A lexical database for Irish based on existing lexicographical resources

Aonghus Ó hAirt
B.A. (Mod.) Computer Science, Linguistics and Irish
Final Year Project, April 2011
Supervisor: Dr. Carl Vogel
Declaration

I hereby declare that this thesis is entirely my own work and that it has not been submitted as an exercise for a degree at any other university.

____________________________
Aonghus Ó hAirt
April 11, 2011
Permission to Lend

I agree that the Library and other agents of the College may lend or copy this thesis upon request.

__________________________________________
Aonghus Ó hAirt
April 11, 2011
Acknowledgements

I would like to thank my supervisor, Dr Carl Vogel, for his help and guidance with this project, and for all his support throughout CSLL.

I wish to acknowledge Foras na Gaeilge, particularly Cathal Convery, for access to the dictionary material and permission to use it here, and for feedback and support for the project. I would also like to thank Elaine Úi Dhonnchadha for her advice on various aspects of this project.

I am grateful to all my colleagues in CSLL, past and present, who have made it a most enjoyable experience.

Finally, I wish to thank my family for all of their encouragement and support.
Contents

1 Introduction ................................................. 9

2 Implementation ............................................. 13

  2.1 Typesetting files ....................................... 14
  2.2 HTML version ........................................... 17
  2.3 Latex version ............................................ 21
  2.4 XML version ............................................. 24
  2.5 Database .................................................. 28
    2.5.1 Structure .......................................... 30
    2.5.2 Population .......................................... 34
  2.6 Web interface ........................................... 35
  2.7 Thesaurus generation .................................... 35

3 Evaluation ................................................ 38

  3.1 HTML version ........................................... 39
  3.2 Latex version ............................................ 39
4 Conclusion

4.1 Summary ................................................. 43

4.2 Future work ............................................. 43
List of Figures

1.1 FGB entry ‘aigne’ in the program WinGléacht .................. 11
2.1 Example of typesetting versions of entries ‘admhaigh’, ‘aosaigh’, ‘ardchéim’, ‘athbhuaile’ .................................................. 15
2.2 Typesetting version of the entry ‘acant’ .......................... 16
2.3 Example dictionary entries with formatting applied .......... 17
2.4 Description of typsetting file markup characters ................ 18
2.5 Examples of transformations used to convert typsetting files to HTML (first pass) .................................................. 19
2.6 Example of HTML versions of entries rendered in Chrome browser .......................................................... 21
2.7 \LaTeX version of entry ‘aigne’ (left), compared to scanned version from the printed version of FGB (right) .............................. 23
2.8 DTD for XML version of the dictionary ......................... 29
2.9 Entry ‘aigne’ displayed in the web interface ....................... 36
2.10 Example entry in the thesaurus ..................................... 37
List of Tables

2.1 Transformations applied to typesetting files to produce HTML . . . 22
Listings

2.1 Example of HTML version of the entry ‘admhaigh’ . . . . . . . . 20
2.2 Example of XML version of the entry ‘aigne’ . . . . . . . . . . 25
2.3 CREATE statement for ‘entries’ table in SQL . . . . . . . . . . 30
2.4 CREATE statement for ‘hwd-translations’ table in SQL . . . . . 31
2.5 CREATE statement for ‘sentences’ table in SQL . . . . . . . . . 32
2.6 CREATE statement for ‘xrefs’ table in SQL . . . . . . . . . . . 33
Chapter 1

Introduction
The aim of this project is to create a lexical database for Irish. Although a number of related and similar projects have made some progress towards this goal, it remains the case that such a database has not been completed with comprehensive coverage of the language and containing the wealth of information that exists in such printed dictionaries as the Foclóir Gaeilge-Béarla (Ó Dónaill, 1977) (Irish-English Dictionary) or the English-Irish Dictionary (de Bhaldraithe, 1959). This project aims to help fill some of the gaps that remain in the area of Irish-language computational lexicography and it is hoped that in doing so, it will facilitate further development of Irish language tools and resources.

Specifically, the project is the creation of a relational database containing the contents of the Foclóir Gaeilge-Béarla (Ó Dónaill, 1977) (referred to as ‘FGB’ in this report), along with associated tools and applications. The content of the database is based on the data from a number of typesetting files which were used to print the dictionary when it was first published in 1977. The typsetting files are first converted into HTML to facilitate easier parsing. The HTML is in turn converted into XML files in which the content is marked up in great detail. Finally the XML files are used to populate the database. In addition database construction, a Latex version of the dictionary is created based on the HTML format, a web interface is developed to allow access to the database, and a thesaurus is ‘bootstrapped’ from the dictionary data.

There are two general-purpose Irish dictionaries in widescale use today; the Foclóir Gaeilge-Béarla (Ó Dónaill, 1977) and the English-Irish Dictionary (‘EID’) (de Bhaldraithe, 1959). Previous to that, the Dineen Foclóir Gaedhilge agus Béarla was popular (Dineen, 1904). FGB is the only comprehensive dictionary which follows the ‘Caighdeán Oifigiúil’ (official standard for spelling and grammar) (Rannóg an Aistriúcháin, 1979). A number of smaller-scale general-purpose dictionaries have been published more recently such as the Foclóir Póca (Foras na Gaeilge, 2007), but FGB and EID continue to be the authoritative dictionaries. Work is continuing on a new English-Irish dictionary which is expected to be published in 2012 (Foras na Gaeilge, 2011). In addition to these dictionaries, a number of subject-specific dictionaries have been published, and a comprehensive terminology database is maintained by the Terminology Committee and hosted by DCU Fiontar (Fiontar,
There is no comprehensive online Irish dictionary available at present. The focal.ie terminology database is an excellent resource, and the most up-to-date, but it is specifically for terminology and does not cover ‘everyday’ Irish. The WinGléacht CD-ROM is an electronic version of FGB, but without most of the example sentences in the dictionary. A key advantage of WinGléacht is that it provides users with the various inflected forms of headwords. Despite these resources, the need remains for a database which would contain the entire contents of a comprehensive general-purpose dictionary.

Figure 1.1 FGB entry ‘aigne’ in the program WinGléacht

A number of projects have been completed which share some of the goals of this project. Magee (2000) aimed to produce a LaTeX version of FGB from scanned versions of the printed dictionary. The difficulty with using scanned dictionaries is
that OCR software is still not very accurate, particularly with printed dictionaries where the type tends to be quite small. Clear (2002) aimed to use the OCR methods developed in Magee (2000) to produce a database containing FGB. The database structure meant that entries weren’t ‘broken down’ to constituent parts, rather the entire text of each sense was stored in a single field. This approach would be suitable where the aim is to simply display entries, but would not be suitable for many other applications. Lunney (2008) aimed to produce a relational database containing the contents of FGB, but a notable exception from the revised database structure proposed is the absence of example sentences. In many cases, example sentences might not be required, but the aim of this project is to produce a database which contains the full contents of the dictionary, including example sentences. In Herbert (2006) Irish synsets are mapped onto Princeton WordNet Synsets. This project makes use of the Foclóir Póca which is considerably smaller than FGB.
Chapter 2

Implementation
This chapter describes the implementation of the lexical database. The content of the database is all derived from 24 typesetting files which contain the Ó Dónaill Irish-English Dictionary (FGB). Section 2.1 describes these files and the mark up system used in them. Section 2.2 describes the methods by which these typesetting files were parsed and a HTML version of the dictionary produced. The HTML version is a useful format on which to work and subsequent parsing is based on these files. While not strictly contributing to the construction of the database, it was decided that a LaTeX version of the dictionary would be a useful resource to create and is discussed in Section 2.3. Producing the database requires that all of the elements of the dictionary be properly recognised, e.g. what text is grammatical information, example sentences etc. Rather than attempt to do this as the database is being populated, an XML version of the dictionary is produced as an intermediate phase. The XML version is discussed in Section 2.4. Once the XML files are generated, these are used to populate the database which is discussed in Section 2.5. A simple web interface is discussed in Section 2.6 which allows users to query the database and view dictionary entries in a web browser. Finally, the generation of a basic thesaurus from the contents of the database is described in Section 2.7.

2.1 Typesetting files

The FGB dictionary was published in 1977 and was printed using a “new computer process” (Ó Dónaill, 1977, p. vi). This involved the use of typesetting files which contained the contents of the dictionary marked up with formatting information. These typesetting files were supplied by Foras na Gaeilge for use in this project. The aim of this section is to describe these typesetting files and provide details of the markup system adopted in them.

A total of 24 files were supplied - one for each letter of the alphabet (with the exception of ‘K’ and ‘W’ which are not typically used in the Irish language). Each file contains the entries for the headwords beginning with its respective letter in the typesetting format. It is from these files that the dictionary contents are extracted.
Figure 2.1 Example of typesetting versions of entries ‘admhaigh’, ‘aosaigh’, ‘ardchéim’, ‘athbhuaile’


and used to produce HTML, \LaTeX, XML, and database versions of the dictionary discussed later on.

An example of the typsetting information for the entries ‘admhaigh’ and ‘aosaigh’, ‘ardchéim’, and ‘athbhuaile’ is given in Figure 2.1. At first glance, the entries are almost incomprehensible, but using the legend given in Figure 2.4 (also supplied by Foras na Gaeilge), it is possible to recognise the content of the dictionary entries.

All of the characters in the text file are in uppercase, but should be assumed to be lower case in the printed version of the dictionary, unless the ‘[’ character is used to indicate that letter following it is uppercase. No accented characters are used in the typesetting files, although vowels in Irish can be accented (e.g. ‘á’); the ‘$’ character is used to indicate that the following letter should be accented.

‘*G’ indicates that the following text is grammatical information and should be displayed in italics, e.g. ‘GV.T. & I.’ (printed v.t. & i.) in the example below indicates that the headword ‘admhaigh’ is a transitive or intransitive verb. ‘B’
indicates that the following text should be displayed in bold face and that the text is Irish; this is used mostly (but not exclusively) for the Irish versions of example sentences, e.g. “Bean nach n-admhódh a fear féin” in the ‘admhaigh’ example. *I* indicates that text should be displayed in italics; this is used for some Irish example sentences (it is not clear why some are shown in bold and some or shown in Italics), e.g. “Admhaín don saol!”.*E* is used for italics where the text is in English; this is used (occasionally) for usage hints such as “used in certain phrases” in the ‘aosaigh’ example. *R* indicates that the following text should be displayed in regular roman characters; most of the English text in the dictionary is displayed like this, including the English translations of example sentences.

In addition to these markup symbols, which are primarily concerned with formatting, a number of other symbols are used which, in addition to formatting, provide useful information about the content of the text. *C* indicates that the text is ‘classification’, i.e. a label indicating that a particular sense is subject-specific such as ‘Sch’ for ‘schools’ in the ‘ardchéim’ example. *O* indicates ‘of’ hints, e.g. “of juvenile” in the ‘aosaigh’ example.

**Figure 2.2** Typesetting version of the entry ‘acant’

\[
\text{ACANT(*HA)(*HI*R)*H-, *GPREF. *R[ACANT(H(O)-} \]

According to the legend, *P is used to mark where the following character is a period. It is unclear, however, what significance this marker has since most period characters are used without *P. Similarly, it is indicated in the legend that *H is used for tildes and hyphens, yet tildes and hyphens are used in many places in the files without the *H marker. In fact, the *H marker is used in some places where there are no hyphens or tildes, e.g. in the entry for the headword ‘acant’ (Figure 2.2) the headword is marked up with two *H markers, ‘ACANT(*HA)(*HI*R)’, but the correct presentation of the headword is \text{acant(a)(i)}, with no hyphens. In any case, it seems that the *P and *H markers can simply be ignored.

*S indicates that a particular piece of text is a cross reference to another headword. *X and *Y indicate the beginning and end of superscript text respectively, which is used to number headwords where multiple entries have the same headword form.
CHAPTER 2. IMPLEMENTATION

Figure 2.3 Entries ‘admhaigh’, ‘aosaigh’, ‘ardchéim’, and ‘athbhuaile’ with formatting applied (these formatted entries were produced in \TeX using the methods described in Section 2.3)

| admhaigh, v.t. & t. (un. -mháil). 1. Acknowledge. (a) Admit. D’- sé go raibh an ceart agam, he admitted that I was right. Admhainn an saol (go), everybody admits, agrees (that). ~ amach é, admit it openly. (b) Own to be. Bean nach n-admhódh a fear féin, a woman who would not acknowledge her own husband. (c) Acknowledge receipt of. Nior ~ tá mo litir, you did not acknowledge my letter. 2. Confess. Adhmáin mo locht, mo pheacai, I acknowledge my fault, my sins. Adhmáin do Dhía, I confess to God. Adhmáin mo chreideamh, I confess, profess, my faith. Adhmáin don saol! I declare to goodness! |
| aosaigh, v.t. (un. -sú m. gs. -ithe; var. pres. aut. aostar used in certain phrases). Age: come of age. Cleithiúnaithe aosaithe, adult dependants. D’- an oiche, the night wore on. Má aostar dó, leis, (of juvenile) if he grows to manhood; (of adult) if he lives to old age. |

(e.g. ‘a’1, ‘a’2, ‘a’3 etc.). Where cross references refer to a specific sense or sub-sense within an entry, this is indicated by the numeral reference marker, *N, and the letter sub-reference marker, *L, respectively. The ‘athbhuaile’ entry has an example a cross-reference to sense 2 of the entry ‘cuairt’1, S.a. CUAIRT 2. *A indicates a ‘see also’ cross-reference, and *V indicates a reference to a variant, e.g. CUAIRT 2 in the ‘athbhuaile’ exam.

An image of how the entries ‘admhaigh’, ‘aosaigh’, ‘ardchéim’, and ‘athbhuaile’ might appear printed, according to the typsetting information in Figure 2.1 is given in Figure 2.3.

2.2 HTML version

In this section I will discuss the implementation of a program, TYPEgoHTML.pl, which translates the contents of the typsetting files, described in the previous section, into HTML files. There are a number of reasons for doing this. Firstly, the markup used in the typesetting files is primarily used to describe the format of the dictionary entries as printed in the published dictionary. However, this markup system is non-standard, and specific to this dictionary. It makes sense, therefore,
Figure 2.4 Description of typsetting file markup characters

[ = The next letter is capitalized (count = 89086).
$ = The next letter (vowel) has an acute accent ($A = a-fada) (count 76907).
* = Indent (count 291).
*A = See also (count = 1961).
*B = Irish bold type (count = 85635).
*C = Classification, e.g. "Gym." (Gymnastics) (count = 7368).
*D = Not used (count = 0).
*E = English in italics (count = 182).
*F = Not used (count = 0).
*G = Grammar (count = 51535).
*H = Hyphen or tilde (count = 701).
*I = Irish italics (count = 7993).
*J = Not used (count = 0).
*K = Colon (count = 3923).
*L = Letter sub-reference, e.g. "(a)" (count 6773).
*M = Not used (count = 0).
*N = Numeral reference, e.g. "1." (count = 2538).
*O = Of -- (count 1784).
*P = Period (count = 960).
*Q = Not used (count = 0).
*R = English in roman font (count = 102647).
*S = Referenced Irish word (count = 22649).
*T = Not used (count = 0).
*U = Not used (count = 0).
*V = Variant (count = 5516).
*W = Not used (count = 0).
*X = Superscript marker (18029).
*Y = Return from superscript (18029).
*Z = Not used (count = 0)
to translate it into HTML so that standard HTML parsers can be used to parse the
dictionary, rather than having to build a complete parser from scratch. Secondly,
given that the markup in the typesetting files is primarily concerned with the
formatting of the entries, producing a program to translate these files to HTML
should be relatively straightforward and would produce a format that is significanly
easier to work with than the typesetting files which are almost unreadable.

Figure 2.5 Examples of transformations used to convert typsetting files to HTML
(first pass)

```
identify_entry
------------------------------
ABAIR*X1*Y, *GV.T. & I. *R(*GPRES. *BDEIR, *GP. *BD$UIRT, *GFUT. *BD$EARFAIDH,
-> <p class="entry">
  <b class="head">ABAIR*X1*Y</b>, *GV.T. & I. *R(*GPRES. *BDEIR, *GP. *BD$UIRT,
  *GFUT. *BD$EARFAIDH,

switch_to_lower
------------------------------
<b class="head">ABAIR*X1*Y</b>/b>, *GV.T. & I. *R(*GPRES. *BDEIR, *GP. *BD$UIRT,
-> <b class="head">abair*x1*y</b>/b>, *gv.t. & i. *r(*gpres. *bdeir, *gp. *bd$uirt,

process_upper
------------------------------
<b class="head">acant(*ha)(*hi</b>*r)*h-, *gpref. *r[acanth(o)>
-> <b class="head">acant(*ha)(*hi</b>*r)*h-, *gpref. *rAcanth(o)>

process_accents
------------------------------
<b class="head">abair*x1*y</b>/b>, *gv.t. & i. *r(*gpres. *bdeir, *gp. *bd$uirt,
<b class="head">abair*x1*y</b>/b>, *gv.t. & i. *r(*gpres. *bdeir, *gp. *bd$uirt,

process_hyphen
------------------------------
<b class="head">acant(*ha)(*hi</b>*r)*h-, *gpref. *rAcanth(o)>
<b class="head">acant(a)(i</b>*r)\- , *gpref. *rAcanth(o)>
```

The program is a two-pass process. The first pass identifies entries and their
headwords, and splits up the file accordingly. It makes basic changes to the text,
such as correcting the case of characters, and adding accents where appropriate.
Examples of the transformations made during the first pass are given in Figure
2.5.

The second pass is where most of the formatting changes are applied. For most
of the formatting changes, markup characters in the typesetting files are replaced
with HTML tags which apply the same formatting styyles. For the mostpart, this
is relatively straightforward; e.g. *I indicates that the following text should be
displayed in italics, so simply replacing all instances of *I with the HTML tag ⟨i⟩
will apply this formatting change in the HTML version.

While the typesetting files indicate where any particular formatting style should
begin (and hence, where a HTML opening tag should be inserted), there is no
explicit indication as to where that formatting style should end. So while each
markup character in the typesetting file has a corresponding HTML opening tag,
there is no markup character to correspond with the closing HTML tags. For
example, the HTML tag ⟨i⟩ can simply be inserted where I appears, but it is not
immediately obvious where ⟨/i⟩ should go. This is relatively easy to resolve as
formatting styles do not overlap in the dictionary. When a new style is activated,
any currently active style is deactivated, e.g. when a piece of text in italics is
followed by a piece of text in bold, the text ceases to be in italics when it begins
to be in bold. This means that the closing tags of any active html tags can be
inserted immediate before any new opening tags are inserted.

A number of the markup symbols in the typsetting files map onto the same HTML
but for different purposes, e.g. *A is for ‘see also’ cross references, *C is for
classification labels, *G is grammatical information etc. By simply using ⟨/i⟩ tags
for these in HTML, the information which distinguished them from each other
is lost. To avoid this loss of valuable information, class attributes are used in
the HTML to record how they were marked up in the original typsetting files.
*A produces an ⟨/i⟩ with class ‘see also’, * produces one with class ‘class’ (for
classification), *G produces one with class ‘gramitalics’. The use of these class
attributes ensures that no useful data is lost between the typsetting and HTML
versions.

Listing 2.1: Example of HTML version of the entry ‘admhaigh’

1 <p class="entry"><b class="head">admhaigh</b>, <i class="granitalics">v.t. &
2 ì. </i>(<i class="granitalics">vn. </i><b class="irbold">-mh´ ail</b>)<b
3 class="irbold">. 1. </b>Acknowledge. (<i class="letref">a</i>) Admit. <b
4 class="irbold">D’~ s´ e go raibh an ceart agam, </b>he admitted that I was
5 right. <b class="irbold">Admha´ ıonn an saol </b>(<b class="irbold">go</b>)<b
6 class="irbold">, </b>everybody admits, agrees (that). <b class="irbold">~"
CHAPTER 2. IMPLEMENTATION

7 amach é, </b>admit it openly. (<i class="letref">b</i>) Own to be. </b>
8 class="irbold">Bean nach n-admhódh a fear féin, </b>a woman who would not
9 acknowledge her own husband. (<i class="letref">c</i>) Acknowledge receipt
10 of. </b class="irbold">Níor - tú mo litir, </b>you did not acknowledge my
11 letter. </b class="irbold">2. </b>Confess. </b class="irbold">Admhaím mo locht,
12 mo pheacaí, </b>I acknowledge my fault, my sins. </b class="irbold">Admhaím do
13 Dhia, </b>I confess to God. </b class="irbold">Admhaím do chreideamh, </b>I
14 confess, profess, my faith. </b class="iritalics">Admhaím don saol! </i>
15 declare to goodness!</p>

Figure 2.6 Example of HTML versions of entries rendered in Chrome browser

A complete list of transformations completed by the TYPEgoHTML.pl program
is in Table 2.1. The HTML version of the entry ‘admhaigh’ is given in Listing 2.1,
and a screenshot of this entry rendered in the Chrome browser is given in Figure
2.6.

2.3 Latex version

The primary problem identified at the beginning of the project was that given
the age of the FGB dictionary, there is no comprehensive electronic version of the
dictionary available in any standard, usable, format. While this project aims to
solve this problem primarily by producing a database containing the contents of
the dictionary, there are a number of reasons why producing a PDF version of the
### Table 2.1: Transformations applied to typesetting files to produce HTML

<table>
<thead>
<tr>
<th>Legend</th>
<th>Pass</th>
<th>HTML transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[</td>
<td>First</td>
<td>Characters [a-z] replaced by [A-Z] where appropriate</td>
</tr>
<tr>
<td>$</td>
<td>First</td>
<td>Characters [aouie] replaced by [áoúié] where appropriate</td>
</tr>
<tr>
<td>*A</td>
<td>Second</td>
<td>(i) tag inserted, with class ‘see also’, e.g.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(i class=&quot;see_also&quot;)S.a. ⟨/i⟩</td>
</tr>
<tr>
<td>*B</td>
<td>Second</td>
<td>(b) tag inserted, with class ‘irbold’, e.g.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(b class=&quot;irbold&quot;)A fada ⟨/b⟩</td>
</tr>
<tr>
<td>*C</td>
<td>Second</td>
<td>(i) tag inserted with class ‘class’, e.g. (i class=&quot;class&quot;)Gym⟨/i⟩</td>
</tr>
<tr>
<td>*D</td>
<td>Not used</td>
<td>N/A</td>
</tr>
<tr>
<td>*E</td>
<td>Second</td>
<td>(i) tag inserted, with class ‘enitalics’, e.g.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(i class=&quot;enitalics&quot;)Introductory⟨/i⟩</td>
</tr>
<tr>
<td>*F</td>
<td>Not used</td>
<td>N/A</td>
</tr>
<tr>
<td>*G</td>
<td>Second</td>
<td>(i) tag inserted with class ‘gramitalics’, e.g.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(i class=&quot;gramitalics&quot;)prep. ⟨/i⟩</td>
</tr>
<tr>
<td>*H</td>
<td>Hyphen or tilde First</td>
<td>*H removed, hyphen and tilde characters left as-is</td>
</tr>
<tr>
<td>*I</td>
<td>Irish italics Second</td>
<td>(i) tag inserted with class ‘iritalics’, e.g.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(i class=&quot;iritalics&quot;)Tar isteach, a Sheán⟨/i⟩</td>
</tr>
<tr>
<td>*J</td>
<td>Not used</td>
<td>N/A</td>
</tr>
<tr>
<td>*K</td>
<td>Colon</td>
<td>*K removed, colon characters left as-is</td>
</tr>
<tr>
<td>*L</td>
<td>Letter sub-reference, e.g. “(a)” Second</td>
<td>(i) tag inserted, with class ‘letref’, e.g.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(i class=&quot;letref&quot;)a⟨/i⟩</td>
</tr>
<tr>
<td>*M</td>
<td>Not used</td>
<td>N/A</td>
</tr>
<tr>
<td>*N</td>
<td>Numeral reference, e.g. “1.” Second</td>
<td>(i) tag inserted with class ‘numref’, e.g.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(i class=&quot;numref&quot;)2⟨/i⟩</td>
</tr>
<tr>
<td>*O</td>
<td>Of – Second</td>
<td>(i) tag inserted with class ‘of’, e.g.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(i class=&quot;of&quot;)Of person⟨/i⟩</td>
</tr>
<tr>
<td>*P</td>
<td>Period First</td>
<td>*P removed, period characters left as-is</td>
</tr>
<tr>
<td>*Q</td>
<td>Not used</td>
<td>N/A</td>
</tr>
<tr>
<td>*R</td>
<td>English in roman font Second</td>
<td>Close any open HTML formatting tags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(small) tag inserted, with class ‘s_ref’, e.g. ⟨small class=&quot;s_ref&quot;⟩aibíd.⟨/small⟩inserted1</td>
</tr>
<tr>
<td>*T</td>
<td>Not used</td>
<td>N/A</td>
</tr>
<tr>
<td>*U</td>
<td>Not used</td>
<td>N/A</td>
</tr>
<tr>
<td>*V</td>
<td>Variant Second</td>
<td>(i) tag inserted with class ‘var’, e.g. (i class=&quot;var&quot;)Var⟨/i⟩</td>
</tr>
<tr>
<td>*W</td>
<td>Not used</td>
<td>N/A</td>
</tr>
<tr>
<td>*X</td>
<td>Superscript marker Second</td>
<td>(sup) inserted</td>
</tr>
<tr>
<td>*Y</td>
<td>Return from superscript Second</td>
<td>⟨/sup⟩ inserted</td>
</tr>
<tr>
<td>*Z</td>
<td>Not used</td>
<td>N/A</td>
</tr>
</tbody>
</table>
dictionary in \LaTeX{} would be desirable. At present, the publishers of the dictionary, An Gúm, have no means of reprinting the dictionary apart from producing verbatim copies of the it based on the original printing process. A PDF version of the dictionary would give them the option of reprinting the dictionary with changes, e.g. the correction of errors or typos if they wished. Having a PDF version of the dictionary which closely resembles the printed version would also be convenient in the development of the database so that entries could be checked against the formatted ‘dictionary’ version without having to constantly consult the hard-copy of the dictionary.

Figure 2.7 \LaTeX{} version of entry ‘aigne’ (left), compared to scanned version from the printed version of FGB (right)

\begin{verbatim}
aigne, f. (gs. –). 1. Lit: Nature, character. 2. Mind, disposition. ~ an duine, man’s mind, way of thinking. Ar aon ~, of one mind. Coimead an mèid seo i d’~, bear this in mind. Tá fios a ~ aigne, he knows his own mind. Ní hé an rud a deir sé atá ina ~ aigne, he is not speaking his mind. Ná lig as d’~ é, don’t let it out of your mind. Cuir d’~ leis, set your mind to it. Níl m’~ leis an obair seo, I don’t care for this work. Ní bhfaighinn i m’~ é a dhéanamh, I couldn’t bring myself to do it. Tá meardadh ar m’~, my mind is distracted. Tá sé ar m’~ o mhaidir, it has been on my mind all day. In aghaidh m’fhoinn agus m’~, against my wish and inclination. 3. Spirit, cheerfulness. Chuir an bia ~ an, the food cheered him. Thadhm ~ air, he became cheerful. Faoi ~, lively, cheerful. Tá sí lán d’~, she is full of life, high-spirited. Tá a chroi agus a ~ aigne, he is in good heart and spirit. Tá ~ baintri aici, she has cast off her widow’s weeds. 4. Intention. Tá sé ar ~ agam labhairt leis, I intend to speak to him. An rud atá i m’~ chugat, my intentions towards you. 5. Fig. Strong spirit, stomach. Níor mhór duí ~ laidir a bheith agat chun feachaint air, one would need a strong stomach to look at it. Chasadh sé d’~, it would nauseate you. (Var: m)
\end{verbatim}

A program, HTMLgoLatex.pl, was written which parses the HTML files produced by the process discussed in Section 2.2. The program uses the HTML::Parser Perl module, which processes the files and calls event handlers ‘start’, ‘text’, and ‘end’ when it encounters opening tags, plain text, and closing tags respectively (Aas and Chase, 2011). It operates in a similar fashion to the SAX XML parser for Java
The program, on encountering opening tags \textbf{\langle b \rangle}, \textsuperscript{\langle sup \rangle}, \textit{\langle i \rangle}, and \texttt{\langle small \rangle} outputs ‘\textbf{t}', ‘\textsuperscript{\tiny t}', ‘\textit{t}', and ‘\texttt{t}' respectively. When it encounters text, special \LaTeX{} characters are escaped and then the text is output. On encountering closing tags, the open braces ‘’ are closed. The output files are then included in a \LaTeX{} template where font styles, page dimensions, column widths etc. are set. Efforts were made to match the font and sizes as closely as possible to the printed version of the dictionary. Figure 2.7 shows the \LaTeX{} version of the entry ‘aigue’ compared a scanned image of the entry from the printed dictionary. They are quite closely matched, but not precisely the same. For the purposes of this project, the version generated by HTMLgoLatex.pl is adequate.

The PDF file of the dictionary produced by this process is 1,305 pages long and 8.46MB in size.

## 2.4 XML version

It was decided that, in order to transfer the data from the HTML versions of the dictionary (discussed in Section 2.2), it would be convenient to first create an XML version of the dictionary. XML is a useful format for storing data and transferring it between programs, formats etc. The creation of an XML version of the dictionary is not the primary aim of this project; to make use of the dictionary in XML format would be slow, inefficient, and often platform-dependent. The aim remains to produce a relational database to store the data from the dictionary. The rationale behind creating an XML version, as a step towards the creation of the database, is that parsing the dictionary is a process of identifying and categorising various types of information in the dictionary and then storing it properly in the dictionary. It was found that the process of categorising the data in the dictionary was, in many ways, similar to the production of XML where text is categorised according to the elements within which it appears. Having an XML
version of the dictionary would be useful if, in future work, the dictionary were to be used for purposes other than the creation of a database described in this project. Additionally, the process by which the XML version of the dictionary is imported into the database could be reused to import other dictionaries; the task to convert other resources into the XML format would be less onerous than building a new parser to populate the database from scratch.

Listing 2.2: Example of XML version of the entry ‘aigne’

```xml
1  <entry>
2    <form><orth>aig`e</orth></form>, <gramgrp><gen>f.</gen></gramgrp> <gram>(<gramgrp>gs.</gramgrp> <cit lang="ga">`aigne</cit></gramgrp>).</n>
3  <sense n="1"><domain name="Lit">Lit</domain> <hwdtrans><trans>Nature</trans>, <trans>character</trans></hwdtrans>. <sentence><cit lang="ga">~ an duine</cit>, <cit lang="en">man’s mind</cit></sentence> bear this in mind.</n>
4  <sense n="2"> <hwdtrans><trans>Mind</trans>, <trans>disposition</trans></hwdtrans>. <sentence><cit lang="ga">Ar aon ~</cit>, <cit lang="en">one mind</cit></sentence> don’t let it out of your mind.</n>
5  <sense n="3"> <hwdtrans><trans>Spirit</trans>, <trans>cheerfulness</trans></hwdtrans>. <sentence><cit lang="ga">Chuir an bia ~ ann</cit>, <cit lang="en">cheered him</cit></sentence> against my wish and inclination.</n>
```

CHAPTER 2. IMPLEMENTATION

25
CHAPTER 2. IMPLEMENTATION

A program, HTMLgoXML.pl, was developed to parse the contents of the HTML files produced by the process described in Section 2.2 and generate XML files. An example entry is given in Listing 2.2, and the Document Type Definition for the XML files is given in Figure 2.8. The naming of the XML elements was influenced by work being undertaken separately by Foras na Gaeilge on the English-Irish Dictionary, and a desire to keep the two similar.

The program uses the HTML::Parser which was also used in the HTMLgoLa-tex.pl program described in Section 2.3. As with the HTMLgoLatex.pl program, HTMLgoXML.pl works by parsing each HTML file and calling subroutines to handle starting tags, text, and closing tags. When starting tags are encountered, they are pushed onto a stack, and popped off when closing tags are encountered. This means that when processing text, all of the tags which apply to that text are recorded on the stack. The class of each HTML tag is also recorded in a hash so that when processing text, the classes of the tags that apply to the text are also available.
CHAPTER 2. IMPLEMENTATION

Below is a description of the XML produced by the program, and the criteria used to identify the elements being processed:

- Entries are delimited in the HTML files by \textlangle p \textrangle tags, and so on encountering opening and closing \textlangle p \textrangle tags, the program produces opening and closing XML \textlangle entry \textrangle tags respectively.

- Text which appears within \textlangle b \textrangle tags with class ‘head’ in the HTML files is headword data; this is placed within \textlangle form \textrangle tags in the XML version. The headwords themselves are placed within \textlangle orth \textrangle tags in the XML version, and optional HTML \textlangle sup \textrangle tags are replaced with \textlangle lbl \textrangle tags.

- Many entries contain more than one sense which are numbered 1..n, and some senses contain sub-senses labeled a..z. Numbers and letters which are found in the text and increase incrementally are identified and considered to delimit senses. \textlangle sense \textrangle tags are used as sense containers in the XML files.

- Grammatical information, indicated in the HTML by \textlangle i \textrangle tags with class ‘gramitalics’, is placed within \textlangle gramGrp \textrangle tags which act as containers in the XML version. These \textlangle gramGrp \textrangle containers can contain \textlangle pos \textrangle part of speech, and \textlangle gen \textrangle gender tags.

- Individual English translations of an entry’s headword are placed in \textlangle trans \textrangle tags which are contained within \textlangle hwdtrans \textrangle tags. Identifying the English translations of headwords in the HTML files is slightly difficult as they have no special formatting or explicit indicators - they are displayed in the dictionary in plain text. Once all of the other text is classified, it is assumed that what is ’left over’ is the translations of headwords.

- Example sentences and phrases are contained within \textlangle sentence \textrangle tags. The Irish versions of the sentences are contained in \textlangle cit \textrangle tags, with the ‘xml:lang’ attribute set to ‘ga’, and English version with the value ‘en’. The Irish versions of the sentences are identified in the HTML files as text contained within \textlangle b \textrangle tags with ‘irbold’ class attributes, and \textlangle i \textrangle tags with ‘iritalics’ class attributes. Some other types of text can also appear within these
tags, but after other possibilities are eliminated, it is assumed that they are example sentences. The following, unformatted text, is taken to be the English versions of these sentences.

- Subject-specific labels are placed within \textlangle domain\textrangle tags with the attribute ‘name’ set to the relevant subject area (e.g. \textit{Sch} for School). These are identified in the HTML as text within \textlangle i\textrangle tags with the ‘class’ class attribute.

- Cross-references are contained in \textlangle xr\textrangle tags. These contain the headword of the the entry being referenced, and optionally, \textlangle sup\textrangle tags to identify specific entries where multiple entries have the same headword. The numbers within the \textlangle sup\textrangle tags correspond to the \textlangle lbl\textrangle tags contained within the \textlangle form\textrangle tags in the referenced entries. The cross-referenced headwords are identified in the HTML by the \textlangle small\textrangle HTML tag with ‘s_ref’ class attributes. Some cross-references also refer to specific senses (specified by a number) and sub-senses (specified by a letter) within entries. These sense numbers and sub-sense letters are included within \textlangle nr\textrangle and \textlangle letref\textrangle tags respectively.

- Usage hints included in the dictionary, e.g. “(Used in lit. phrases)” are recorded in the XML files withing \textlangle usg\textrangle tags. These are not explicitly marked in the typesetting files and are represented in plain text in the HTML files. As they are contained within parentheses, this is the criteria by which they are identified.

2.5 Database

This section describes the implementation of a relational database for storing the contents of the FGB dictionary. The XML version of the dictionary discussed in Section 2.4 is a useful resource and suitable for many applications. The primary aim of this project, however, is the production of a database which can be accessed, queried, updated and expanded in an efficient manner. For many of these criteria, XML falls sort. In addition, the database should be able to run on any standard
Figure 2.8 DTD for XML version of the dictionary

```xml
<!ELEMENT body ( entry+ ) >
<!ELEMENT cit ( #PCDATA | usg )* >
<!ATTLIST cit xml:lang ( en | ga ) #REQUIRED >
<!ELEMENT domain ( #PCDATA ) >
<!ATTLIST domain name CDATA #REQUIRED >
<!ELEMENT entry ( #PCDATA | cit | domain | form | gram | gramgrp | hwdtrans | letref | nr | sa | sense | sentence | sup | usg | var | xr )* >
<!ELEMENT form ( #PCDATA | lbl | orth )* >
<!ELEMENT gen ( #PCDATA ) >
<!ELEMENT gram ( #PCDATA | cit | gramgrp | usg | var )* >
<!ELEMENT gramgrp ( #PCDATA | gen | pos )* >
<!ELEMENT hwdtrans ( #PCDATA | trans )* >
<!ELEMENT lbl ( #PCDATA ) >
<!ELEMENT letref ( #PCDATA ) >
<!ELEMENT nr ( #PCDATA ) >
<!ELEMENT orth ( #PCDATA ) >
<!ELEMENT pos ( #PCDATA ) >
<!ELEMENT sa ( #PCDATA ) >
<!ELEMENT sense ( #PCDATA | cit | domain | gramgrp | hwdtrans | letref | nr | sa | sense | sentence | sup | usg | var | xr )* >
<!ATTLIST sense l ( a | b | c | d | e | f | g | h | i | j | k | l | m ) #IMPLIED >
<!ATTLIST sense n ( 0 | 1 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 ) #REQUIRED >
<!ELEMENT sentence ( cit+ ) >
<!ELEMENT sup ( #PCDATA ) >
<!ELEMENT text ( body ) >
<!ELEMENT trans ( #PCDATA ) >
<!ELEMENT usg ( #PCDATA | cit | gramgrp | sa | sup | xr )* >
<!ATTLIST usg type NMTOKEN #FIXED "hint" >
<!ELEMENT var ( #PCDATA ) >
<!ELEMENT xr ( #PCDATA | sup )* >
<!ATTLIST xr hwd CDATA #REQUIRED >
```
server without the need for specialist and/or commercial software. While some native-XML server-side database solutions are available, these are not widely used and are usually expensive commercial products. Another important consideration is that the database should represent the relations between entries in the dictionary. There is extensive use of cross-references in the dictionary (almost 26,000 in total), which are best represented in a relational database using semantic constraints.

For this project, an SQL database was designed and populated with the contents of the dictionary. The structure of the database is discussed in Section 2.5.1, and the process by which the database was populated is discussed in Section 2.5.2. The relational database management system (RDBMS) used is MySQL. MySQL was chosen due to its widespread availability, its easy installation, and reasonably good performance given that it is not particularly resource-intensive. The SQL statements used should work on any SQL database and it would be quite easy to transfer the database to another RDBMS.

### 2.5.1 Structure

The database consists of four tables: ‘entries’, ‘hwds-translations’, ‘sentences’, and ‘xrefs’. Each table is described below:

The ‘entries’ table has a row for each entry in the dictionary. Each entry has a unique id, ‘entry-id’ which is assigned by the DBMS. The headwords and their numerical labels (where relevant) are stored in the ‘hwd’ and ‘num’ fields respectively. The part of speech is stored in the ‘pos’ field, and for nouns, the gender is stored in the ‘gender’ field. For entries which contain grammatical hints at the top of the entry, these hints are stored in the ‘gram-hint’ field. ‘source’ indicates which dictionary the entry is from - at present, only FGB is used, but if additional entries were to be imported from other dictionaries, this field would distinguish between them. There are also ‘latex’, ‘html’, ‘text’, and ‘xml’ fields. These contain the full entry in the relevant format. The intention is that the contents of the database could be exported in any of those formats without it having to be generated from scratch each time.
CHAPTER 2. IMPLEMENTATION

Listing 2.3: CREATE statement for ‘entries’ table in SQL

```sql
-- Table structure for table 'entries'
CREATE TABLE IF NOT EXISTS 'entries' (  
'entry-id' int(10) NOT NULL AUTO_INCREMENT,  
'hwd' varchar(50) CHARACTER SET latin1 COLLATE latin1_general_cs NOT NULL,  
'num' int(3) DEFAULT NULL,  
'pos' varchar(20) NOT NULL,  
'gender' varchar(2) DEFAULT NULL,  
'gram-hint' varchar(50) DEFAULT NULL,  
'source' varchar(10) NOT NULL,  
'latex' text,  
'html' text,  
'xml' text NOT NULL,  
'text' text NOT NULL,  
PRIMARY KEY ('entry-id'),  
KEY 'hwd' ('hwd'),  
KEY 'num' ('num'),  
KEY 'pos' ('pos'),  
KEY 'gender' ('gender'),  
KEY 'source' ('source')  
);  
```

The ‘hwd-translations’ table contains the English translations of the Irish headword for each entry. Each translation has its own row, so where there are multiple translations of the same headword, there will be multiple rows in the ‘hwd-translations’ table. ‘hwd’ indicates which headword the translation is of, and ‘hwd’ and ‘hwd-num’ together specify the entry the translation belongs to. These fields are related to the ‘hwd’ and ‘num’ fields in the ‘entries’ table. For each ‘hwd’ and ‘hwd-num’ combination, there is one (and only one) row in the ‘entries’ table. For entries with multiple senses and sub-senses, the sense and sub-sense are indicated in the ‘sense-num’ and ‘sense-letter’ fields. ‘translation’ is the English translation of the Irish headword. As with ‘entries’, ‘source’ indicates the dictionary the entry is from, and only FGB is used at present. Each row in the table has a unique ID, ‘translation-id’ which is assigned by the DBMS.

Listing 2.4: CREATE statement for ‘hwd-translations’ table in SQL

```sql
-- Table structure for table 'hwd-translations'
CREATE TABLE IF NOT EXISTS 'hwd-translations' (  
'entry-id' int(10) NOT NULL AUTO_INCREMENT,  
'hwd' varchar(50) CHARACTER SET latin1 COLLATE latin1_general_cs NOT NULL,  
'num' int(3) DEFAULT NULL,  
'pos' varchar(20) NOT NULL,  
'gender' varchar(2) DEFAULT NULL,  
'gram-hint' varchar(50) DEFAULT NULL,  
'source' varchar(10) NOT NULL,  
'latex' text,  
'html' text,  
'xml' text NOT NULL,  
'text' text NOT NULL,  
PRIMARY KEY ('entry-id'),  
KEY 'hwd' ('hwd'),  
KEY 'num' ('num'),  
KEY 'pos' ('pos'),  
KEY 'gender' ('gender'),  
KEY 'source' ('source')  
);  
```
The ‘sentences’ table contains all of the example sentences included in the dictionary. Each example sentence has its own row in the ‘sentences’ table. The ‘hwd’, ‘hwd-num’, ‘sense-num’, ‘sense-letter’ fields indicate which entry the each sentence belongs to, and which sense and sub-sense (if any) the sentence is contained in. The Irish version of each example sentence is stored in the ‘sentence-irish’ field, and the corresponding English translation in the ‘sentence-english’ field. Each row in the table has a unique ID, ‘sentence-id’ which is assigned by the DBMS.

Listing 2.5: CREATE statement for ‘sentences’ table in SQL
Cross-references are contained in the ‘xrefs’ table. For each cross reference, ‘from-hwd’, ‘from-hwd-num’, ‘from-sense-num’, and ‘from-sense-letter’ indicate where the cross-reference is located (i.e. where it is pointing from), and ‘to-hwd’, ‘to-hwd-num’, ‘to-sense-num’, and ‘to-sense-letter’ indicate the referred location in the dictionary (i.e. where it is pointing to). Each cross reference has a particular type. ‘See also’ cross-references are given type ‘sa’. Cross-references following the ‘=’ character indicate that the source headword has the same meaning as the target headword, but that the target headword is the standard version. These cross-references have type ‘equals’. Cross-references following the ‘:’ character indicate that the source headword is an inflected form of the target headword. These have type ‘colon’. Cross-references to variations are given type ‘var’. Each cross-reference has its own unique ID, ‘xref-id’, which is assigned by the DBMS.

Listing 2.6: CREATE statement for ‘xrefs’ table in SQL
2.5.2 Population

The population of the database is a relatively straight forward operation given that the difficult task of correctly identifying and categorising the different types of text in the dictionary has already been dealt with in the production of the XML files discussed in Section 2.4. The XML files are parsed using a program called XMLgoSQL.pl and the relevant SQL statements required to populate the database are prepared and executed. The program uses the HTML::Parser in Perl which was used in some of the previously developed programs. While specific XML parsers exist, they operate in much the same way as HTML::Parser, calling subroutines on encountering opening and closing tags and text within the files. It was convenient, therefore, to continue using HTML::Parser and it proved perfectly adequate for the task of processing the XML files.

The program inserts a new row for each entry into the ‘entries’ table once it has seen the content within the ⟨form⟩ tags in the XML files. Using the ⟨orth⟩ and (optional) ⟨lbl⟩ data, it fills the ‘hwd’ and ‘num’ fields. The ‘pos’, ‘gender’, and ‘gram-hint’ fields are updated if and when the program encounters the relevant tags in the XML. As the files are being processed, the plain-text contents are collected for putting into the ‘text’ field which is updated after the closing ⟨entry⟩ tag is seen. The ‘xml’ and ‘html’ fields are also updated with the XML and HTML versions of the entries at this point. The populated ‘entries’ table contains 59,992 rows.

When the program encounters the ⟨sentence⟩ tags in the XML files, the English and Irish versions of the sentences within the ⟨cit⟩ tags are saved in variables and when the closing ⟨sentence⟩ tag has been seen, it inserts a new row into the ‘sentences’ table. The populated ‘sentences’ table contains 62,615 rows. Cross-references are extracted from the ⟨xr⟩ tags. New rows are inserted into the ‘xrefs’ table for every
chapter 2. implementation

Cross-reference encountered in the dictionary. The populated ‘xrefs’ table contains 25,963 rows. The program inserts new rows in the ‘hwd-translations’ table when it ⟨trans⟩ tags. The populated ‘hwd-translations’ table has 85,990 rows.

2.6 Web interface

This section describes the implementation of a simple web interface which allows users to query the database and display entries. The web interface is accessed through a Perl CGI script, ‘lexdb.pl’ which was produced for this project. It is hosted on a basic computer, but could be run on any web server which allows Perl/CGI as long as it can access the MySQL database. The interface allows users to enter their query into a HTML form which is processed by the program. The program generates the relevant SQL SELECT statements and returns the result to the browser. The interface allows users to search within headwords, example sentences, or anywhere within the entries. Cross-references are linked so users can navigate from entry to entry.

2.7 Thesaurus generation

This section describes the production of a simple thesaurus based on the contents of the database. Each headword in the dictionary which contains an English translation could be considered to be related to other headwords in the dictionary which share the same English translation. For example, the Irish verb ‘abair’, which means ‘to speak’ or ‘to express’, could be considered to be synonymous with the Irish verb ‘labhair’ which means ‘to talk’. As it happens, the two verbs both share the English translation ‘speak’ in their respective entries in the dictionary. By searching the database for Irish headwords which have common English translations, a simple thesaurus can be ‘bootstrapped’ from the database. The thesaurus can be further developed by searching for the headwords which are cross-referenced by the words in the thesaurus.
Figure 2.9 Entry ‘aigne’ displayed in the web interface

aigne, f (gr. -).  
1. Lit:Nature, character
2. Mind, disposition - an dhíon, man’s mind, way of thinking. An aon-~, of one mind. Coimhead an meid seo i d’~, bear this in mind. Ta fhios a ~aige, he knows his own mind. Ni bhfuil an rud a d’fheischéidh uait ma ~aige, he is not speaking his mind. Na leigheas d’~, don’t let it out of your mind. Cum d’~, let that be your mind to it. Ni m’fhéadraítear d’~, I don’t care for his work. Ni bhfuil aineamh i m’~, a thosaigh mé, I couldn’t bring myself to do it. Tá mé in ann ar an ~-~, my mind is distracted. Tá sé ar an ~-~, amhteas, it has been in my mind all day. Ta aghaidh in ann. agus an ~-~, against my will and inclination.
3. Spirit, cheerfulness Chuma an bh- ~ aon, the food cheered him. Thoir ~ aon, he became cheerful. Fas ~ aon, lively, cheerful. Ta an bhun d’~, she is full of life, high-spirited. Tá a chuid agus a ~ aige, he is in good heart and spirit. Tá buntóir aon, she has cast off her widow’s weeds.
4. Intention Tá sé ar ~ agam bheith leis, I intend to speak to him. An rud atá i m’~ chugat, my intentions towards you.
5. Fig: Strong spirit, stomach Níor maith du! ~ leis, a bheith agus chuam fheachtaí aon, one would need a strong stomach to look at it. Chaith féidir leis, it would nauseate you. (Var. -~)
A program was written which queries the database for a list of unique English translations appearing in the ‘hwd-translations’ table. For each English word which is returned (e.g. ‘speak’), the program then queries the database for a list of Irish headwords the translation applies to (e.g. ‘abair’, ‘labhair’). Then for each of these Irish headwords, the program queries the database for any entries which are cross-referenced and adds those headwords to the synset.

The program produces a text file as a result. On each line of the file, the English translation which the headwords have in common is listed, followed by the character ‘—’ and the list of Irish headwords delimited by commas. An example entry is shown below:

\begin{figure}[h]
\centering
\begin{verbatim}
speak|abair,abar,caintigh,cainteach,can,gabh,cead,compás,labhair,rónhair,rónhar,\end{verbatim}
\caption{Example entry in the thesaurus}
\end{figure}
Chapter 3

Evaluation
3.1 HTML version

The HTML version of the dictionary was produced early on in the project and is the format from which all of the other formats were derived. As such, any problems with the HTML version tended to have a ‘knock-on’ effect further on in the project. The program which generated the HTML version was amended any time such problems became apparent and has been revised numerous times throughout the project to ensure its accuracy. It is reasonably safe to assume, therefore, that the HTML version is quite accurate. In fact, by the end of the project there were no known issues with the HTML version which could be been the result of the program which generated it. There were however, a small number of issues with the typesetting files which could have affected the HTML version. The program which generates the HTML relies on the fact that the markers in the typesetting files distinguish between italic text which is grammatical information, italic text which is classification information, etc. It was found that occasionally these markers were mixed up. The errors would have had no effect on the printed dictionary since the text would be all in italics anyway, but would affect the HTML versions. Where these errors were noticed, they were fixed in the typsetting files, but their occurrence was rare.

3.2 Latex version

The Latex version of the dictionary was designed to be as similar to the printed dictionary as possible so that they could be easily compared. The Latex version of the dictionary was compared with the printed dictionary by taking sample pages from both and looking at them side-by-side. In early versions of the Latex version, differences were easily identified and problems were fixed. This was done on a fairly informal basis but it would appear now that the Latex version is very accurate and matches the printed dictionary very closely. Minor differences in font sizes and styles are unavoidable but don’t have any material effect on the accuracy of the file.
3.3 XML version

The place with the most potential for errors to arise is in the production of the XML files. It is at this stage in the process where most of the ‘decisions’ are made about what exactly is contained in any given entry. Many of the elements of the dictionary are not explicitly marked in the typsetting files and need to be inferred from the text itself. While every effort was made to ensure that the program producing the XML was accurate, a number of issues inevitably remain. Senses are numbered 1..n in entries, but these numbers are not specifically identified in the typsetting files, and so the program has to make educated guesses whether or not a numerical character in the text is indeed a numbered sense. It would appear that in the vast majority of cases its guess is accurate, but not always. Another known issue is that usage hints, which are contained within parentheses are sometimes characterised as English translations of headwords. It would appear that this is fairly rare, but happens occasionally nonetheless. The best way to measure the accuracy of the XML files would be to take one of them and have it carefully proofread and corrected. Comparing the corrected version to the original would allow the errors to be quantified and the accuracy measure. Unfortunately the proofreading would be quite labour-intensive and was not feasible here.

3.4 Database

Given that the transfer of the data from the XML files to the relational database is a reasonably straightforward one, it is unlikely that any new errors would arise at this stage in the process. At the time of writing there are no known systematic issues with the population of the database.
3.5 Web interface

The web interface is fairly basic at this stage. There are only two ‘views’ - that of an entry, and that of a list of results to a search query. The results page displays only the headwords of the resulting entries. Further developments could be made to make the interface more user-friendly and provide more informative search results. As far as errors are concerned, a number of minor formatting problems are apparent in the display of the entries. Whitespace is not always displayed correctly and in some instances characters are displayed with no space between them when there should be and in other cases there is too much whitespace. For a more thorough evaluation, the web interface could be made available to a number of users who would provide feedback, report errors etc. This was not feasible in the time-frame of this project.

3.6 Thesaurus

The most difficult aspect of the project to evaluate is the thesaurus discussed in Section 2.7. An attempt was made to compare the thesaurus to the Léonra Séimeantach na Gaeilge (LSG) (The Irish Language Semantic Network), produced by Scannell (2011), by creating a data structure similar to an grid or a two dimensional array with every word listed on both axes. The cells would contain 1s or 0s. A 1 would indicate that the word on the X axis occurs in the same synset as the word on the Y axis. One of these data structures could be created for the LSG and one for the thesaurus produced here. Then the two structures could be compared to see what overlaps occurs and where the two agree or disagree. In practice, the creation of such a datastructure would be heavily resource-intensive. The LSG lists over 330,000 words (many of them inflected forms), which would require the data structure to represent $330,000^2$ bits, which would be over 1 GB. Even if this number were reduced, there are over 54,000 unique headwords in the FGB dictionary, so the program would also have to represent $54,000^2$ bits and then have the resources to compare the two.
Chapter 4

Conclusion
4.1 Summary

This project aimed to produce a lexical database for Irish. This aim has been realised in the construction and population of a relational database containing the contents of the FGB dictionary. In the process of producing this database, a number of valuable resources were created; the HTML, XML, and Latex versions of the dictionary make the resource available in a range of useful formats suitable for many applications. The generation of a simple thesaurus based on the contents of the database provides an interesting start to the development of a thesaurus based on the FGB dictionary. The web interface, while simple, gives access to the dictionary in a way which is not available at present and which has many advantages over the current tool of choice, WinGléacht, including the presence of example sentences.

4.2 Future work

The project has scope for expansion and for further development. The example sentences are stored in simple text exactly as they appear in the printed dictionary. If these sentences were tagged using a POS tagger, it would allow for the sentences and entries to be queried and searched through on the basis of lemmas, tense, case etc. which would be useful given that Irish is a highly inflected language. The web interface is quite simple at present, but could be developed with more advanced search features and better search results pages. A feature of WinGléacht which is not included in the web interface is the ability to display the inflected forms of Irish headwords. Both of these developments could be done using existing tools described in Úi Dhonnchadha et al. (2003); Úi Dhonnchadha and Van Genabith (2006); Úi Dhonnchadha (2002). These developments would also facilitate the production of a parallel corpus based on the example sentences in the dictionary.
Bibliography


G. Herbert. A web user interface to an irish dictionary database. M.Phil. in Speech and Language Processing, October 2006.


