Automated Compilation of Semitic Lexica

(The Arabic-Syriac Lexicon Project: Report I)

George Anton Kiraz
Computer Lab., University of Cambridge
and
The Syriac Computing Institute

Daniel Ponsford
Cambridge University Press
and
The Syriac Computing Institute

This paper constitutes the first formal report of the Arabic-Syriac Lexicon project, from The Syriac Computing Institute. We show here a practical method for compiling an electronic lexicon, based on published lexica, for Semitic languages. Our method, though is not a computational morphology system, makes a practical use of the root-and-pattern nature of Semitic morphology.

1 Introduction

The Arabic-Syriac Lexicon (ASL) Project aims at producing a hard-copy Arabic-Syriac lexicon based on the current published Syriac-Arabic lexica. The methodology of compiling the lexicon is as follows:

Phase 1. Compiling an electronic Syriac lexicon which we call the Syriac word list (SWL).

Phase 2. Compiling an electronic Arabic lexicon which we call the Arabic word list (AWL).

Phase 3. Marking correspondences between SWL and AWL based on published Syriac-Arabic lexica.

Phase 4. Generating a draft Arabic-Syriac lexicon and submitting it to a reviewing committee.

Phase 5. Incorporating the changes proposed in Phase 4.

Phase 6. Compiling the final Arabic-Syriac lexicon.

This paper represents the first formal report of the project. It discusses the work done so far (Phases 1 and 2), and describes, in a brief manner, the remaining phases. Section 2 describes the database schema used to hold the two word lists, SWL and AWL. Section 3 describes an automated method for compiling the word lists SWL and AWL and constitutes the main contribution of this paper. Section 4 describes Phases 3-6 briefly.

2 Word Lists Database Schema

This section describes the database schema which will manage the Syriac and Arabic word lists. The schema is based on the schema of SEDRA (Syriac Electronic Data Retrieval Archive), described in detail in Kiraz (forthcoming in *Orientalia Christiana Analecta*). Here we shall give a simple version.

The database consists of the following records: the record ROOT holds instances of roots, e.g. */ktb/*; the record LEXEME holds instances of lexical entries derived from roots, e.g. */kitaab/* 'book' < */ktb/*; the record WORD holds instances of words derived from

lexemes, e.g. */kutub/* < */kitaab/*.

The database is designed in the network model, as opposed to the relational. In the network model, a set defines a one-to-many relationship between two records. In our schema, the following sets are defined: ROOT-LEXEME allows each root instance to own many lexeme instances, e.g. the root */ktb/* owns the lexemes */kitaab/* 'book' and */maktab/* 'office', and the set LEXEME-WORD allows each lexeme instance to own many word instances, e.g. the lexeme */kitaab/* owns the word */kutub/*. The order of sets is determine from the morphological features of the entries; for details see Kiraz (forthcoming in *OCA*).

For the purposes of this project, the following additional records were added to the schema of SEDRA: the record IDIOM holds instances of idioms and is a member of the LEXEME record, e.g. the lexeme كتاب owns the idiom أهل الكتاب .

3 Automated Compilation of Semitic Word Lists

The first two phases of the project aim at compiling two word lists: one for Syriac (SWL) and one for Arabic (AWL). The former list is based on C. Brockelman 1928, P. Smith, and T. Audo 1897; the latter is based on Wehr 1974.

To achieve this, we took advantage of the *root-and-pattern* nature of Semitic morphology. A list of templates was compiled. Given the consonantal root of a lemma and choosing the templates where this root occurs, an algorithm generates the strings of the chosen forms. For example, given the root */ktb/* and the template $C_1iC_2aaC_3$, the algorithm computes the string */kitaab/*. (This does not imply we are using/proposing a computational morphology generator.)

The code is written in dBASEIII+. The system runs under Sakhr2 (running on top of DOS) to allow data entry in Arabic. Since dBASEIII+ is a relational database, the schema used by the actual system differs from the one described in Section 2. We have a utility which converts the relational database into the network database.

3.1 Stating the Problem

The problem can be stated as follows: given a root, the system should determine its class (i.e. strong, initial hamzated, etc.) and should generate a list of *feasible* verbs and nouns for that root class; from this list, the user can choose the *cited* verbs and nouns. Therefore, the system needs to know a list of all feasible verbal and nominal patterns for each root class.

In addition, the system should allow: homograph roots, e.g. سكر 'notion of drunkenness' and سكر 'notion of sugar'; variant vocalisation of verbs and nouns, e.g. سَخُنَ ، سَخَنَ and سَخِنَ ; variant templates for a lexical entry, e.g. مَدُوسٌ and مُدَاسٌ ; lexical homographs, e.g. ظَهر 'cast iron' and ظَهر 'back, rear'; irregular and foreign forms, e.g. شُوَيَّة and كُوكُتَيْل; assimilation of form VIII, e.g. زَهَرَ > *ازتَهَر > ازدَهَر .

3.2 Templates database

The *Templates database* (TDB) holds the morphological information of a Semitic language. Templates are written in a text file; here is an example of nominal patterns

```
01  1a2.3      p:1u2uW.3     p:?a1.2aA3     d:1u2aY.3
01  ?a1.2a3    p:?a1aA2i3    f:1a2.3aA'     f:1u2.3a$      p:1u2.3aYaAt
```

The first column gives the root class (01 = triliteral strong, 02 = quadriliteral, 03 =

geminate 2nd radical, etc.); the second column gives the singular pattern, followed by other nominal forms prefixed by p:, fp:, d: or dp: (where p = plural, f = feminine, d = diminutive). In the pattern string, numbers represent radicals; all other characters are either vowels or affixes (. = سكون, A = ا, W = و, Y = ي, @ = ة, $ = ى, ? = ء).

The data is imported into database records as shown in the example below which represents the first line of the above example:

RootClass:	01
Singular:	1a2.3
Plural:	{ 1u2uW.3, ?a1.2aA3 }
Diminutive:	1u2aY.3

(sets of maximun *n* entries are defined in dBASEIII+ as *n* fields). After importing the data, the patterns are Arabised, e.g. ma1.2uW.3 > مَ1ُ2ُوْ3. The lists of all templates were drawn from Wright's *A Grammar of the Arabic Language*. Templates found in Wehr's dictionary were also added.

3.3 Selecting the right templates

The user enters a root, e.g. كتب (and a version number in case of root homographs). The following simple algorithm identifies the root class (r_n = *n*th radical, but r_3 for double roots is a shedda; Is...(*ch*) functions return True/False):

```
GetRootClass(r1, r2, r3, r4)
    if r4 != 0   class = QUADRILITERAL
    else if IsShedda(r3) do
            if IsStrong(r1)   class = STRONG_SHEDDATED
            else if IsHamza(r1)   class = HAMZATED_SHEDDATED
            else if IsWaw(r1)   class = WAAW_SHADDATED
            else   class = YAA_SHEDDATED
    else if IsHamza(r1) do
            if IsStrong(r3)   class = INITIAL_HAMZATED
            else if IsWaaw(r3)   class = INITIAL_HAMZATED_FINAL_WAAW
    else if IsHamza(r2)   class = MEDIAL_HAAMZATED
    else if IsHamza(r3)   class = FINAL_HAMZATED
    else if IsWaaw(r1) do
            if IsStrong(r2) and IsStrong(r3)   class = INITIAL_WAAW
            else if IsWaaw(r2)   class = INITIAL_MEDIAL_WAAW
            else if IsYaa(r3)   class = INITIAL_WAAW_FINAL_YAA
    else if IsWaaw(r2)   class = MEDIAL_WAAW
    else if IsWaaw(r3)   class = FINAL_WAAW
    else if IsYaa(r1)   class = FINAL-YAA
    else if IsYaa(r2)   class = MEDIAL_YAA
    else if IsYaa(r3)   class = FINAL_YAA
    else class = STRONG_TRILITERAL
```

The corresponding templates are read from TDB and the radical numbers are replaced with the root radicals, e.g. 1i2aa3 > كِتَاب. A list of feasible entries appear on the screen. The user selects those which are cited in the lexicon. Once a form is selected as a cited form, the user has the option of entering plural, feminine, diminutive and affixed forms, idioms and prepositional phrases which are associated with the entry, e.g. كِتَابِي، الكِتَاب، كُتُبِّب، كُتُب، أهل

الكِتَاب.

3.4 The Lexicon Database

The information selected by the user is stored in four files for verbs, nouns, verbal idioms and nominal idioms. These constitute the lexical database (LDB). An example of a verbal entry is:

FormNo:	I
Root:	كتب
Pattern:	1a2a3
ArabicForm:	كَتَبْ
Preposition:	على

An example of a nominal entry is:

Link:	5 (record no. of the singular form)
RootClass:	strong
Root:	كتب
Pattern:	1i2aa3
ArabicForm:	كِتَاب
VariantFlag:	F (this field is set T if the entry is a variant)
VariantOf:	nil (if VariantFlag is T, this field gives the form this entry is variant of)
Affix:	كتابي
Preposition:	nil

3.5 Relational to Network Conversion

LDB is shown above in a relational form. In section 2 we gave the network model of our database. We have a utility which converts LDB into the network model database.

4 Phases 3-6 Briefly

In this section we give a brief description of what we intend to do for Phases 3-6. After the two word lists SWL and AWL are compiled, correspondences between the two lists will be marked based on published Syriac-Arabic lexica (Phase 3). This is not a straight forward matter: A word in one list may correspond to a phrase or idiom in the other list; different verbal forms of the same root in one list may correspond to verbal forms of different roots in the other (e.g. Syriac form I perf. */zban/* 'to buy' corresponds to Arabic form I pass. */'ištara/* < */šry/*, while Syriac form II of the same root, */zabben/* 'to sell', corresponds to Arabic */baa'a/*. The will system handle complex correspondences.

Phase 4 aims at generating a draft Arabic-Syriac lexicon from the database. The draft will be reviewed by a reviewing committee. The committee tackles problems such as: if word A in one language corresponds to word B in the other, it does *not* imply that word B corresponds to word A (in other words one cannot just convert an x-y lexicon to a y-x lexicon); grouping the set of Arabic words corresponding to each Syriac word in subsets based on the semantics of the words.

Phases 5 and 6 aim at incorporating the changes of Phase 4 and compiling the final

Arabic-Syriac lexicon using the same algorithm of Phase 4.

Acknowledgements

We would like to thank Dr Andrew Criddle for entering Syriac lexical data.

Addresses:

George A. Kiraz
St John's College
Cambridge CB2 1TP, United Kingdom
E-mail: george.kiraz@cl.cam.ac.uk

Daniel Ponsford
11 Orchard Road
Great Shelford
Cambridge CB2 5AA, United Kingdom