Tools facilitating better use of online dictionaries: Technical aspects of Multidict, Wordlink and Clilstore

Caoimhín P. Ó Donnaíle
Sabhal Mòr Ostaig
An t-Eilean Sgitheanach
IV44 8RQ, UK
caoimhin@smo.uhi.ac.uk

Abstract

The Internet contains a plethora of openly available dictionaries of many kinds, translating between thousands of language pairs. Three tools are described, Multidict, Wordlink and Clilstore, all openly available at multidict.net, which enable these diverse resources to be harnessed, unified, and utilised in ergonomic fashion. They are of particular benefit to intermediate level language learners, but also to researchers and learners of all kinds. Multidict facilitates finding and using online dictionaries in hundreds of languages, and enables easy switching between different dictionaries and target languages. It enables the utilization of page-image dictionaries in the Web Archive. Wordlink can link most webpages word by word to online dictionaries via Multidict. Clilstore is an open store of language teaching materials utilizing the power of Wordlink and Multidict. The programing and database structures and ideas behind Multidict, Wordlink and Clilstore are described.

1 Introduction

At `multidict.net` three tools are to be found, Multidict, Wordlink and Clilstore. Their development was funded by EC projects with the aim of developing and sharing tools for language learning, and thanks to this they are a freely and openly available resource. They support not only the major European languages, but also place a particular emphasis on supporting minority languages including the Celtic languages. They also currently support scores of non-European languages and have the potential to support many more.

The central idea behind them is that one of the best ways of learning a language is to use authentic materials as early as possible - materials which are of interest for their own sake. This is the "**CLIL**", "Content and Language Integrated Learning", in the name "Clilstore". In the past, this would have meant either the students laboriously looking up word after word in the dictionary, or else the teacher laboriously preparing glossaries of the most difficult words for each piece of reading material. Good authentic content is easy to find via the Internet for most subjects in most languages, but preparing the glossaries was tedious.

For the students, online dictionaries, and there are many of them, sped up the process of looking up words compared to the old paper dictionaries. But it was still tedious typing in words, and then typing or copying them in again to try them in another dictionary. Far better if you could just click on a word in a text to look it up. This is the idea behind **Wordlink**. It takes any webpage and modifies the html so that every word is linked to online dictionaries while the presentation of the page remains the same.

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Page numbers and proceedings footer are added by the organisers. Licence details: `http://creativecommons.org/licenses/by/4.0/`

Automatic glossing of text as an aid to learners is not an idea unique to this project. It is used by the Rikaichan[1] Firefox add-on for Japanese, by the BBC Vocab[2] facility for Welsh and Gaelic, by the Readlang[3] website, by the PIE[4] Chrome add-on for English, and by many e-books. While these systems have many advantages, they also have severe restrictions compared to Wordlink: restrictions to particular languages, or particular browsers, or particular websites, or particular in-house dictionaries. Wordlink differs in that it attempts to generalize to very many languages and to harness the many freely available online dictionaries.

The earliest versions of Wordlink contained the code and knowledge required to link to a range of online dictionaries translating to various target languages. But the list quickly became ridiculously long and it was realized that the work of selecting and accessing different dictionaries needed to be hived off to a separate facility. So **Multidict** was created, and is a tremendously useful standalone facility in its own right.

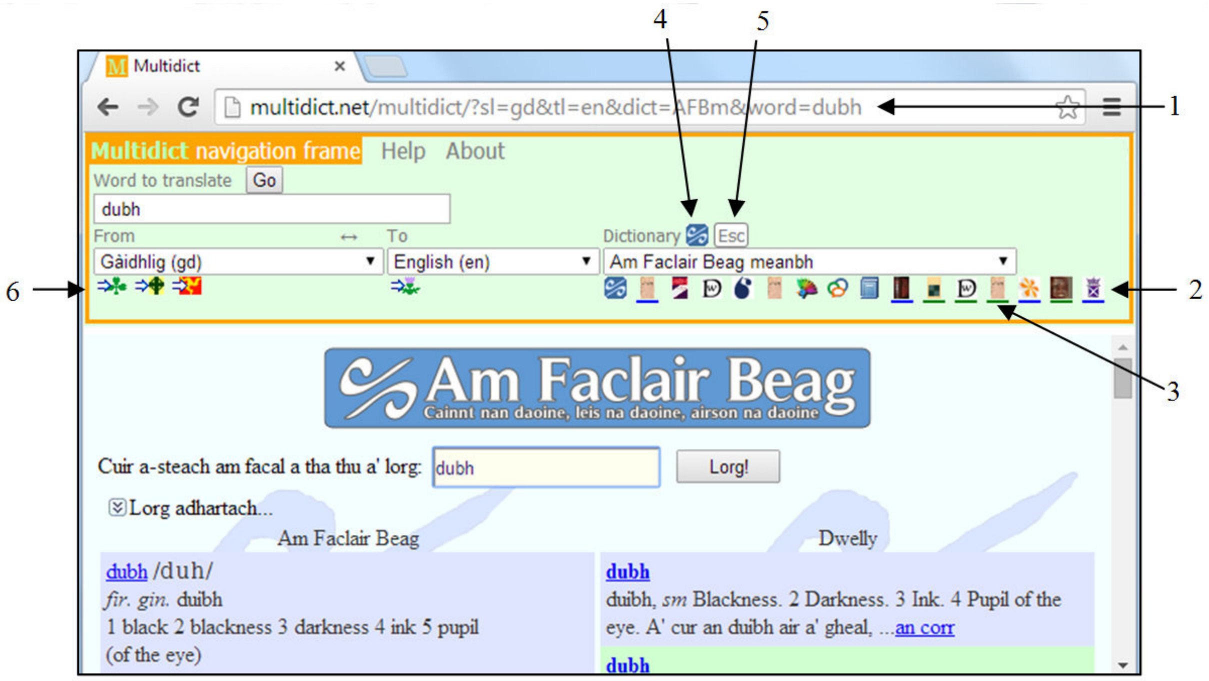
Finally **Clilstore** was created to make it easy for language teachers to create materials and lessons utilizing the power of Wordlink and Multidict, and to make it easy for students and teachers to find material of interest stored openly in Clilstore. The great thing about Clilstore is that it enables students to access interesting material which would otherwise be a bit too difficult for them to cope with. It has proved to be particularly useful to intermediate level learners, and to learners coming from cognate languages.

We now look at the technical workings behind each of these three tools in turn.

2 Multidict

2.1 The interface

Here is what Multidict looks like in use:



dubh /duh/
fir. gin. duibh
1 black 2 blackness 3 darkness 4 ink 5 pupil (of the eye)

dubh
duibh, *sm* Blackness. 2 Darkness. 3 Ink. 4 Pupil of the eye. A' cur an duibh air a' gheal, ...an corr

dubh

Figure 1. The Multidict interface

The section at the top is the “Multidict navigation frame” which controls dictionary selection and lookup. (Yes, Multidict uses old-fashioned frames[5].) Below that is the frame containing the output returned by the online dictionary. In this case Multidict is being used to look up the Gàidhlig word

[1] http://rikaichan.mozdev.org (this and all web references are as accessed on the date of writing, 2014-06-24)
[2] http://www.bbc.co.uk/cymru/vocab/
[3] http://www.learngaelic.net/advanced/lganla/index.jsp?lang=gd
[4] https://sites.google.com/site/phoneticallyintuitiveenglish/using-pie/getting-a-word-s-meaning
[5] http://www.w3.org/TR/html401/present/frames.html
http://www.w3.org/TR/html5/obsolete.html

dubh in the Gàidhlig to English dictionary *Am Faclar Beag meanbh*[6] (the concise version of *Am Faclair Beag*[7]).

Note (**1**) the url which can be used to refer to the dictionary output for this word. This can be particularly useful in the case of dictionaries which do not themselves have any way of linking to their output via a url. The “sl” stands for “source language” and “tl” stands for “target language”.

Note (**2**) the row of 16x16 pixel **favicons** for dictionaries. Clicking on one of these switches you to the corresponding dictionary. They give the navigation frame a cluttered appearance, but once you get to know them they are much quicker and more convenient than selecting a dictionary using the dropdown selector. If the dictionary website has its own favicon, as most do, then Multidict uses that. If not, we try to construct a mnemonic favicon for the dictionary using the dictionary’s own colours. Both in the row of favicons and in the dictionary dropdown, the dictionaries are placed in some kind of compromise order of preference. Note (**3**) that some favicons have an underline. This signals that the dictionary is a **page-image** dictionary where the user will have to scan around by eye on the page to find the word in question. More about page-image dictionaries in section 2.7 below. An overline where present above a favicon signals that the dictionary is a concise version, perhaps designed for mobile phones, which can often be very useful if the dictionary is being used together with Wordlink.

Note (**4**) the favicon for the current dictionary, and (**5**) the **Esc** button which provides a convenient way of escape from Multidict’s frames to the dictionary’s own homepage. Multidict is in fact a very convenient way of finding dictionaries and we have no desire to keep users on Multidict if they prefer to head off and use the dictionary directly.

Multidict does not itself have any dictionary information, but relies entirely on directing users to online dictionaries. So we need to be fair and maintain **good relations with dictionary owners**. Multidict makes a point of never “scraping”[8], never even caching information from dictionary pages. Output is always presented exactly as it comes from the dictionary, complete with any advertising. In fact, whenever possible, Multidict operates by sending a simple HTTP “redirect” to redirect the user’s browser to the dictionary page. Multidict advertises to dictionary owners that they can ask for their dictionary to be removed from Multidict’s database at any time for any reason, but no dictionary owner has ever requested this.

Note (**6**) the “favicon” symbols for switching to **closely related languages**. This makes it easy, for example, to switch and look for the word *dubh* in Irish dictionaries instead of Scottish Gaelic. For most languages we just use language codes for these symbols, but for the Celtic languages we have colourful symbols available. The same is possible for the target language, although in the example above the only symbol shown is the “Gàidhlig” symbol for switching to Gàidhlig-Gàidhlig monolingual dictionaries. To support this system, the Multidict database has two tables holding information on closely related languages. Two tables because “closely related” for the purposes of the target language field may not be the same as closely related for the purposes of the source language field. There would be no point in trying an “sr-Latn” (Serbian in Latin script) word in an “sr” (Serbian in Cyrillic script) dictionary, but someone who understood “sr” could be expected to understand “sr-Latn”.

2.2 The database behind it

How does Multidict work? For many dictionaries, very very simply. If when you look up the word *dubh* at `friendlydict.org`, you notice that the url is

```
http://friendlydict.org/find?facail=dubh
```

then you can be sure that by simply replacing `dubh` with `geal` in the url, you would look up the word *geal*. For such dictionaries, the Multidict database would store the string

```
http://friendlydic.org/find?facail={word}
```

and when the time came to look up a word, Multidict would simply replace `{word}` with the word in question and redirect the results frame to this address.

However, for many dictionaries, both good ones and less good, things are not so simple. Their html form submission uses **POST** method instead of **GET** method and there is no sign of a nice url containing the word to search for. In this case, Multidict has to construct and send an http POST request. It

6 http://www.faclair.com/m/
7 http://www.faclair.com
8 The practice of extracting partial information from webpages on another site: http://en.wikipedia.org/wiki/Web_scraping

does this using the HTTP_Request2 PEAR[9] class. (PEAR being a repository of software for the PHP language.) Multidict captures the response to the request and despatches it to the results frame.

Multidict, Wordlink and Clilstore are written in PHP, and behind them is a mySQL (or MariaDB[10] to be precise) database. The database has a `dict` table with a record for each dictionary, storing the long name, the favicon and the dictionary's homepage address.

However, many dictionaries serve several languages, and the main business is done by the table `dictParam`, which is indexed by `(dict, sl, tl)`. This table stores the url, as described above, any post parameters required, and has many other fields. A field called `message` can contain a tip to be displayed to users in the navigation frame, such as "Right-click to zoom". A field `charextra` can specify certain different kinds of extra processing to be applied to the word before lookup to satisfy the peculiarities of particular dictionaries. Some dictionaries require accents to be stripped from the word, some require them to be urlencoded[11]. The Irish Dineen[12] dictionary requires 'h's to be stripped from the word to convert to old spelling and dictionary order, and this is indicated by the string "striph" in the `charextra` field. A field `handling` specifies any particular handling required to obtain the output from the dictionary. The best behaved dictionaries get the value "redirect". Some particularly awkward dictionaries which require POST parameters and only accept requests from the user's browser get the value "form". This causes Multidict to construct a form in the results frame, fill in the search word, and cause the user's browser via Javascript to immediately submit it. Thus Multidict has a whole range of clever tricks and tools available to it, which means that it manages to handle between 80% and 90% of all dictionaries we have attempted to link to.

2.3 Language codes

Multidict currently tries to use IETF language codes[13] both externally and internally. i.e. It uses a two-letter ISO 639-1[14] language code such as "en", "fr", "de", "ga", "gd" if such is available, or a three letter ISO 639-3[15] language code such as "sco", "sga" when no two-letter code is available, and it sometimes makes use of country code and script code extensions such as "pt-BR" and "sr-Latn". When these are inadequate, such as for historic languages and dialects, it turns to LinguistList[16] codes for inspiration: e.g. "non-swe" (Old Swedish[17]), and "oci-ara" (Aranese[18]).

Where ISO 639-3 equates a two-letter language code with a three letter code denoting a **macrolanguage[19]**, as in the case of Latvian lt=lav which also includes Latgalian, Multidict uses the ISO 639-3 code for the precise language, in this case "lvs" for Standard Latvian. This differs from Google Translate, for example, which continues to use the two-letter code code for the dominant language in the macrolanguage grouping. Other languages where similar questions arise include Estonian et/ekk, Malay ms/zsm, Albanian sq/als, Azari az/azj, Uzbek uz/uzn, Persian fa/pes, Guarani gn/gug, Swahili sw/swh.

2.4 Closely related languages

As we increasingly try to cater for minority languages and dialects, the questions of how to deal with closely related languages become ever greater. On the one hand, we want to distinguish European Portuguese, currently coded as "pt", and Brazilian Portuguese, "pt-BR", especially if the dictionary site itself clearly distinguishes them among its language choices. On the other hand, we don't want users to be unable to find dictionaries which might be very useful to them, simply because of a small

9 http://pear.php.net/package/HTTP_Request2/
10 https://mariadb.org
11 http://www.php.net//manual/en/function.urlencode.php
12 http://glg.csisdmz.ul.ie
13 https://tools.ietf.org/html/rfc5646
14 http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes
15 http://www-01.sil.org/iso639-3/
16 http://linguistlist.org/forms/langs/find-a-language-or-family.cfm
17 http://multitree.org/codes/non-swe
18 http://multitree.org/codes/oci-ara
19 http://www-01.sil.org/iso639-3/macrolanguages.asp
http://en.wikipedia.org/wiki/ISO_639_macrolanguage

difference in language code. The “closely related languages” feature in the Multidict interface goes a very small way towards addressing this difficulty, but the problem requires more thought.

A webpage[20] available via the Multidict help system lists all the languages currently handled by Multidict. It lists languages ordered by language family, then sub-family and so on. Closely related languages are therefore located close together, and the webpage can be used to maintain Multidict’s tables of closely related languages. To achieve this ordering, the Multidict database links each of its language codes to the corresponding LinguistList code, and holds a copy of the LinguistList Multitree[21] Composite Tree. However, because the Composite Tree provides nothing but a tree structure, albeit a tremendously useful finely-detailed tree structure, it is in itself inadequate for defining the required linearization of the tree. We always prefer to place the most closely related branches (closely related by geography if nothing else) adjacent to one another, rather than the children of each node being listed in some random order (as they currently are in Multitree itself, which places Baltic languages next to Celtic and Armenian, rather than next to Slavic). To do this, in Multidict’s copy of the Composite Tree, we maintain, where relevant to Multidict, an ordering of the children of a parent node. This has to be laboriously researched each time a language is added to Multidict. It would be very useful if this ordering information were to be provided as a resource together with the LinguistList Composite Tree.

2.5 “n×n” dictionaries

Most online dictionaries only handle a limited number of language pair `(sl, tl)` combinations, and each of these is given a separate record in the `dictParam` table. However, some online dictionaries can translate between any of n×n language pairs. Most notably in recent years, Glosbe[22] and Global Glossary[23] translate surprisingly successfully between any pair out of hundreds of languages. To harness the tremendous power of these “n×n” dictionaries without cluttering the `dictParam` table with tens of thousands of records, the Multidict database uses the following tactic. In the sl field in the `dictParam` table, a “¤” symbol is placed, and this indicates to Multidict to refer to a separate table `dictLang` to obtain a list of the n languages which this particular n×n dictionary handles. The table can also translate between the language code used by Multidict and a different language code used by the dictionary. In the `dictParam` table, the url required for linking to the dictionary can (as can also the POST parameters) contain placeholders for sl and tl, such as for example:

```
  http://friendlydic.org/find?from={sl}&to={tl}&facail={word}
```

When Multidict looks up a word, it substitutes the relevant sl and tl. The tl field in the `dictParam` record for the n×n dictionary also contains a “¤” symbol if this is truly an n×n dictionary, including monolingual pairs such as English-English. If it is actually an “n×(n-1)” dictionary excluding monolingual pairs, this is denoted by placing instead an “x” in the tl field.

2.6 Quality ranking

To try to place the “best” dictionaries at the top of the list in the user interface, and also to ensure that the “best” dictionary for the language-pair is used by default, the `dictParam` table stores a “quality” figure for each dictionary. Of course, this is necessarily a compromise. What is best for one purpose might not be best for another. And things get messy when it comes to n×n dictionaries. Multidict already records and defaults to the previous dictionary which the user used for that language-pair. It might be best, instead of over-relying on a “quality” figure, to extend this recording system to the second and third most recent dictionaries used, or perhaps move to a system based on usage statistics.

2.7 Web Archive dictionaries

Online dictionary resources are often very scarce for minority languages. However, many excellent old paper dictionaries are now available in page-image format on the Web Archive at `www.archive.org`[24], and also on Google Books[25]. The wonderful thing is that these dictionaries

[20] http://multidict.net/multidict/languages.php
[21] http://multitree.linguistlist.org
[22] http://glosbe.com
[23] http://www.globalglossary.org
[24] https://archive.org/details/texts

can be addressed by url on an individual page basis. So all we need to do to make the dictionary available via Multidict is to provide Multidict with a table giving it the first word on every page of the dictionary. Or actually, the last word on every page works slightly better because of the technicality that several headwords can have the same spelling. Providing such a table sounds like a daunting task, but in fact, by getting very ergonomically organized the time can be reduced to a few seconds per page, meaning that even a 1000 page dictionary can be dealt with in a few hours. To date, 23 such page-image dictionaries have been made available via Multidict (counting the reverse direction separately in 5 cases), namely 8 for Scottish Gaelic; 2 Irish; 1 Old Irish; 3 Manx; 1 Cornish; 1 Old English; 1 Middle English; 3 Nyanja and 3 Maori. In total, about 55,000 pages have been indexed. The biggest example is that all 4323 columns of the Old Irish eDIL[26] dictionary have been indexed, and in fact eDIL is currently more usable for most purposes via Multidict than using its own native search interface. Although the native search will search the whole dictionary, which can sometimes be wonderfully useful, it will find nothing at all if the search word is not specified exactly as written in the dictionary, including all accents and hyphens. With the vagaries of Old Irish spelling, it can be more useful to take the user to the right spot in alphabetic order as Multidict does, leaving him or her to complete the search by eye.

To enable access to these page-image dictionaries, Multidict uses two tables, `dictPage` which records the first (or last) word on every page, and `dictPageURL` which records the url templates required to translate these page numbers into urls. The mechanism can also cope with dictionaries which are split into several volumes, as is Dwelly in the Web Archive . A program `dictpage.php` does the job of redirecting the browser to the appropriate url.

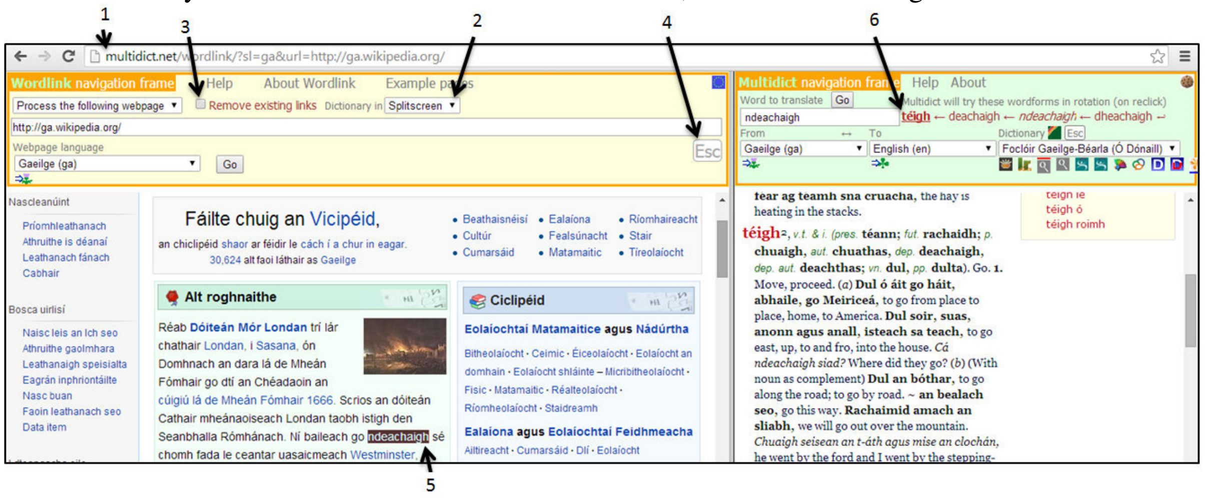
2.8 Statistics

Multidict currently handles 271 different online dictionaries - there are 271 records in the `dict` table. The dictParam table has 2101 records covering 1041 language pairs, but the numbers would be tens of thousands higher if the n×n dictionaries Glosbe and Global Glossary were included. Multidict currently handles 202 languges, or 140 if the n×n dictionaries are excluded.

3 Wordlink

3.1 The interface

In the example shown below, Wordlink is being used to view the Irish Wikipedia homepage. At the top is the Wordlink navigation frame which is used for control. Below that is a frame with what looks exactly like the Wikipedia page, but it is in fact a doctored version, with the html modifed by Wordlink to link every word to online dictionaries via Multidict, as shown on the right.



1
3
2
4
6
5
multidict.net/wordlink/?sl=ga&url=http://ga.wikipedia.org/
Wordlink navigation frame
Help
About Wordlink
Example pages
Process the following webpage
Remove existing links
Dictionary in Splitscreen
http://ga.wikipedia.org/
Webpage language
Gaeilge (ga)
Go
Esc
Multidict navigation frame
Help
About
Word to translate
Go
ndeachaigh
Multidict will try these wordforms in rotation (on reclick)
téigh ← deachaigh ← ndeachaigh ← dheachaigh ←
From
To
Dictionary
Esc
Gaeilge (ga)
English (en)
Foclóir Gaeilge-Béarla (Ó Dónaill)
Nascleanúint
Príomhleathanach
Athruithe is déanaí
Leathanach fánach
Cabhair
Bosca uirlisí
Naisc leis an lch seo
Athruithe gaolmhara
Leathanaigh speisialta
Eagrán inphriontáilte
Nasc buan
Faoin leathanach seo
Data item
Beathaisnéisí
Cultúr
Cumarsáid
Ealaíona
Fealsúnacht
Matamaitic
Ríomhaireacht
Stair
Tíreolaíocht
Alt roghnaithe
Ciclipéid
Eolaíochtaí Matamaitice agus Nádúrtha
Bitheolaíocht · Ceimic · Éiceolaíocht · Eolaíocht an domhain · Eolaíocht shláinte – Micribitheolaíocht · Fisic · Matamaitic · Réalteolaíocht · Ríomheolaíocht · Staidreamh
Ealaíona agus Eolaíochtaí Feidhmeacha
Ailtireacht · Cumarsáid · Dlí · Eolaíocht
téigh le
téigh ó
téigh roimh

Fáilte chuig an Vicipéid,
an chiclipéid shaor ar féidir le cách í a chur in eagar.
30,624 alt faoi láthair as Gaeilge

Réab **Dóiteán Mór Londan** trí lár chathair Londan, i Sasana, ón Domhnach an dara lá de Mheán Fómhair go dtí an Chéadaoin an cúigiú lá de Mheán Fómhair 1666. Scrios an dóiteán Cathair mheánaoiseach Londan taobh istigh den Seanbhalla Rómhánach. Ní baileach go ndeachaigh sé chomh fada le ceantar uasaicmeach Westminster,

fear ag teamh sna cruacha, the hay is heating in the stacks.

téigh2, *v.t. & i. (pres.* **téann;** *fut.* **rachaidh;** *p.* **chuaigh,** *aut.* **chuathas,** *dep.* **deachaigh,** *dep. aut.* **deachthas;** *vn.* **dul,** *pp.* **dulta**). Go. **1.** Move, proceed. (*a*) **Dul ó áit go háit, abhaile, go Meiriceá,** to go from place to place, home, to America. **Dul soir, suas, anonn agus anall, isteach sa teach,** to go east, up, to and fro, into the house. *Cá ndeachaigh siad?* Where did they go? (*b*) (With noun as complement) **Dul an bóthar,** to go along the road; to go by road. ~ **an bealach seo,** go this way. **Rachaimid amach an sliabh,** we will go out over the mountain. *Chuaigh seisean an t-áth agus mise an clochán,* he went by the ford and I went by the stepping-

Figure 2. The Wordlink interface

[25] http://books.google.com
[26] http://edil.qub.ac.uk/dictionary/search.php

Note (**1**) the **url**:
`http://multidict.net/wordlink/?sl=ga&url=http://ga.wikipedia.org/`
which can be used to refer to the wordlinked page. An additional paramater `navsize=1` can be used to reduce the navigation frame away to 1 pixel size if it is not required. If the url is specified in the form `url=referer`, the url is taken from the referer information in the http request. This means that by adding a link of this form to every page of a website, each page is linked to a Wordlinked version of itself for the benefit of language learners. This can be seen in use on the Fòram na Gàidhlig[27] website.

Note (**2**) the choice of **mode**, "Splitscreen" which causes Multidict and the dictionary results to be shown in a frame on the right. Wordlink has three other choices of mode available "New tab", "Same tab" and "Popup". Although Splitscreen is the default and is overwhelmingly the most used, the other modes could actually be very useful on smaller screens.

Note (**3**) the option to "**Remove existing links**". By default, Wordlink does not actually link every word to a dictionary lookup. If you click on the word *Dóitean*, it will take you instead to a Wordlinked version of the *Dóiteán Mór Londan* Wikipedia page. "Remove existing links" does what it says and will instead ensure you are taken to a dictionary lookup of *Dóiteán*.

Note (**4**) the **Esc** button. Wordlink like Multidict makes it easy for you to escape from its frames to the webpage itself.

Note (**5**) that the word *ndeachaigh* has been clicked on to find it in the dictionary, and it is therefore **highlighted** and remains highlighted until another word is clicked. This small point is of major importance. Very often the user will need to scroll the dictionary information (as indeed in this example), and it is essential that the word be highlighted to make it easy to look back and continue reading.

Note (**6**) that although Multidict has been handed the wordform *ndeachaigh* by Wordlink, it has chosen instead to look up *téigh*, which it thinks is probably the appropriate "lemma", the dictionary headword to look up, and it has also lined up a row of other lemma suggestions to be tried in turn if the user reclicks "ndeachaigh" or clicks "Go" in Multidict. This new **lemmatization** feature built into Multidict has resulted in a big improvement in the user experience when using Wordlink and Clilstore. Some online dictionaries can do their own lemmatization, but many good dictionaries do not. And even when the dictionary itself offers excellent lemmatization suggestions, as does *Ó Dónaill*[28] in the example above, the new "click to retry" feature is so slick to use that it can be much quicker to just reclick and let Multidict do the work. The feature is described more fully in section 3.4 below.

3.2 The Wordlink program

The Wordlink program, like all the facilities at `multidict.net` is written in PHP[29]. It first sends off an HTTP request to fetch the webpage to be processed. It then converts it to UTF-8 character encoding[30] if it is not already in UTF-8, because all the facilities work internally entirely in UTF-8. It then processes the page to (1) convert existing links into links to Wordlinked pages (if this has not been switched off by "Remove existing links"), and (2) convert each word in runs of text into a link to make Multidict look up that word. We will not go into the details, but suffice it to say that it is not an easy task, and it is essential to ensure that relative links to images, stylesheets and Javascript libraries are all appropriately converted. It currently works by processing the html serially, but it would probably be better to convert it to use an html parser and then traverse the resulting DOM tree.

Wordlink does not work well with all webpages, particularly flashy games pages or TV company websites and suchlike. But it produces good to excellent results with a good 90% of the more textual webpages likely to be of interest to language learners. With well-behaved pages such as Wikipedia it works perfectly. It does not work at all with webpages requiring a login, such as Facebook or pages in virtual-learning environments. To do this would require it to store and forward user-credentials and would get us into the very iffy field of trust relationships. Nor does it work with the https (secure http) protocol.

27 http://www.foramnagaidhlig.net/foram/
28 http://breis.focloir.ie/ga/fgb/
29 http://www.php.net
30 http://en.wikipedia.org/wiki/UTF-8

3.3 Word segmentation

Wordlink links "words" to dictionaries, and for most languages it identifies words by the whitespace or punctuation characters surrounding them. This means that it does not deal with collocations or phrases or even hyphenated words such as "trade-union". In such cases, the user can always type additional text into the Multidict search box. But it would be nice if some sort of Javascript or browser extension could be devised to allow the user to select phrases with the mouse and look them up.

Breton and **Catalan** presented Wordlink with a slight problem, because "c'h" in Breton is regarded as a letter, as is "l·l" in Catalan, and at first Wordlink was splitting the word at what it thought was a punctuation character. This was easily cured by a small change to the program.

Japanese, Chinese, Korean and Thai webpages present it with the much bigger problem that these languages are normally written without any space between "words". However, we have newly built into it an interface with the **Japanese** word segmenter **Mecab**[31]. This seems to be successful, and gives the spinoff benefit that hovering over a Japanese word now displays its pronunciation in Hiragana. Japanese learners have such a hard task to face with unknown Kanji that even partial success could be of tremendous benefit. For **Chinese**, we managed to do the same with the **Urheen**[32] word segmenter and the results seem to be good, but at the time of writing this is performing far too slowly to be useful and has been switched off. The bother seems to be that Urheen does a lot of inefficient initialization every time it is called, but we might manage to find ways round this.

3.4 The "lemmatization" facility in Multidict

Although this belongs to Multidict as regards programming, it is described here because it is when Multidict is used together with Wordlink that all sorts of inflected wordforms are thrown at it. We put "lemmatization" in inverted commas, because the facility is only semi-trying to produce grammatical lemmas. Because it is only going to present the user with a string of possibilities, it does not need to go for grammatical purity and "headword suggestions" might be a better term than lemmas.

The basis of this facility in Multidict for most source languages is the **Hunspell**[33] spellchecker, which is the opensource spellchecker used by LibreOffice, OpenOffice, Firefox, etc. Old-fashioned spellcheckers just had a long list of wordforms in a .dic file. Hunspell, on the other hand, was originally developed for Hungarian which is a highly inflected language and works in a much more intelligent way using also a .aff file (aff<affix). The words in the .dic file can be labelled for grammatical category, and the .aff file contains the rules to produce a range of inflected wordforms relevant to that grammatical category. The great thing is that we do not need to attempt to understand or reverse engineer these rules. Hunspell itself has built into it a function to return the possible lemmas corresponding to any given wordform. All we need to do is to pull in from the Internet the Hunspell .dic and .aff files for lots of languages, and this we have done.

How successful Hunspell is at lemmatizing depends on the language and how Hunspell has been implemented for it. It is possible for an implementer to just throw lots of wordforms into the .dic file and put very few rules in the .aff file. Hunspell lemmatizes Basque very well, for example, but the current implementation does very little for German. For Scottish Gaelic it was not great and for Irish not much better, and so we turned to another solution, the use of a **lemmatization table**.

We were very fortunate and very grateful to be donated huge lemmatization tables for both Scottish Gaelic and Irish. And a huge public domain table for Italian, Morph-it[34] (Zanchetta and Baroni, 2005), was found on the Internet. Smaller batches added to this include the Old Irish verbforms from In Dúil Bélrai[35]; tables from the Internet converting between en-US and en-GB English spelling; and tables converting between pre-Caighdeán and post-Caighdeán Irish spelling. These form the basis of an alternative method of lemmatization which Multidict has at its disposal, namely the `lemmas` table in the Multidict database which currently has 1.4 million wordforms. These can be labelled with the "batch"

[31] http://mecab.googlecode.com

[32] http://www.openpr.org.cn/index.php/NLP-Toolkit-For-Natural-Language-Processing/68-Urheen-A-Chinese/English-Lexical-Analysis-Toolkit/View-details.html

[33] http://hunspell.sourceforge.net

[34] http://sslmitdev-online.sslmit.unibo.it/linguistics/morph-it.php

[35] http://www.smo.uhi.ac.uk/sengoidelc/duil-belrai/

field, which can be used for example to denote those to be given priority, or those to be applied only for certain dictionaries.

Algorithmic "lemmatization" provides yet another tool in Multidict's lemmatization armoury. Again this is divided into a "priority" algorithm to be used first, and a non-priority algorithm. The priority algorithm includes the removal of initial mutations from Irish and Scottish Gaelic words, because this is nearly always something sensible to do. The non-priority algorithm includes throwing out any final 's' from English words, because this is normally a last resort when the word has not been recognized by Hunspell. The non-priority algorithm includes crude attempts to lemmatize words in the p-celtic languages, Welsh, Cornish and Breton, by naively changing the initial letter.

It turns out to be rather crucial, especially for Irish and Scottish Gaelic, to have priority records in the the `lemmas` table for the lemmatization of **irregular** verbs, otherwise many of them would not be recognised after initial mutation was removed. This has been done, and all the prepositional pronouns have been added too. This is something we really ought to do for every language: namely feed into the lemmatization table all the irregular verbs, irregular nouns, etc, because Hunspell deals with these rather poorly. Hunspell's priorities and ours are different. Its priority is to reduce the size of the .dic file by placing rules for regular verbs and nouns in the .aff file. Irregular verbforms take up relatively little space in the .dic file, so it just throws them in there and doesn't help us at all to lemmatize them. Multidict now has in place a very sophisticated, flexible mechanism for lemmatization, pulling in as required the different tools at its disposal. It would be good if experts for individual languages could co-operate to help implement and tailor these tools for each particular language.

The default "wfrule" string which Multidict uses to generate headword suggestions for a particular wordform is "`lemtable~pri|prialg|self|lemtable|hun|lemalg`". What this means in plain English is: concatenate the lists of headword suggestions produced by (1) those labelled "pri" in the `lemmas` table, (2) those produced by the priority algorithm, (3) the wordform itself, (4) those with no batch label in `lemmas`, (5) those provided by Hunspell, and (6) those produced by the non-priority algorithm. The | operator not only concatenates but causes duplicates to be removed from the list. However, different "wfrule" strings can be applied for different languages and dictionaries. As well as the | operator, there is another operator > which causes the array of suggestions generated by the previous rule to be used as input to a following rule. And brackets ( ) can also be used in this "algebra".

3.5 Beware of robots

In any publicly available facility such as Wordlink which can take any webpage and process it to produce another, it is essential to be very careful about `robots.txt`[36] and robots meta tags in the html header. At one point the server hosting multidict.net was running very slowly and on investigation it was found that Google was attempting to spider and index the entire Internet via Wordlink! The links on one Wordlinked webpage were leading it to other Wordlinked webpages. It took months before it completely stopped.

4 Clilstore

Clilstore is the most recent of the three facilities. It makes it easy for teachers to harness the power of Wordlink and Multidict, by adding teaching "units" to the openly available online "store". The formula which has been found to be most successful has been a video or soundfile together with a transcript, and perhaps some exercises to test student understanding. Clilstore itself stores the text, and can store attachment files of limited size. But storing the video or soundfile is left to the very many media hosting services available on the Internet, such as Youtube, Vimeo, TED, Teachertube, Ipadio and Soundcloud, from where they can be very easily added to the Clilstore unit by using the embed code supplied by the hosting service. This avoids us getting into large storage requirements, and hives off any copyright questions to services with mechanisms in place to deal with infringements.

Each unit is labelled with a level, A1, A2, B1, B2, C1 or C2, from the Common European Framework of Reference for languages (CEFR[37]). The index provides a rich facility for searching by words

36 http://en.wikipedia.org/wiki/Robots_exclusion_standard
37 http://en.wikipedia.org/wiki/Common_European_Framework_of_Reference_for_Languages
http://www.coe.int/t/dg4/linguistic/Cadre1_en.asp

in the title or text, and for searching or ordering by language, CEFR, media length, number of words, number of views, etc. A wysiwyg editor, **TinyMCE**[38], provides a facility for authors to produce rich colourful units without getting involved in html, although an html editor is also available.

To date (2014-06-24), Clilstore has 1072 units (excluding test units) in 49 different languages. The biggest number (416) are in English, but there are 116 in Arabic, 101 in Scottish Gaelic, 65 in Slovenian, 51 in Irish, 40 in Portuguese, 38 in Spanish, 34 in Italian, 27 in Lithuanian, 26 in German, 22 in Danish. There is even one, complete with soundfile in Old Irish. Clilstore and Wordlink work fine with right-to-left languages such as Arabic, although good online dictionaries are still rather lacking for Arabic. Statistics show that the units have had so far over 203,000 views in total. Perhaps more interestingly and reliably, in the 3 months since we started collecting such statistics, there have been 6773 clicks (dictionary lookups) on words in Clilstore units.

Experience from workshops for Gaelic language summer courses[39] at various levels at Sabhal Mòr Ostaig shows that the Clilstore facility is most useful to intermediate level learners. Advanced users find it very useful too, as a store of videos and transcripts, but tend to click fairly seldom because they can understand well enough from context anyway. Learners coming from **cognate languages** with somewhat different spelling rules such as Irish learners of Scottish Gaelic find it particularly useful, as was seen on the summer courses on Scottish Gaelic for Irish speakers at Sabhal Mòr Ostaig.

5 Conclusion

The facilities described here work, have proved their worth[40], and are freely and openly available. Much more could be done to develop them, of course. The interface is entirely through English at present, which is not good when trying to provide an immersion environment for Gaelic students, for example. Nor is good for Italian students at a Portuguese university, to have to go through an English interface to access Portuguese units. It would be good to internationalize the programs and provide localized interfaces.

Multidict and Wordlink use old-fashioned html frames[41], which have no support in modern standards[42], although they work well for the job in hand. It would be good to investigate switching to iframes[43], although this would require increasing use of Javascript libraries for resizing.

Users can and do recommend new dictionaries for Multidict, but it would be good to develop this into more of a community facility.

Acknowledgements

Multidict and Wordlink were first developed under the EC financed[44] POOLS-T[45] project. Clilstore was developed, and Multidict and Wordlink further developed under TOOLS[46] project financed by the EC's Lifelong Learning Programme. Much of the credit for their development goes to the suggestions, user testing and feedback by the project teams from 9 different European countries, and in particular to the project leader Kent Andersen. Wordlink was inspired by Kent's Textblender program. We are grateful to Kevin Scannell for the Irish lemmatization table used by Multidict, and to Mìcheal Bauer and Will Robertson for the Scottish Gaelic lemmatization table.

References

Eros Zanchetta and Marco Baroni.. 2005. *Morph-it! A free corpus-based morphological resource for the Italian language*, proceedings of Corpus Linguistics 2005, University of Birmingham, Birmingham, UK

[38] http://www.tinymce.com
[39] http://www.smo.uhi.ac.uk/gd/cursaichean/cursaichean-goirid
[40] There are now 1072 Clilstore units, and new are created almost daily both by people inside the project and people completely unconnected with it. Wordlink has clocked up over 315,000 dictionary lookups in the past six years.
[41] http://www.w3.org/TR/html401/present/frames.html
[42] http://www.w3.org/TR/html5/obsolete.html
[43] http://www.w3.org/TR/html5/embedded-content-0.html#the-iframe-element
[44] Standard disclaimer applies: This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein
[45] http://languages.dk/pools-t
[46] http://languages.dk/tools