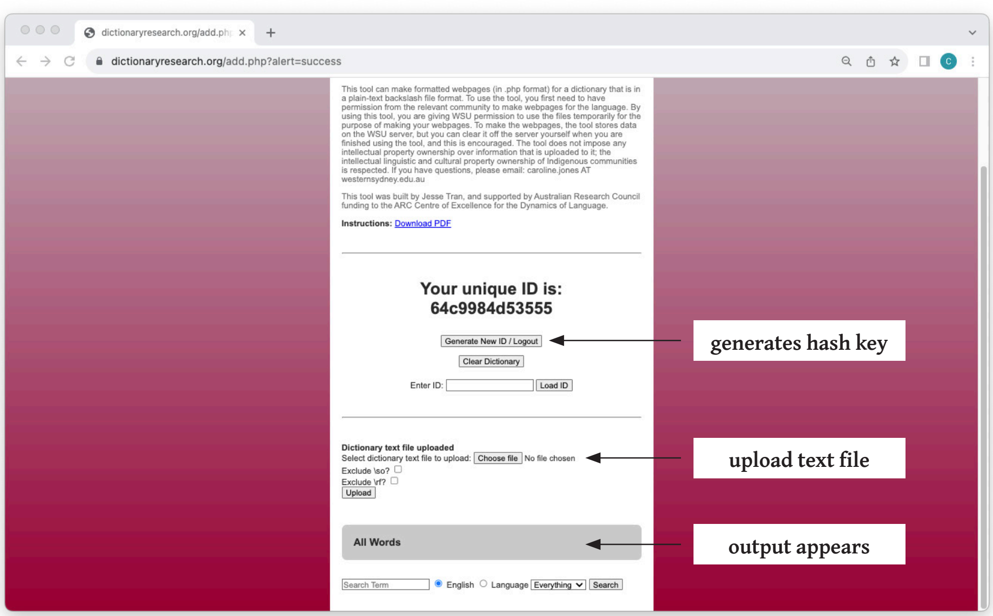


Figure 5. Current interface for WordSpinner

Once the user clicks the "Upload" button, the data is processed. First, the program stores the names of all the backslash fields using a variable, $allVars. For example, a variable holds an array (a data structure containing many elements) of all the backslash field names, such as \lx, \sf, and \ps. It does not store the values of these backslash fields. This enables the system to keep track of all the dictionary's backslash fields so that the user can create new backslash fields in the future and still have them processed without requiring rules for each field to be hard coded into the system beforehand.

Next, the entire text file is separated into a set of entries. If there are ten entries, the entire text file would be separated into ten parts. After the separation, each entry's fields are further separated, line by line. For example, if there are twenty backslash fields for the entry, the entry would be separated into twenty parts. The fields are filtered to have unnecessary spacing and characters removed and are stored as an item in the array variable, $text_lines. For example, the first element $text_line[0] will contain \lx plus its headword contents, the second element $text_line[1] will contain \sf, and so on. For each item from the $text_lines array, the backslash fields are extracted and are used as array variable names. For example, \lx dynamically becomes $lx, which then stores the actual value of the backslash fields as one of the array items. For example, in relation to the headword *gumulyurrany* 'budgerigar' in Ngarinyman, \lx gumulyurrany becomes $lx = array("gumulyurrany").

The reason for making the variables hold arrays instead of single sentences is to enable duplicate backslash fields of the same name to be saved as additional elements in the array and not be overwritten. For example, if there is another \lx backslash field for the same entry, instead of replacing the value of the first field, it would be added to the next item in the $lx array, hence being able to keep track of an unlimited number of values regardless of the number of times they repeat.[19] There is also the advantage of allowing words to be sorted alphabetically. With this in mind, if an entry happens to have several definitions, it would follow the same process so that the corresponding backslash field is processed in tune with each definition. If the backslash fields have no value, an empty item is created.

After processing the text, a collection of webpages named by their semantic domains are generated. Each of these webpages is built using a template styled with CSS and HTML. Based on the semantic domain (or other categorising) information in each entry's \sd backslash field, the PHP appends the entry's text values to the relevant category page. For the system to know the list of variables it had generated and processed, along with their orderings, it references back to the values for each backslash field it stored in $allVars from the beginning, loops through each element, and inserts them into the corresponding webpages.

The final output of WordSpinner is a set of webpages, which the user downloads from the online tool as a set of .zip files containing the output webpages. The user can then clear all the data (inputs and outputs) from the server. The sets of webpages are Language-to-English and English-to-Language, both of which are organised by semantic domain. Within each semantic domain, the listing of entries is alphabetical. (Note that the software does not require English to be the learner's reference language, but we use this description for the Ngarinyman example here.)

Once a dictionary file is uploaded to WordSpinner, the time taken to process the file depends in part on the size and internal complexity of the file. Testing undertaken in July 2023 indicates that for a file consisting of 3,034 headwords, WordSpinner will return a visible output (i.e., a list of webpages into the grey box seen in Figure 5) within fifteen seconds. When the user clicks on one of those webpages to load and view them, it takes two minutes for the pages to load. For a larger file (our test file was 5,700 headwords), which also has more internally complex lexical entries, webpage creation takes less than fifteen seconds, and the webpages load for viewing in eight minutes.[20]

[19] We are grateful to a reviewer for querying what happens in the case of homophones (headwords that are spelled the same but have different meanings). If these are formatted in the dictionary as a single headword with multiple senses, all will appear in the output. If the homophones are formatted as separate lexical entries whose headwords \lx are identical, only the first one will actually appear (all are added to the array as described, but only the first survives due to sorting and hyperlinking operations). This is a limitation of the current system. A pair of homophones in which one is capitalised (e.g., as a placename) and one is uncapitalised (e.g., as a regular word) will both survive in the output as they are not strictly identical headwords. See the online instructions (and Appendix B) for advice about homophones.

[20] These timings were done on a MacBook Air (M1, 2020) with 8 GB of RAM, running on university ethernet, using Chrome. No other browser tabs were open, and no other applications were running in the background.

5. Use and improvements during five dictionary projects

As explained in §4.2, WordSpinner has been iteratively developed through two versions. Version 1 (2015–2018) was a system for creating webpages that were then displayed online (with a password). This was used for Ngarinyman dictionary development. The format of display was Language-to-English only (i.e., target learning language to a user's reference language, which does not necessarily need to be English). In Version 2 (2018–2021), the system has been developed to individual user accounts. This gives the user explicit control over their own dictionary-building area and the abilities to return to that area after they have logged out to continue work on uploaded files and to clear files from the system. This version also includes a search function (rather than just the browser's search function as in Version 1), and an ability to display English-to-Language reversals (i.e., user's reference language to target learning language, again not necessarily with English as the reference language). This system was used in Dhurga, Gurindji, Bilinarra, and Mudburra dictionary development.

§5.1–5.3 provide some details of how WordSpinner has supported dictionary creation and how the dictionary projects have in turn shaped WordSpinner.

5.1 Version 1 (2015–2018)

The initial impetus for this software in 2015 was to create versions of the in-progress dictionary for Ngarinyman, an Australian language in the Northern Territory (NT). This was shared with a specific group of highly literate, young to middle-aged adults for the purposes of checking the accuracy of the dictionary entries in the years leading up to dictionary print publication (2019). The dictionary entries had been compiled over the previous twenty years by a team of more than thirty community members and seven linguists. In 2015, when the software was first developed, the Yarralin community, a key community of Ngarinyman people located in remote NT, did not have mobile phone reception. (Reception eventually arrived in early 2019.) The community did, however, have satellite internet access on computers at the community school, Shire office, and health clinic, so webpages viewable on computers were an accessible option. For this reason, we developed the software to make webpages that could be viewed on a computer or other device, rather than pursuing mobile phone dictionary development.

The ideas behind sharing the dictionary webpages with the specific group of adults were also to check the dictionary over an extended period of time (months, years) and to produce updates based on feedback for further checking. So, the ability to remake and repost the pages easily was important. The idea behind using webpages was also to facilitate intermittent, at-will involvement by highly literate and highly knowledgeable community members who were also extremely busy, holding full-time jobs in key community roles (e.g., Lucy Pedwell in the school at Yarralin, NT, and Sharon Macmillan in the Aboriginal Interpreter Service (AIS), which is based in the regional town centre, Katherine, 300 km away, but who is often travelling for work). These individuals typically had small windows of time, now and then, in which they could look at the in-progress dictionary and provide feedback. Sharon Macmillan's employer fortunately granted her official time (about one hour) each fortnight, to review the webpages. With these busy individuals, it needed to be a technically easy process so they could focus on content, and as their government computers did not allow installation of special software, being able to use just the existing browser was a functional option.

For the purpose of collaborative checking, we made the tool so that it output a set of webpages, one per semantic domain, and with each domain in alphabetical order by Ngarinyman headword. The usefulness of this was that in the broader project, the dictionary content was being prepared incrementally (2015–2016) in semantic domains (to stage the work, and also because some domains were already well-formatted and well-checked and others were not). The dictionary was then also being checked (2015–2018) by semantic domain so that the discussion stayed on a coherent topic and the relationships between words were clear (e.g., words for family and relatives). It thus became a good part of the process to create a webpage per semantic domain when the contents had been checked to some extent, often having been checked verbally with more Ngarinyman Elders during intensive but intermittent fieldwork visits to the community by the linguist team (2015, 2016). Providing the whole entry for view by the younger community adults was useful because sometimes they found errors in parts of the entry other than those which had been the focus of prior checking, or which had slipped through as a persistent misunderstanding. Within each page, the order was alphabetical, which worked for those specific highly literate young to middle-aged adults whom the system was then supporting to help check the dictionary. They were also able to use the browser's search tool.

The main things that this process facilitated were checking with the young adults the meaning(s) of words that were still not clear or holding a discussion about the specific sounds in the words, and consulting about spelling in general. It also enabled us to check the proposed images to illustrate the headwords, as these images were displayed in the webpages.

Because the dictionary was in development by the collaborative team, and we did not want errors to upset anyone else, access to the page was password-restricted, with the password shared among the dictionary team members only. At one stage, we also used the software to make a formatted print-out and posted it to the community team as a hard copy.

5.2 Creating Version 2 (2018–2021)

Improvements were introduced to WordSpinner in its use during the development of the dictionary for Dhurga, a south coast language of New South Wales. Dhurga is taught in a number of primary schools and in adult TAFE (Technical & Further Education) classes. In a context of revitalisation and varying comfort with language learning and technology, this required an accessible and user-friendly dictionary. In 2018, senior Dhurga Elder Patricia Ellis worked with Eleanor Jorgensen, Romi Hill, and Jane Simpson from the Australian National University. Eleanor and Romi reformatted an existing Dhurga wordlist into backslash format in the summers of 2018 and 2019, in consultation with Patricia Ellis and with support from Jane Simpson. During the summer, this team worked with Jesse Tran and Caroline Jones to identify ways the software could be used in the Dhurga dictionary development and what changes to the software would be required to make it more useful. This produced Version 2. The major benefit of using WordSpinner to create the Dhurga dictionary was its automated mass editing and formatting potential. It allowed easy reorganisation of entries with minimal training.

The specific needs of the Dhurga dictionary prompted several changes to the software so that it could support dictionary creation and formatting, rather than just distribution, viewing, and checking, as had been the case for Ngarinyman. Part of the dictionary's user-friendly design was the decision to format the entries in three ways: alphabetical with Dhurga headwords, alphabetical with English headwords, and by semantic domain with English headwords. Only this last one was possible with the initial version of WordSpinner. The system was revised by Jesse Tran so that, as well as the set of webpages organised by semantic domain, a webpage was also output in alphabetical order (Language-to-English). Importantly, dictionary reversal (English-to-Language) was implemented. The system was also revised so that any set of backslash codes was permitted in the input (as long as each entry started with a headword (\lx) field and contained a semantic domain code (\sd)). This produced a more flexible system than Version 1 for Ngarinyman, where the set of backslash codes had been written into the program.

The output display was also modified slightly to be sensitive to the cultural and political context within which the Dhurga dictionary would exist. Concern for the authenticity of source material and absence of L1 speakers meant that the contents of the \so field was changed to "Source" rather than "Speaker." As protection of Indigenous knowledge and language was a concern for Dhurga Elders, a separate profile for each user was developed, meaning a dedicated 'space' for each user to upload their files, and to delete those files from their profile (rather than leaving any language files on the server). This was done to better track access to dictionary material and prevent premature wider distribution.

The dictionary was published in 2020 with 728 Dhurga words, arranged as Dhurga–English, English–Dhurga, and also English–Dhurga by semantic category (Ellis et al. 2020). As a project, it illustrates how WordSpinner and other such technology can have a direct positive influence on the creation of language materials for community initiatives.

5.3 Improvements to Version 2 (2020–2021)

From October 2020 to July 2021, the WordSpinner system was reworked slightly to accommodate master text files for several Ngumpin languages of the NT: Bilinarra, Gurindji, Mudburra, and Ngarinyman. In doing so, webpages could be produced for these languages.[21] The text file for each of these languages includes fields for sound files (\sf and \sfx), images (\pc), and redirects (\cf). Redirects are used to send the user to the standard spelling entry for alternative word spelling entries. These are also relatively large and complex files (about 3,000+ headwords), reflecting good documentation of languages that have, until relatively recently, been spoken regularly by elderly community members. This complexity revealed a few formatting issues (e.g., subentries), which were fixed in the WordSpinner code to address the potentially recurring fields (e.g., example sentences or derived forms). There were three other main changes to WordSpinner at this stage.

The WordSpinner system was reworked to support the inclusion of redirects (i.e., lexical entries with an approximately correct spelling, which send the user to the correct headword spelling, and so function like fuzzy spelling searches). This involved adding a semantic domain \sd field to each redirect entry. Additionally, the \cf field was converted in the output to "See: ...," and a link was implemented to the correct headword on the same webpage. The redirects should make it easier for users to find the word for which they are searching, even if they are not sure of the exact spelling (e.g., whether the word is spelled with *l* or *rl*, *d* or *rd*, *n* or *rn*).

As a second improvement, an option was introduced for the user to choose whether to display the source information for a word (from the \so field) and/or the source or reference information for an example sentence (from the \rf field). This was to respond to different preferences around how much detail to display.

The third main improvement was to accommodate the need for audio from a deceased speaker to be temporarily disabled during Sorry Business (a period of mourning after a First Nations community member in Australia has passed away). Mourning the loss of someone takes on various forms and differs across communities but commonly includes not being allowed to say their name or similar sounding words, seeing their image, or hearing their voice. The latter custom is relevant for WordSpinner.

One proposed approach to the need to temporarily disable audio was to organise the media into folders according to speaker names. This would enable a speaker's content to be disabled by removing the folder from the hosting file system. However, the links remaining in the dictionary webpages would remain, and clicking a linked sound icon on the webpage would result in a poor user experience, with a "file not found" error.

Another approach is recommended based on characteristics of the audio-file naming conventions. If the audio files are named according to a consistent naming system that includes the speaker's initials, the WordSpinner template generation process will copy the speaker's initials from each entry's \sf file name and write those initials into the class attribute of the entry's speaker icons. (Each speaker's initials must be unique for this approach to work.)

To disable a person's audio, a rule for that attribute can be included in the dictionary webpage's style sheet to hide corresponding speaker icons. This removes the sound icons for that speaker from all relevant entries. A server configuration file can also be included in the server hosting to redirect access to audio files containing those speaker's initials, preventing direct browsing to the audio. The style and redirection rules can be removed once the period of Sorry Business has passed, allowing the speaker's audio to be heard publicly once more. This feature allows WordSpinner to adhere to cultural protocols of some First Nations communities.

The process of creating webpages for the Ngumpin languages Bilinarra, Ngarinyman, Mudburra, and Gurindji also provided an opportunity to troubleshoot the system and clean up some inconsistencies in the Toolbox files. As part of this work, we documented the minimum MDF format required for use with WordSpinner and some specific requirements. These are included in the help file on the WordSpinner site. The main points are that (1) each lexical entry must have a headword \lx field and contents in that field, (2) each entry must have a semantic domain \sd field and contents in that field,[22] (3) each lexical entry must be separated from the previous entry by two carriage returns, (4) each lexical entry must have a definition \de field and contents in that field, and (5) each lexical entry must have an \re field (if English-to-Language webpages are desired). If there are no \re contents (as with some place names that have no English names), the \re is still required but should be an empty field. (This last point is very important to note since otherwise the English-to-Language webpages can display reversal information from the preceding lexical entry, leading to inaccu-

[21] The resulting webpages for these four languages are at http://www.ngumpin.org.au.

[22] This allows the system to create a separate webpage for each semantic domain. (If instead an A–Z alphabetical webpage display is desired, that is possible by using the semantic domain field but with a single content (e.g., \sd All), instead of various semantic domain names (e.g., \sd Birds).)

racy in dictionary display). Within each lexical entry, some flexibility is possible; for instance, the WordSpinner system will just display the fields in the order they are in the file, so absolute consistency in the order of fields within each entry, while desirable, is not necessary. WordSpinner recognises typical Toolbox fields like \lx, \ps, \de, \sd, \re, \so, and \dt (and that particular series of fields would be a workable lexical entry format, as a 'minimum lexical entry' structure). The help file displays a template used for the Ngumpin languages, which works with the WordSpinner system. Potential users are encouraged to try their files out with WordSpinner[23] (see Appendix A for login instructions) and to use Grep to manipulate the file systematically if a change to the input format is needed.

6. Future directions

There is still considerable scope for improving WordSpinner, which is open-source. Additional useful functionality would be to allow the user more choice in which fields they wanted displayed in the webpages (the current default is that all fields except \nt are displayed, and there are options to not display source information from \so and/or \rf). This function would make it possible to create multiple versions from the same master dictionary file. For example, it would be possible to produce a low-text webpage with just a headword, meaning, and image and sound for child learners. Another useful function would be to allow for text font, size, and colour to be options chosen by the user. This would allow users to make dictionaries with particular accessibility features (e.g., large font) or a particular colour theme, style, or "look."

There is also considerable scope for further research into how community members and linguists might want to use WordSpinner (or indeed any digital dictionary), and when they do use it, what issues or barriers they might find. Can adults independently find the word they want, easily and efficiently? Are children and teachers able to use the webpages as reference resources or to create learning materials? If so, what are "low hanging fruit" that could perhaps be facilitated in the longer term through interoperability with resource-creation software for First Nations languages?

7. Conclusion

WordSpinner is functional, free, and available for use by community members and linguists who have a backslash-formatted dictionary text file. The webpages that it produces may, as in the case studies described in this paper, be useful in collaborative dictionary development, in checking as in the Ngarinyman case or in formatting as in the Dhurga, Gurindji, Bilinarra, and Mudburra cases described above. Alternatively, the webpages that WordSpinner produces may be useful in displaying a dictionary to learners and teachers. This may increase access to the dictionary if it is publicly available for free on the web or on a password-protected intranet if more privacy is desired. The webpages may also increase access to the contents of the dictionary, for example in making the headwords audible as pronunciation exemplars or models in general of how to pronounce the spelling system. Images can also be added, further increasing access and appeal. Electronic searching can support learners and teachers to locate information more quickly and without the need to use alphabetical order.

The relationship between dictionary webpages (which can be made in different versions and recreated easily) and the master dictionary file (which should persist into the future) is highlighted by WordSpinner software. We hope that engagement with dictionaries via accessible dictionary webpages will highlight the importance of creating and archiving master dictionary files in backslash format. Our goal in creating web dictionaries for First Nations languages is to provide accessible and long-lasting resources that should persist for many generations to come.

[23] https://dictionaryresearch.org (Accessed 2023-12-15.)

Appendix A. User Documentation

How to use WordSpinner

Jesse Tran, Caroline Jones, & Marcel Reverter-Rambaldi
July 2023

We hope these instructions work for you! We know there are improvements that could be made. Please send any questions or comments to caroline.jones@westernsydney.edu.au.

Step 1: Preparing your text file

1.1. Make sure that your file is a text-only .txt file. The dictionary fields must include \lx for the headword and \sd for the semantic domain. There must be a new line at the end of each field, and two new lines separating each lexical entry.

1.2. Remove any header data so that the file starts with the first lexical entry. Also make sure that in the final lexical entry, the last field ends in a new line.

1.3. Using Notepad, Notepad2, or TextWrangler (or another text file editor), ensure that the text encoding is UTF-8. If you are on a Mac, ensure that the file has Windows CRLF line endings.

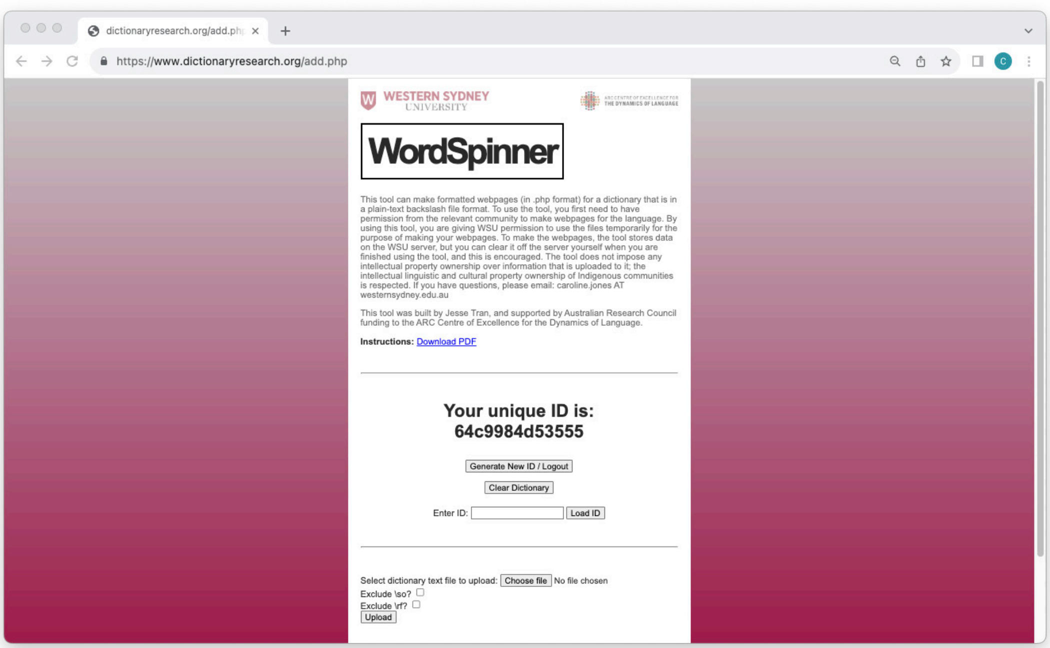
Step 2: Uploading your text file and making your webpages

We recommend closing all applications and all tabs before you start.

2.1. Using a web browser (e.g., Chrome), go to https://dictionaryresearch.org. The password is dictionary2017#

You will be redirected to https://www.dictionaryresearch.org/add.php

You will see a screen like this:



2.2. Copy the unique ID to a safe place. This is your key to log in to your session and to access your files in the future.

If you use another computer to work with WordSpinner in the future or if the ID displayed at the top is different from your ID, type your ID into the "Enter ID" box and click "Load ID."

2.3. Upload the dictionary backslash text file by clicking the "Select dictionary text file to upload: *Choose file*" button. If you have already previously uploaded a backslash text file and would like to upload a new file, it is recommended you click "Clear Dictionary" before you upload the new file.

2.4. Browse to select your dictionary backslash text file. If you wish to suppress the source (\so) and/or reference (\rf) fields, tick the box(es) then click "Upload."

This step uploads your file to WordSpinner's server; the intellectual and cultural property rights in the dictionary are unaffected by using WordSpinner, and your files will not be shared on or used for any purpose except for the webpages you create. We recommend noting your unique ID so that you can actively manage the data on WordSpinner and remove it from the WordSpinner server when you are finished using WordSpinner. The server is cleared of all files every three months, so if you forget your ID, your data will be deleted eventually anyway. For more details, see Step 4 below.

2.5. The file you upload (and the resulting webpages) will stay in storage on the server of your session. See how to clear a session at Step 4: Tidying up.

2.6. Within a few moments, your webpages will be created. These webpages are derived from the categories you have created with the \sd code. They could be semantic domains, lessons 1–10, or a single alphabetical category "AZ." A grey box will appear at the bottom of the screen with your categories in bold font. These are hyperlinks. Click on each to view the webpage that it is associated with.

2.7. If you wish, you can use the search box (underneath the grey box). This will search the webpages you have created. You can look for specific lexical entries, for example.

Step 3: Saving your webpages

3.1. Click once on the name of the semantic domain you want to save.

3.2. Your webpage will now load. This can take between a few seconds (for a small semantic domain of <100 headwords) and up to eight minutes or more (for a full alphabetical list of >5,000 headwords). Scroll down to the very bottom of the window.

3.3. Click on "Download .ZIP." This will download a set of files (your webpage organised into a folder for you, including an empty .img folder).

3.4. Upload the contents of the expanded .zip file to your own server. You will need to upload to your own server (where you want to host the webpages) into the \img folder, all the images and/or audio files that you have referred to within the \pc and \sf or \sfx fields.

3.5. Repeat steps 3.1–3.5 for each webpage you want to save. Do this for the Language-to-English webpages as well as the English-to-Language webpages by clicking on the "English | Language" toggle at the top right of the window.

3.6. To return to the main page, close the tab.

Step 4: Tidying up

4.1. To remove your data off the server, go to the main page. Click "Clear Dictionary."

4.2. If you do not remove your data from the system after generating the dictionary, you are able to return to it by logging in with your ID.

Please note that all data on the server will be automatically deleted every three months (on 1 January, 1 April, 1 July, and 1 October). It is suggested that you keep all your data organised locally on your computer and that when you visit the WordSpinner utility, you use the function to clear your data off the WordSpinner tool after use. WordSpinner is only a utility to make webpages. While the data you upload to WordSpinner will not be shared with others or used for any other purposes, WordSpinner is not intended or recommended as a repository, archive, or publishing platform. We urge you to archive your electronic data in an archive that is funded to provide long-term storage and access.

Appendix B: Input format required for the dictionary master text file

Caroline Jones
July 2023

The system is designed flexibly to work with a variety of backslash-formatted dictionary files. There are some requirements for the input, however. The first field in each lexical entry must be \lx. There must also be a semantic domain field (\sd) for each lexical entry. Any type of semantic domain classification system can be used; it can be designed with the teachers and learners, for example. Each entry must have a reversal field (\re) even if it is empty.

There must be a single space between each backslash code and the contents of that field. Each field must be on a new line. Between lexical entries, there must be two new lines. There must be two new lines at the top of the file and two new lines at the end of the file.

For a simple dictionary, the requirements above are satisfied in a simple lexical entry structure such as this:

\lx headword

\ps part of speech

\de definition

\sd semantic domain

\re reversal (e.g., English lookup word)

\so source

\dt date last modified

\lx headword

\ps part of speech

...

Most dictionaries will need a larger range of fields. As an example, below is a fuller dictionary field structure for which the system works and will provide specific formatting (e.g., boldface for the headword, italics for the part of speech). If other fields are used, the contents will be displayed, but not with any particular formatting.

The field structure below (i.e., a specific selection and ordering of MDF fields) was originally developed by Felicity Meakins at the University of Queensland for Ngumpin language dictionaries. It uses internationally recognised field types, though not all of the MDF options. (Note: When this field structure is processed, the notes (\nt) and gloss (\ge) fields are suppressed and not visible in the output webpages.)

--

\lx headword

\sf headword sound file name (e.g., corkwood.wav)

\ps part of speech

\sn 1 (no numeral is entered if there is only one sense)

\ge gloss

\de definition

\sd semantic domain (you can use any set of categories you wish, such as semantic domains or other organising principles (e.g., Lessons 1–10))

\re reversal

\sy synonym

\va variant (enter phonological variants separated with commas)

\sc scientific name

\so source of information (speaker initials or published reference)

\xv example sentence

\sfx example sentence audio file name

\xe English translation of example sentence

\rf reference for example sentence

\ue usage (use for marked dialect information) (e.g., “This word is used at Yarralin.”)

\ee encyclopaedic information (public)

\cf lookup entry

\pc image file name (e.g., corkwood.jpg)

\nt notes (e.g., among linguists)

\se subentry

\de definition

\xv example sentence

\sfx example sentence audio file name

\xe English translation of the example sentence

\rf reference for the example sentence

\dt date last modified (e.g., 10/Aug/2015)

--

A note about homophones and how they are processed by WordSpinner:

In any dictionary, there are some homophones (headwords that are spelled the same but have different meanings). If these are formatted in the dictionary as a single headword with multiple senses, all will appear in the WordSpinner output. (This is a recommended way of handling homophones for the purpose of using WordSpinner, if it is possible.)

Warning: If the homophones are formatted as separate lexical entries whose headwords \lx are identical, only the first one will actually appear (all are added to the array as described, but only the first survives due to sorting and hyperlinking operations). This is a limitation of the current WordSpinner system. A pair of homophones in which one is capitalised (e.g., as a place name) and one is uncapitalised (e.g., as a regular word) will both survive in the output as they are not strictly identical headwords. Full homophones could be differentiated by hard-coding numbers within the headword field so that the lexical entries are not identical as headwords.

References

Archibald-Binge, Ella. 2019. 'A symbol of strength': Dictionary immortalises NT Aboriginal language. *NITV News*, July 4. (https://www.sbs.com.au/nitv/article/2019/07/04/symbol-strength-dictionary-immortalises-nt-aboriginal-language) (Accessed 2019-07-04.)

Austin, Peter. 2003. Australian Aboriginal lexicography. In Hartmann, Reinhart R. K. (ed.), *Lexicography: Critical concepts in linguistics*, 288–294. London: Routledge.

Austin, Peter & David Nathan. 1996. *Kamilaroi/Gamilaraay web dictionary*. (https://www.dnathan.com/language/gamilaraay/dictionary/index.html) (Accessed 2023-12-15.)

Birch, Bruce. 2013. The Ma! Project: Crowdsourcing software for language documentation. In *Research, records and responsibility (RRR): Ten years of the Pacific and Regional Archive for Digital Sources in Endangered Cultures (PARADISEC)*. Melbourne: PARADISEC. (http://hdl.handle.net/2123/9858)

Corris, Miriam, Christopher Manning, Susan Poetsch, & Jane Simpson. 2004. How useful and usable are dictionaries for speakers of Australian Indigenous languages? *International Journal of Lexicography* 17(1). 33–68. doi:10.1093/ijl/17.1.33

Coward, David F. & Charles E. Grimes. 1995. *Making dictionaries: A guide to lexicography and the Multi-Dictionary Formatter, version 1.0*. Waxhaw, North Carolina: Summer Institute of Linguistics.

Ellis, Patricia, Kerry Boyenga, & Waine Donovan. 2020. *The Dhurga dictionary and learners' grammar: A south-east coast, NSW Aboriginal language*. Canberra: Aboriginal Studies Press.

Endangered Languages and Cultures. 2009. Wunderkammer. *Endangered Languages and Cultures*, January 29. (https://www.paradisec.org.au/blog/2009/01/wunderkammer/) (Accessed 2023-12-15.)

Frawley, William, Kenneth C. Hill, & Pamela Munro (eds.). 2002. *Making dictionaries: Preserving Indigenous languages of the Americas.* Berkeley: University of California Press.

Garrett, Andrew. 2018. Online dictionaries for language revitalization. In Hinton, Leanne, Leena Huss, & Gerald Roche (eds.), *The Routledge handbook of language revitalization*, 197–206. London: Routledge.

Hill, Clair. 2012. Review of Wunderkammer Import Package: A tool for the display of multimedia dictionaries on mobile phones. *Language Documentation & Conservation* 6. 282–291. (http://hdl.handle.net/10125/4570)

Himmelmann, Nikolaus P. 2006. Language documentation: What is it and what is it good for? In Gippert, Jost, Nikolaus P. Himmelmann, & Ulrike Mosel (eds.), *Essentials of language documentation*, 1–30. New York: Mouton de Gruyter. doi:10.1515/9783110197730.1

Jones, Caroline, Eva Schaultze-Berndt, Jessica Denniss, & Felicity Meakins (compilers). 2019. *Ngarinyman to English dictionary*. Canberra: Aboriginal Studies Press.

Littell, Patrick, Aidan Pine, & Henry Davis. 2017. Waldayu and Waldayu Mobile: Modern digital dictionary interfaces for endangered languages. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, Honolulu, 6–7 March. doi:10.18653/v1/W17-0119

Manning, Christopher D., Kevin Jansz, & Nitin Indurkhya. 2001. Kirrkirr: Software for browsing and visual exploration of a structured Warlpiri dictionary. *Literary and Linguistic Computing* 16(2), 135–151. doi:10.1093/llc/16.2.135

Maxwell, Mike. 2004. From legacy lexicon to archivable resource. In *Language Resources and Evaluation Conference, LREC 2004*. (https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/lrec2004-from-legacy-lexicon-to-archivable-resource.pdf) (Accessed 2023-12-15)

McElvenny, James. 2008. Mobile phone dictionaries. *Endangered Languages and Cultures*, July 17. (https://www.paradisec.org.au/blog/2008/07/mobile-phone-dictionaries/) (Accessed 2023-12-15.)

McElvenny, James. 2010. Mobile phone dictionaries for small languages: The Whitesands electronic dictionary. In Dykstra, Anne & Tanneke Schoonheim (eds.), *Proceedings of the XIV Euralex International Congress*, Leeuwarden, 6–10 July, 1505–1507. Leeuwarden: Fryske Akademy. (https://www.euralex.org/elx_proceedings/Euralex2010/142_Euralex_2010_10_MCELVENNY_Mobile%20phone%20dictionaries%20for%20small%20languages_the%20Whitesands%20electronic%20dictionary.pdf) (Accessed 2023-12-15.)

McElvenny, James & Aidan Wilson. 2009. Electronic dictionaries for language reclamation. (Paper presented at the 1st International Conference on Language Documentation and Conservation, Honolulu, 16–17 March.) (http://hdl.handle.net/10125/5062)

Nash, David. 2022. Appendix 5 | The emergence of the Warlpiri Dictionary: A chronological account. In Laughren, Mary, Kenneth Hale, Jeannie Egan Nungarrayi, Marlurrku Paddy Patrick Jangala, Robert Hoogenraad, David Nash, & Jane Simpson (compilers), *Warlpiri encyclopaedic dictionary*, 1375–1383. Canberra: Aboriginal Studies Press.

Ransdell-Green, Margaret. 2021. Kala dictionary making with the WOLF lexicography software: A collaborative effort. (Presentation at the 7th International Conference on Language Documentation and Conservation, Honolulu, 4–7 March.) (http://hdl.handle.net/10125/74492)

Richards, Mark, Caroline Jones, Francesca Merlan, & Jennifer MacRitchie. 2019. Revitalisation of Mangarrayi: Supporting community use of archival audio exemplars for creation of language learning resources. *Language Documentation & Conservation* 13. 253–280. (http://hdl.handle.net/10125/24865)

Simpson, Jane. 2003. Representing information about words digitally. In Barwick, Linda, Allan Marett, Jane Simpson, & Amanda Harris (eds.), *Researchers, communities, institutions, sound recordings*. Sydney: PARADISEC. (http://hdl.handle.net/2123/1517)

Wilson, Stephen, Mark Harvey, Lulu Martin Dalpbalngali, Helen Liddy Emorrotjba, Paddy Huddlestone Benbo, Clara McMahon Gumbirtbirtda, & Lenny Liddy Gapbuya. 2001. *The Wagiman online dictionary*. Canberra: Australian Institute of Aboriginal and Torres Strait Islander Studies. (https://aphasialab.org/wagiman/home.html) (Accessed 2023-12-15.)

Caroline Jones
caroline.jones@westernsydney.edu.au
orcid.org/0000-0001-6277-8262

Jesse Tran
orcid.org/0000-0001-5865-067X

Eleanor Jorgensen
orcid.org/0000-0001-6703-094X

Romi Hill
orcid.org/0009-0009-6180-6685

Jane Simpson
orcid.org/0000-0002-7487-1234

Felicity Meakins
orcid.org/0000-0003-4487-4351

Marcel Reverter-Rambaldi
orcid.org/0000-0002-4116-5376

Gari Tudor-Smith
orcid.org/0000-0002-9664-6150

Paul Williams
orcid.org/0000-0002-8137-8085

Clair Hill
orcid.org/0000-0002-3937-0813

Mark Richards
orcid.org/0000-0002-8630-9097