LEXIE - an Experiment in Lexical Information Extraction

John J. Camilleri
Dept. Intelligent Computer Systems
University of Malta
MSD2080 Msida, Malta
john@johnjcamilleri.com

Michael Rosner
Dept. Intelligent Computer Systems
University of Malta
MSD2080 Msida, Malta
mike.rosner@um.edu.mt

Abstract

This document investigates the possibility of extracting lexical information automatically from the pages of a printed dictionary of Maltese. An experiment was carried out on a small sample of dictionary entries using hand-crafted rules to parse the entries. Although the results obtained were quite promising, a major problem turned out to errors introduced by OCR and the inconsistent style adopted for writing dictionary entries.

Keywords

lexicon extraction, lexical information, lexicon, semitic

1 Introduction

This paper describes an experiment carried out within the general context of the Maltilex project [6], the objective of which is the development of a computational lexicon for the Maltese language. The compilation of such a body of lexical information for an entire language in machine readable form is a formidible task. Intuitively, we are aiming for a lexicon that comprises a set of entries under which all the information relating to a particular word is stored.

It is particularly difficult for Maltese because the Semitic component of the language is morphologically rich, so the set of valid, naturally occurring word forms is, in principle, larger than the set of lexical entries. Hence, certain issues about the role of morphological analysis and lexical organisation must be resolved if the set of entries is to be effectively delimited.

Fortunately, a lot of the of the work has already been carried out by the compiler of what still remains the most comprehensive printed Maltese-English dictionary, Joseph Aquilina [2]. The question, therefore, is whether the information, and to some extent, the organisation already present in that dictionary can be exploited.

The more general issue under investigation in the context of the present workshop is the re-use of expensively produced paper lexical resources by translation into a more useable electronic form.

The approach is not entirely new. Some work along these lines was carried out for the Longman Dictionary of English by Boguraev et al [3] and we owe much to the general approach adopted there. However, the two main differences we see between their approach and ours are that (a) Boguraev's work was oriented rather heavily towards the GPSG grammatical framework and (b) they were able to access the original source tapes of the dictionary. The input they used was therefore faithful to the orginal.

In our case we had to rely upon OCR input as described further in section 4. The inherent errors caused certain problems.

One of the aims of this work is to establish a viable method for extracting lexical entries for lesser-studied languages lacking the usual battery of language resources in machine readable form. Most languages of the world fall into this category. The availability of a paper dictionary, is, however, fairly widespread, even for exotic languages so that the methods being proposed could provide a robust alternative to dictionary extraction for a relatively large number of languages.

In terms of the types of information to be extracted from these pages, the aim of this experiment was to produce, with a reasonable level of accuracy, the following: (i) a correct list of headwords appearing in the dictionary, and (ii) associated lexical information for each headword in this list.

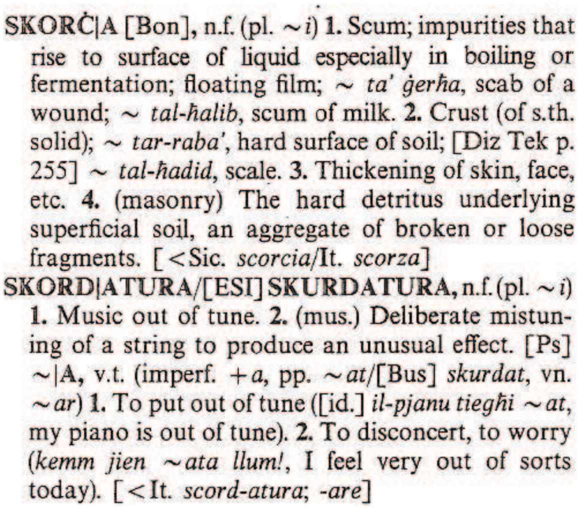
Additionally, to facilitate the interoperability of the extracted information with any other NLP applications which may make use of it, the format chosen for the final output lexicon was to conform to the evolving Lexical Markup Framework (LMF) [5] ISO standard.

This paper is structured as follows. Sections 2 and 3 respectively describe the formats of printed dictionary and LMF output. Section 4 describes the data used for the experiment. The main part the paper is in section 5 which explains the pipeline architecture used to process the input. The paper concludes with sections 6, 7, 8 and 9, describing results, limitations and future work.

2 Aquilina's Dictionary Entries

Figure 1 is a scan of part of a typical dictionary page. We can discern two main entries, *SKORĊ|A* and *SKORD|ATURA* where the vertical bar divides the stem on the left from an affix on the right. In subsequent parts of the entry, occurrences of tilde are replaced with the stem. The reader should also note the presence of an alternate spelling *SKURDATURA*

Workshop Adaptation of Language Resources and Technology to New Domains 2009 - Borovets, Bulgaria, pages 19–26



SKORĊ|A [Bon], n.f. (pl. ~*i*) **1.** Scum; impurities that rise to surface of liquid especially in boiling or fermentation; floating film; ~ *ta' ġerħa*, scab of a wound; ~ *tal-ħalib*, scum of milk. **2.** Crust (of s.th. solid); ~ *tar-raba'*, hard surface of soil; [Diz Tek p. 255] ~ *tal-ħadid*, scale. **3.** Thickening of skin, face, etc. **4.** (masonry) The hard detritus underlying superficial soil, an aggregate of broken or loose fragments. [<Sic. *scorcia*/It. *scorza*]

SKORD|ATURA/[ESI] SKURDATURA, n.f. (pl. ~*i*) **1.** Music out of tune. **2.** (mus.) Deliberate mistuning of a string to produce an unusual effect. [Ps] ~|**A**, v.t. (imperf. +*a*, pp. ~*at*/[Bus] *skurdat*, vn. ~*ar*) **1.** To put out of tune ([id.] *il-pjanu tiegħi* ~*at*, my piano is out of tune). **2.** To disconcert, to worry (*kemm jien* ~*ata llum!*, I feel very out of sorts today). [<It. *scord-atura*; *-are*]

Fig. 1: *Sample dictionary entries from Aquilina*

for the second entry. In this case it is clear that the two spellings are part of the same entry, but this is not always the case. The numbered subentries refer to alternate word senses. Items in square brackets give the etymology of the word. We are mostly interested in the morpho-syntactic information which is represented by various abbreviated forms. A full list of such abbreviations is supplied with the dictionary, and the most frequent ones are are shown in Table 1.

3 LMF

In order to fullfil out goal of creating a viable method for the extraction of lexical information, the output format has to be defined.

Paper lexicons contain quite a variety of information, as revealed from a rudimentary glance at the scanned entries of figure 1. Besides spelling and morpho-syntactic information, there is also definition, translation, and semantic field information. Aquilina's dictionary also contains references to other entries as shown by the entry for *skorfnott* in figure 2 which refers to the entry for *SKORFNA*.

Our choice of format has been heavily influenced by a desire to have the potential to capture all these different kinds of information and at the same time to remain theory independent.

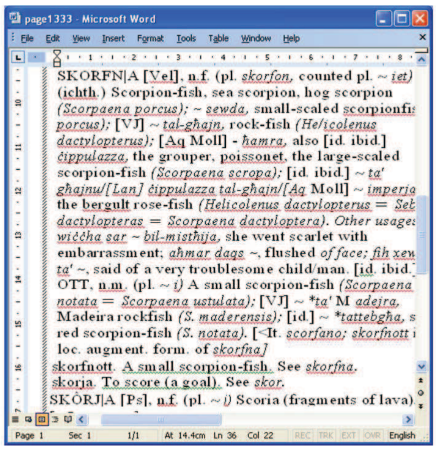
There are many potential formats that could be adopted. However, many of these are to some extent theory specific, or else overly biased towards particular kinds of information.

Lexical Markup Framework (LMF) is an evolving ISO[1] standard now in its 16th revision whose main goals are "to provide a common model for the creation and use of lexical resources, to manage the exchange of data between and among these resources, and to enable the merging of large number of individual electronic resources to form extensive global electronic resources" (see Francopoulo-et-al [4]).

[1] ISO-24613:2008

PARTS OF SPEECH	
a./adj	adjective
a.ag.	adjective of agent
adv.	adverb
ap.	active participle
conj.	conjunction
interj	interjection
n.	noun
n.ag.	noun of agent
nom.	nominal
n.u.	noun of unity
pp.	past participle
pron.	pronoun
pr.n.	proper noun
v.	verb
vn.	verbal noun
vn.u.	verbal noun of unity
TENSE/ASPECT/MOOD	
fut.	future
pres.	present (tense)
imperat.	imperative
imperf.	imperfect
perf.	perfect (tense)
pass.	passive
VERB TYPE	
intr.	intransitive
t./trans.	transitive
refl.	reflexive
GENDER/CASE	
c.	common
f./fem	feminine
m./masc.	masculine
obj.	object
voc.	vocative
NUMBER	
du.	dual
pl.	plural
s./sing.	singular

Table 1: *Common Abbreviations*



SKORFN|A [Vel], n.f. (pl. *skorfon*, counted pl. ~ *iet)* (ichth.) Scorpion-fish, sea scorpion, hog scorpion *(Scorpaena porcus)*; ~ *sewda*, small-scaled scorpionfi
porcus); [VJ] ~ *tal-għajn*, rock-fish *(Helicolenus dactylopterus)*; [Aq Moll] - *ħamra*, also [id. ibid.] *ċippulazza*, the grouper, poissonet, the large-scaled scorpion-fish *(Scorpaena scropa)*; [id. ibid.] ~ *ta' għajnu/[Lan] ċippulazza tal-għajn/[Aq* Moll] ~ *imperia* the bergult rose-fish *(Helicolenus dactylopterus = Seb dactylopteras = Scorpaena dactyloptera). Other usage: wiċċha sar ~ bil-misthija*, she went scarlet with embarrassment; *aħmar daqs* ~, flushed *of face; fih xew ta'* ~, said of a very troublesome child/man. [id. ibid.
OTT, n.m. (pl. ~ *i)* A small scorpion-fish *(Scorpaena notata = Scorpaena ustulata)*; [VJ] ~ **ta'* M *adejra*, Madeira rockfish *(S. maderensis)*; [id.] ~ **tattebgħa*, s red scorpion-fish *(S. notata)*. [<It. *scorfano; skorfnott* i loc. augment. form. of *skorfna]*
skorfnott. A small scorpion-fish. See *skorfna*.
skorja. To score (a goal). See *skor*.
SKÔRJ|A [Ps], n.f. (pl. ~ *i)* Scoria (fragments of lava).

Fig. 2: *References to other entries*

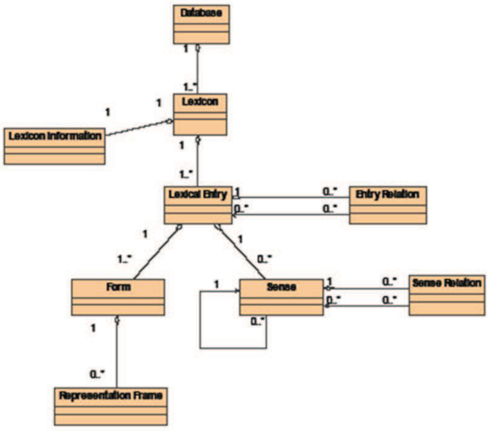


Fig. 3: *LMF framework [4]*

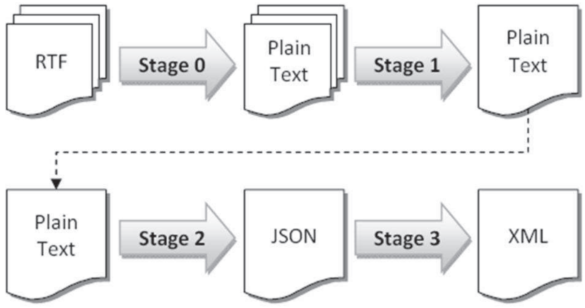


Fig. 4: *Processing Overview*

The model for lexical resources consists of a *core package*, which specifies how a lexical entry fits into the structure of a generalised lexical database, together with a set of extensions for handling various types of linguistic information such as morphology, syntax, semantics etc.

The diagram of figure 3 shows the organisation of the core package. Here we can see that a model is a Database which comprises one or more entities of type Lexicon, and a Lexicon comprises one or more entities of type Lexical Entry. The main interest is in the structure of the lexical entry.

4 Data

As part of a separate project being carried out at the University of Arizona on lexical perception by Ussishkin and Twist [1], the entire Aquilina dictionary has been scanned and run through Optical Character Recognition (OCR) software and saved as RTF files. Subsets of these files have been proof read and were kindly provided for experimental use in LEXIE.

The experiment reported here was carried out on two datasets. The first was used for development of the system and comprised pages 1333-1347 of the dictionary, containing approximately 360 entries. The second dataset, pages 1480-1490, was "unseen", and used for subsequent validation as described further in section 6

5 Processing

The different processing tasks required in this project were split into four distinct stages as shown in figure 4.

5.1 Stage 0 Pre-Processing: RTF-UTF8

The scanned dictionary pages provided for this task were saved in rich text format (RTF), with one document supplied per page. A part of the page for the entry shown is illustrated in figure 5.

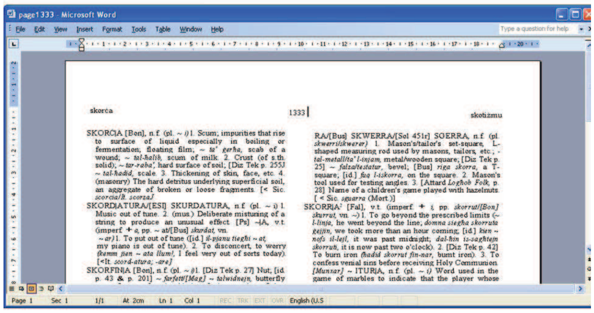


Fig. 5: *RTF Format*

Thus the first task was to convert these files in a format which could be more easily read by subsequent stages of the program. The initial task was correct encoding of characters into Unicode (UTF8).

At first, the idea of manually parsing out all the RTF controls from the sample pages was considered. However it soon became clear that writing a separate tool for converting RTF files into plaintext was beyond the scope of this project.

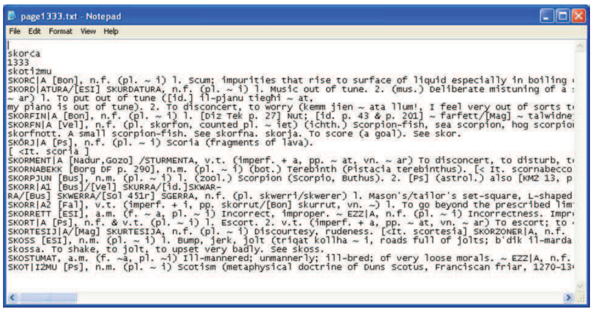
A number of freeware third-party tools available over the internet were tried but none of them were able to to correctly translate the Maltese characters into Unicode UTF8. For this reason a macro was recorded for Microsoft word.

5.2 Stage 1: Separation of Entries

The result of conversion to UTF8 is shown in figure 6.

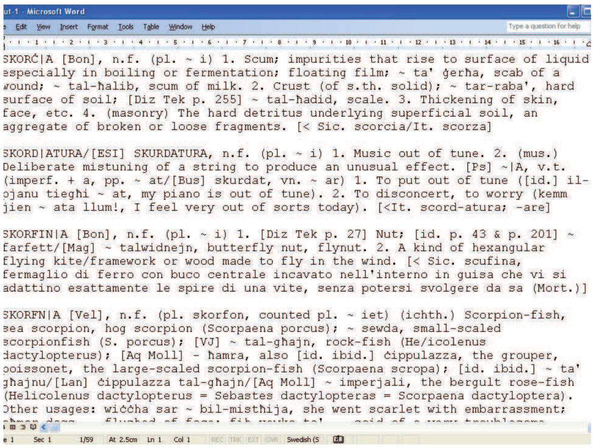
While the dictionary pages were now in plain text form, they still contained a large number of irregularities which needed to be cleaned up before the parsing of the pages on a per-entry level could begin. The purpose of this processing stage was to bring all of the entries from the plain text pages together in a single file, neatly separated by newline characters.

However, as a result of the OCR quality of the sam-



```
page1333.txt - Notepad
File Edit Format View Help
skorċa
1333
skotiżmu
SKORĊ|A [Bon], n.f. (pl. ~ i) 1. Scum; impurities that rise to surface of liquid especially in boiling
SKORD|ATURA/[ESI] SKURDATURA, n.f. (pl. ~ i) 1. Music out of tune. 2. (mus.) Deliberate mistuning of a
~ ar) 1. To put out of tune ([id.] il-pjanu tiegħi ~ at,
my piano is out of tune). 2. To disconcert, to worry (kemm jien ~ ata llum!, I feel very out of sorts t
SKORFIN|A [Bon], n.f. (pl. ~ i) 1. [Diz Tek p. 27] Nut; [id. p. 43 & p. 201] ~ farfett/[Mag] ~ talwidne
SKORFN|A [Vel], n.f. (pl. skorfon, counted pl. ~ iet) (ichth.) Scorpion-fish, sea scorpion, hog scorpio
skorfnott. A small scorpion-fish. See skorfna. skorja. To score (a goal). See skor.
SKÒRJ|A [Ps], n.f. (pl. ~ i) Scoria (fragments of lava).
[ <It. scoria ]
SKORMENT|A [Nadur,Gozo] /STURMENTA, v.t. (imperf. + a, pp. ~ at, vn. ~ ar) To disconcert, to disturb, t
SKORNABEKK [Borg DF p. 290], n.m. (pl. ~ i) (bot.) Terebinth (Pistacia terebinthus). [< It. scornabecco
SKORPJUN [Bus], n.m. (pl. ~ i) 1. (zool.) Scorpion (Scorpio, Buthus). 2. [Ps] (astrol.) also [KMŻ 13, p
SKORR|A1 [Bus]/[Vel] SKURRA/[id.]SKWAR-
RA/[Bus] SKWERRA/[Sol 451r] SGERRA, n.f. (pl. skwerri/skwerer) 1. Mason's/tailor's set-square, L-shaped
SKORR|A2 [Fal], v.t. (imperf. + i, pp. skorrut/[Bon] skurrut, vn. ~) 1. To go beyond the prescribed lim
SKORRETT [ESI], a.m. (f. ~ a, pl. ~ i) Incorrect, improper. ~ EZZ|A, n.f. (pl. ~ i) Incorrectness. Impr
SKORT|A [Ps], n.f. & v.t. (pl. ~ i) 1. Escort. 2. v.t. (imperf. + a, pp. ~ at, vn. ~ ar) To escort; to
SKORTESIJ|A/[Mag] SKURTESIJA, n.f. (pl. ~ i) Discourtesy, rudeness. [<It. scortesia] SKORZONER|A, n.f.
SKOSS [ESI], n.m. (pl. ~ i) 1. Bump, jerk, jolt (triqat kollha ~ i, roads full of jolts; b'dik il-marda
skossa. To shake, to jolt, to upset very badly. See skoss.
SKOSTUMAT, a.m. (f. ~a, pl. ~i) Ill-mannered; unmannerly; ill-bred; of very loose morals. ~ EZZ|A, n.f.
SKOT|IŻMU [Ps], n.m. (pl. ~ i) Scotism (metaphysical doctrine of Duns Scotus, Franciscan friar, 1270-13
```

Fig. 6: *UTF8 Format*



```
ut-1 - Microsoft Word
Edit View Insert Format Tools Table Window Help
SKORĊ|A [Bon], n.f. (pl. ~ i) 1. Scum; impurities that rise to surface of liquid
especially in boiling or fermentation; floating film; ~ ta' ġerħa, scab of a
wound; ~ tal-ħalib, scum of milk. 2. Crust (of s.th. solid); ~ tar-raba', hard
surface of soil; [Diz Tek p. 255] ~ tal-ħadid, scale. 3. Thickening of skin,
face, etc. 4. (masonry) The hard detritus underlying superficial soil, an
aggregate of broken or loose fragments. [< Sic. scorcia/It. scorza]

SKORD|ATURA/[ESI] SKURDATURA, n.f. (pl. ~ i) 1. Music out of tune. 2. (mus.)
Deliberate mistuning of a string to produce an unusual effect. [Ps] ~|A, v.t.
(imperf. + a, pp. ~ at/[Bus] skurdat, vn. ~ ar) 1. To put out of tune ([id.] il-
pjanu tiegħi ~ at, my piano is out of tune). 2. To disconcert, to worry (kemm
jien ~ ata llum!, I feel very out of sorts today). [<It. scord-atura; -are]

SKORFIN|A [Bon], n.f. (pl. ~ i) 1. [Diz Tek p. 27] Nut; [id. p. 43 & p. 201] ~
farfett/[Mag] ~ talwidnejn, butterfly nut, flynut. 2. A kind of hexangular
flying kite/framework or wood made to fly in the wind. [< Sic. scufina,
fermaglio di ferro con buco centrale incavato nell'interno in guisa che vi si
adattino esattamente le spire di una vite, senza potersi svolgere da sa (Mort.)]

SKORFN|A [Vel], n.f. (pl. skorfon, counted pl. ~ iet) (ichth.) Scorpion-fish,
sea scorpion, hog scorpion (Scorpaena porcus); ~ sewda, small-scaled
scorpionfish (S. porcus); [VJ] ~ tal-għajn, rock-fish (He/icolenus
dactylopterus); [Aq Moll] - ħamra, also [id. ibid.] ċippulazza, the grouper,
poissonet, the large-scaled scorpion-fish (Scorpaena scropa); [id. ibid.] ~ ta'
għajnu/[Lan] ċippulazza tal-għajn/[Aq Moll] ~ imperjali, the bergult rose-fish
(Helicolenus dactylopterus = Sebastes dactylopteras = Scorpaena dactyloptera).
Other usages: wiċċha sar ~ bil-mistħija, she went scarlet with embarrassment;
Sec 1 1/59 At 2.5cm Ln 1 Col 1 REC TRK EXT OVR Swedish (S
```

Fig. 7: *Separated Entries, one per line*

ple pages, considerable work was required to get the entries into this format. The major issues encountered at this stage were the following:

- inclusion of extra page information such as page numbers and first and last headwords. These can clearly be seen in the first three lines of figure 6.
- Distribution of single entries over multiple lines. The line starting "my piano is out of tune" is an an example, being part of the entry for *SKORDAR*, one of the senses of the entry whose stem is *SKORD*.
- multiple entries combined together into single line.

  Clearly, a key issue is the individuation of distinct entries, and in particular how to determine whether a given line represents the beginning of a new entry.

  The following criteria were employed. If all these are met, then the line is deemed to be the start of a new entry (this does not exclude the possibility that other entries might be appended to it:

  – The first character is a letter (euphonic i is ignored).
  – It is followed by a punctuation character.
  – Last character of previous entry is not a hyphen.
  – The first letters of the line are in ascending alphabetic order with respect to the previous entries.

  The selection of these criteria proved to be quite problematic, as for every feature which a group of headwords seem to share, there were inevitably be some which did not follow the pattern. The objective success of the criteria adopted is reflected in the results reported in section 6.

- Finally, there are a number of OCR errors. Some of these were common enough to warrant automatic correction. For example, the string "l." at the beginning of a word sense should be replaced with the string "1."

The output of this phase, with one entry per line, is shown in figure 7.

5.3 Stage 2: Parsing of Entries

Now that the text entries have been neatly separated, the next step was to parse each of them and extract all necessary pieces of information which would be needed in the third and final output stage.

The basic order of operations is as follows:

```
For each line
    - Get first word
    - If uppercase:
        - Set type = "ENTRY"
        - Get number after headword
        - Get headword, stem & root
        - Get etymological info, eg "[< Eng./It. Stalatt
        - Get & format alternate spellings
        - List any subentries
        - Get senses for each subentry

    - If lowercase:
        - Set type = "CROSS-REF"
        - Get & format alternate spellings
        - List cross-references with variation types
    Add to entries list
Encode entries list using JSON and write to file.
```

While the output from stage one was a simply plain-text file, the output for this stage needed to have a lot more structure to it, now that each entry has been broken down into its sub-parts. As per the system design, it is only in stage three where the entries are converted into the final LMF output.

Thus the internal output from the current stage of processing must be independent of the final LMF output format. One option for encoding these entries was to save them in a generalised XML format, independent of the LMF output in stage three. While viable, a faster and equally as robust method was to output the structured entries in JSON format.

JSON (JavaScript Object Notation)[2] is a very simple text-based data-interchange format that is com-

[2] www.json.org

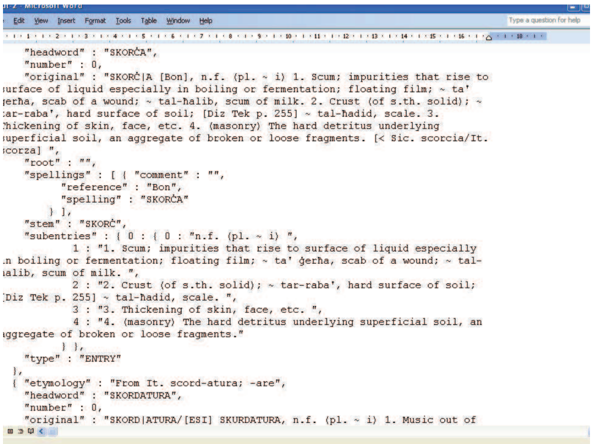


Fig. 8: *Output after parsing entries*

Fig. 9: *Output after conversion to LMF*

pletely language independent but uses conventions that are familiar to programmers of the C-family of languages. Essentially, JSON is built on two structures: (i) a collection of name/value pairs, and (ii) an ordered list of values, realized in most languages as an array, vector, list, or sequence.

The actual translation was done using the open source `demjson` Python module (7). With just one line of code, the entire entries list was converted into JSON format and saved to file, to be fed as input into the final stage of processing. Figure 8 shows the result of this process.

5.4 Stage 3: Output LMF

With the entries now individually separated and broken down into their sub-parts, the final stage of the process was to generate the output lexicon containing the parsed entries in an LMF-compliant XML format as diagrammed in figure 10.

To further separate the processing from the presentation aspects of the code, it was decided to use a number of templates in the generation of the final LMF output. This is not to say that the code of stage three is completely output-independent, however the use of templates definitely helped to promote this separation.

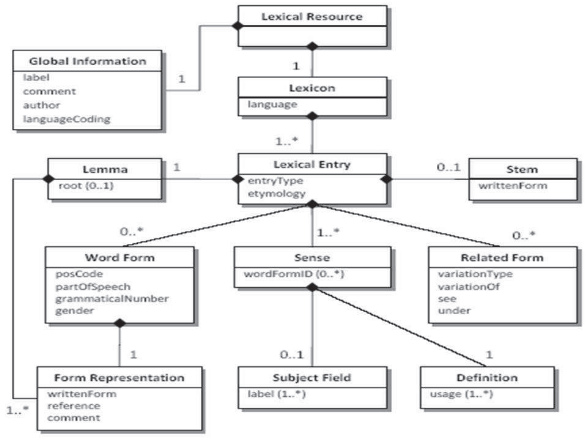


Fig. 10: *XML output format*

The templates used are listed below:

- `LexicalResource.xml` Represents the entire lexicon and root element for the output XML.
- `LexicalEntry.xml` A single entry in the lexicon (including cross-references).
- `FormRepresentation.xml` Used to represent alternative spellings of the same word.
- `WordForm.xml` Used to represent different forms of a word, e.g. through conjugation.
- `Sense.xml` Represents a single sense (definition and usage) or a word/word form.
- `RelatedForm.xml` Used in cross-references to indicate the referenced entry.
- `feat.xml` A simple attribute/value pair.

The output of stage 3 is the final LMF-compliant lexicon XML file as shown in figure 9.

5.5 Format Validation

To ensure that the generated XML lexicon was well structured and consistent with the LMF standard, a final step of validation was performed on the file. Firstly, by opening the newly created file with an XML parser, the file contents was parsed and implicitly checked to be a valid XML document.

The successfully parsed XML was then validated against the LMF DTD file to check conformity to the LMF standard. Both these tasks are achieved using the `lxml` Python library.[3]

6 Results and Evaluation

6.1 LMF Conformity

When validating against the official LMF Rev 16 DTD the generated lexicon did not pass because (i)

[3] The `lxml` library is available at `http://codespeak.net/lxml/validation.html`

the `WordForm` element had no attribute id, (ii) the `RelatedForm` element had no attribute id and (iii) in many cases the `RelatedForm` elements targets attribute contains nonexistent IDs.

The first 2 points are genuine non-conformities to the official LMF standard. However, the inclusion of a simple id attribute is only a very minor infringement, and for the purposes of the project was deemed a fair one to make. In order to accommodate the addition of these new attributes, a modified version of the original LMF DTD was created and used for subsequent validation.

In the third case, the issue is that as this lexicon only covers a sample of pages from the entire dictionary, some of the cross-referenced entries do not appear in the same file. This is quite understandable. To get around this issue and continue validation, all cross-referenced entries were temporarily removed from the lexicon. Once this was done, the output successfully passed DTD validation suggesting that if the file were to contain the entire dictionary it should also comply with the LMF standard.

6.2 Method of Evaluation

Once the output lexicon had been generated and validated, the next important step was to evaluate its accuracy against the original dictionary. This was achieved by manually comparing the output of the program with the original OCRd dictionary pages, and enumerating the number of correct, partial, incorrect, and missing entries.

First a more human-friendly version of the original XML lexicon was generated for evaluation purposes using PHP. Two HTML documents were generated and printed out for the evaluator to manually check and mark against for each entry.

6.3 Evaluation Criteria

Each entry extracted and placed into the output lexicon was given one of the following designations:

- **Correct**: the entire entry was correctly extracted.
- **Partial**: the headword and some parts of the entry are correct; however some parts are in error.
- **Incorrect**: the headword is seriously corrupted or not a headword at all.

In addition, for each extracted entry a count of any missing entries not picked up by the parser was also kept. This information was then used in the calculation of the programs accuracy, as explained in the following section.

6.4 Equations Used

The equations used in the calculation of the accuracy score are given below.

- Strict Score $= \frac{\text{Correct}}{\text{Total+Missed}}$
- Lax Score $= \frac{\text{Correct+Partial}}{\text{Total+Missed}}$

	known	unknown
Page Range	1333-1347	1480 - 1490
Total Entries	360	370
Correct	290	261
Partial	64	84
Incorrect	6	25
Missed	34	47
Strict Score %	73.6	62.59
Lax Score %	89.85	82.73

Table 2: *Evaluation data*

6.5 Known and Unknown Pages

In the development of this project, the same subset of dictionary pages was used throughout. This would certainly have introduced a certain bias of the program to perform better on these pages than it would on the dictionary as a whole. To test this, the programs accuracy was evaluated and analyzed on two subsets of dictionary pages one which was used throughout development ("known"), and one which has never been shown to the program (or developer) before ("unknown"). The results of both cases are presented in the next section.

7 Discussion

7.1 Results

Although the 62.59% for unknown pages leaves plenty of room for improvement, as discussed further below, these results are quite promising. This percentage represents a sufficently high level of accurate results to warrant further investigation of methods which can further reduce the human effort required to filter out incorrect results.

7.2 OCR Problems

The primary difficulty encountered in this project was the quality and consistency of the sample dictionary pages provided. Although passed through OCR software and supposedly checked by hand, the accuracy of the provided pages was far from ideal.

Apart from oft-mistaken character sequences such as "]"for "J" and "|" for "I", the major issue encountered was that of inconsistent entry formatting. This in particular included entries split across multiple paragraphs, multiple entries collapsed into a single line, and incorrect block indentation. While noticeable to human readers, these issues presented a major hurdle for the extraction process, and at least half of all the effort put into this project was devoted to correcting these OCR-related errors.

7.3 Variation of Notation in Dictionary

Another source of difficulty encountered was the notational variation present in the source dictionary. This was especially true for multiple word forms or definitions within an entry.

While in some entries they are listed in one format, in others they may be listed in a different format. It should be noted that these inconsistencies have been created by the author of the dictionary. Though the author may have had his reasons for such variations, they are neither obvious nor fully documented. As a result, a number of errors found in the output of this program can be attributed to these inconsistencies.

Another case of inconsistency is the use of the tilde character as a back-reference. Most of the time it refers to the previously-established headword stem, but sometimes it refers to the entire headword. Once again, what it refers to in each case is not always obvious to the reader, let alone a computer, and this ambiguity contributed to a substantial number of word form errors generated by the program.

8 Limitations

8.1 Lossiness RTF Conversion

The first stage in this project involved converting the RTF sample pages into plain text equivalents. While this provided many benefits it terms of ease of development, it also inevitably presented its own set of limitations. One of these is the loss of all formatting information, such as bold and italic text. As such formatting may contain additional information about the entry (e.g. examples of use are written in italics), it would have been preferred if these could have been retained and used during the extraction process.

8.2 Cross-Reference IDs

In the submitted program, whenever a cross-reference entry is processed, the existence of the main entries referred to are not explicitly checked. Instead, they are simply transcribed from the source, which means that cross-references may exist in the output lexicon with invalid reference IDs. As only a handful of pages were processed for the purposes of this project, the verification of these IDs would be somewhat futile. However in a complete version of the lexicon, these ID references would need to be verified.

8.3 Entry Definitions

Most of the effort carried out in this project is devoted to extracting headword variations and different word forms. Less focus however was placed on the parsing of the word definitions themselves, and in many cases the information placed in each Sense element is simply copied verbatim from the dictionary. In particular,the following issues were not addressed:

- Non-textual characters are not removed.
- Word definitions are not separated from examples of usage.
- Abbreviations and back-references are not replaced with their full equivalents.

  We do not anticipate that addressing any of these points would introduce major structural changes to the program.

9 Future Work

9.1 Scaling Up

This experiment was carried out on a total of approximately 700 lexical entries taken from 20 dictionary pages. Although results are promising, the extent to which they generalise is not clear and for this reason an absolute priority is to repeat the experiment on a much larger dataset.

9.2 More Thorough Error Correction

While a substantial amount of work in this project was devoted to error correction, the techniques used are far from complete. Many of the OCR errors found in the sample pages are not easily correctable with basic pattern-matching techniques, and require deeper analysis as to how they occur and can be removed. With a more dedicated effort devoted to the correction of these errors, the accuracy of the system could undoubtedly be pushed significantly higher.

9.3 Use of Statistical Methods

The level of accuracy achieved in this project was achieved through the use of standard string pattern-matching with regular expressions. Whilst these methods are highly effective when used in the correct contect, one major limitation is that such methods do not exploit the statistical regularities inherent in the language of dictionary entries.

A possible way forward would be to develop statistical models for language of dictionary entries, and to use these models to error correct the dictionary entries obtained by OCR. Inspection of dictionary entries reveals that a dictionary entry is composed of several parts not all of which share the same language. Hence, there is scope for investigating the sublanguages that make up dictionary entries and developing statistical models for each.

References

[1] U. A. and A. Twist. Auditory and visual lexical decision in maltese. In C. B., R. Fabri, E. Hume, M. Mifsud, T. Stolz, and M. Vanhove, editors, *Introducing Maltese Linguistics, Selected papers from the 1st International Conference on Maltese Linguistics*, pages 233–249. John Benjmins Publiching Company, 2007.

[2] J. Aquilina. *Concise Maltese Dictionary*. Midsea Books, Valletta, 2006.

[3] B. Boguraev, T. Briscoe, J. Carroll, D. Carter, and C. Grover. The derivation of a grammatically indexed lexicon from the longman dictionary of contemporary english. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 193–200, Stanford, California, USA, July 1987. Association for Computational Linguistics.

[4] G. Francopoulo, M. George, N. Calzolari, M. Monachini, N. Bel, M. Pet, and C. Soria. Lexical markup framework (lmf). In *Proc. LREC 2006*, pages 233–236. ELRA, 2006.

[5] ISO/TC37/SC4. *Language Resource Management, Lexical Markup Framework (LMF).* International Organisation for Standardisation, 24613:2006 edition, 2006.

[6] M. Rosner, J. Caruana, and R. Fabri. Maltilex: A computational lexicon for maltese. In *In Proceedings of the Workshop on Computational Aspects of Semitic Languages, ACL/COLING98*, pages 97–105, 1998.