ARMY RESEARCH LABORATORY

# Adapting an Arabic Morphological Analyzer to Serve Word Lookup for Military Tasks

**by Jeffrey C. Micher**

**ARL-TR-4945** **September 2009**

## NOTICES

### Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

**Army Research Laboratory**
Adelphi, MD 20783-1197

**ARL-TR-4945** **September 2009**

# Adapting an Arabic Morphological Analyzer to Serve Word Lookup for Military Tasks

**Jeffrey C. Micher**
**Computational and Information Directorate, ARL**

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED (From - To) |
|---|---|---|
| September 2009 | Final | |

**4. TITLE AND SUBTITLE**

Adapting an Arabic Morphological Analyzer to Serve Word Lookup for Military Tasks

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

Jeffrey C. Micher

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

U.S. Army Research Laboratory
ATTN: RDRL-CII-T
2800 Powder Mill Road
Adelphi, MD 20783-1197

**8. PERFORMING ORGANIZATION REPORT NUMBER**

ARL-TR-4945

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for public release; distribution unlimited.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

The U.S. Army Research Laboratory has forward engineered the Buckwalter Arabic Morphological Analyzer (BMA) to perform basic lookup functions of ambiguous Arabic text, resulting in a new Buckwalter-based Lookup Tool (BBLT). BBLT is implemented as a Web application on top of the existing BMA algorithms and dictionaries. The variety of output types and options allow the user to see the output in different ways, for example, grouped by unique meanings, with unknown words transliterated, or as Arabic text with full diacriticization. The creation of this tool serves as an example of how one can extend the functionality of an existing translation resource to apply to a language of interest to the Army.

**15. SUBJECT TERMS**

Morphological analyzer, Arabic, dictionary, forward engineering, diacritics, language learning

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Jeffery Micher |
| Unclassified | Unclassified | Unclassified | UU | 26 | 19b. TELEPHONE NUMBER (*Include area code*) (301) 394-0316 |

## Contents

## List of Figures



## List of Tables

# 1. Introduction and Goals

As human linguist resources are scarce, the Army and other Department of Defense (DoD) entities benefit from using automated tools for language processing and translation. The Army's warfighters and analysts must perform many tasks that require language translation, from simple summarizing, or gisting, of documents for triage to full translation of spoken input in the field. However, the translation and language processing tools currently available in languages of interest are not customized for the military domain or for military tasks. Rather, these resources have been developed for generic commercial purposes or for specific research purposes (e.g., advancing the state of the art in algorithms in a particular processing area). The challenge, therefore, is to reconfigure, re-engineer, or build upon the available tools to apply them to such new domains and tasks.

One research tool with particular promise for military uses is the Buckwalter Arabic Morphological Analyzer (BMA) (Buckwalter, 2005), which provides a complete morphological analysis of Arabic words, including English glosses for the various analyzed portions of words. The BMA was developed for the generic purpose of providing a complete morphological analysis of Arabic words, with English glosses for each unique analysis. But because of its extensive coverage of Arabic words, it has great potential to support the task of word lookup by both linguists and non-linguists such as analysts or military personnel who are performing the task of document triage. To realize that potential, an adaptation of the BMA was undertaken and is documented in this report.

# 2. Background

## 2.1 Arabic Morphology

The morphology of the Arabic language is rich and complex. Words can be inflected (prefixes and suffixes added) to express variations in tense-aspect, person, number, and gender. Inflected words can, in turn, undergo further modification by the addition of clitics, which are "grammatically independent and phonologically dependent words. Clitics are pronounced like an affix, but work at the phrase level" (Wikipedia, 12 April 2009). For example, Arabic clitics express possession on nouns, objects of verbs and prepositions, prepositions themselves, and various conjunctions, all of which can be grammatically analyzed as independent phrasal constituents, yet they form a phonological unit with the words to which they are attached. Figure 1 shows a schema that depicts this morphology (parentheses indicate optionality).

| (clitic) + (prefix) + stem + (suffix) + (clitic) |
|---|

Figure 1. Clitic morphology.

Figure 2 shows some examples of Arabic words containing clitics.

walilmaktabati وللمكتبت

wa-li-l-maktabat-i
wa + li + Al + maktab + at + i
clitic clitic clitic stem suffix suffix
and/conj for/prep the/det library/noun plural genitive
*and for the libraries*

fa-sa-ya-ktab-Unna-hA فسيكتبونها
fa + sa +ya + ktab + Unna + hA
clitic clitic prefix stem suffix clitic
so/conj future 3masc-imp write 3pl-imp 3fem-sg-obj
*so they will write it*

wa-li-sayArat-iy ولسيارتي
wa + li + sayArat + i + y
clitic clitic stem suffix clitic
and/conj for/prep car/noun genitive my/possessive-pn
*and for my car*

Figure 2. Examples of Arabic clitics.

In addition, words in Arabic are derived from roots consisting usually of three consonants, which convey a semantic concept, to which are added "infixes," expressing many different derivational forms (figure 3).

| | |
|---|---|
| **k-t-b** | basic meaning of "writing" |
| **k**a**t**a**b**a | wrote |
| **k**i**t**A**b** | book |
| ma**kt**a**b**at | library |
| ma**kt**U**b** | written / letter |
| ma**kt**a**b** | office |
| **k**A**t**i**b** | writer |

Figure 3. Examples of Arabic infixes.

Furthermore, Arabic script allows the omission of short vowel diacritics, which is the way that most Arabic is written. For example, most of the noun case inflections are short vowels added to the end of a word, which then are not written (table 1).

Table 1. Examples of the omission of short vowel diacritics.

| Pronounced | Case | Written |
|---|---|---|
| AlkitAbu الكتابُ | Nominative | AlktAb الكتاب |
| AlkitAba الكتابَ | Accusative | AlktAb الكتاب |
| AlkitAbi الكتابِ | Genitive | AlktAb الكتاب |
| AlkitAbun الكتابٌ | Nominative Indefinite | AlktAb الكتاب |
| AlkitAbin الكتابٍ | Genitive Indefinite | AlktAb الكتاب |

As a result of this rich and complex morphology coupled with the writing convention of omitting short vowels, Arabic tokens (elements between white space) are highly ambiguous. Any natural language processing of Arabic needs to take these aspects into account.

## 2.2 Buckwalter Morphological Analyzer 2.0

The BMA 2.0 (Buckwalter, 2004) provides a complete analysis of the ambiguous morphological complexity of Arabic tokens. This analysis includes all possible analyses for each token, which include any spelling variants of the token; the lemmaID for the stem of the token; a vocalization, including short vowels restored; the part of speech of the token; and an English gloss.

*Output of the Morphological Analyzer*

The output of the BMA version 2.0 is delivered as extensible markup language (XML) text. The following details the XML tags and their meanings:

- <token_Arabic>: This tag is an Arabic token. The analyzer first tokenizes input text on whitespace.
- <variant>: This tag shows possible spelling variants of the Arabic token. There can be multiple variants.
- <solution>: This tag features the morphological analysis containing a valid solution. There can be multiple solutions.
- <x_solution>: This tag shows words not found in the Buckwalter database, but which could be valid words, such as proper nouns. There can be multiple x_solutions.
- <lemmaID>: This tag is an identifier for a unique meaning of a token.
- <voc>: This tag indicates vocalization, the Arabic token with diacritics added.
- <pos>: This tag shows the part of speech in the morphological analysis, reflecting any clitics or inflections as morphemes, each of which are marked with a tag to indicate a grammatical function. These morphemes are separated by a "+" and are tagged with a "/".
- <gloss>: This tag provides an English gloss of each of the morphemes from the analysis. There may be multiple glosses in a given solution, separated by "/".

Figure 4 shows an example of the XML tags in use.

```
<token_Arabic>ولد
 <variant>wld
  <solution>
   <lemmaID>walad-i_1</lemmaID>
   <voc>walada</voc>
   <pos>walad/PV+a/PVSUFF_SUBJ:3MS</pos>
   <gloss>give birth to + he/it [verb]</gloss>
  </solution>
  …
  <x_solution>
      <voc>wald</voc>
      <pos>wa/CONJ+ld/NOUN_PROP</pos>
      <gloss>and + NOT_IN_LEXICON</gloss>
    </x_solution>
 </variant>
 </token_Arabic>
```

Figure 4. Examples of the XML tags applied to Arabic text.

The BMA also identifies non-Arabic tokens—Latin characters, Western numbers, and punctuation—are marked as shown in figure 5.

```
<token_notArabic>test
  <analysis>test/LATIN</analysis>
</token_notArabic>
<token_notArabic>888
  <analysis>888/NUM</analysis>
</token_notArabic>
<token_notArabic>.
  <analysis>./PUNC</analysis>
</token_notArabic>
```

Figure 5. Example of non-Arabic tokens identified by the BMA.

## 3. Adapting the BMA for Word Lookup

### 3.1 Rationale

While the BMA could be used as is for word lookup, reading the XML output can be difficult due to its complexity and length. Also, it may not be useful for certain Arabic natural language processing (NLP) tasks to provide all of the possible case inflections on a noun or adjective. Additionally, certain useful elements are embedded inside of the <pos> and <gloss> tags. Thus,

straightforward XML parsing is not sufficient for exploiting the richness of the output for a variety of Arabic NLP purposes, or specifically, for using the tool to look up the meanings of words. To address this need, we have adapted the BMA to take advantage of the richness of the BMA output while rendering that output easier to read and use for particular military tasks. Our work differs from a previous Arabic display system, Linear B (Calliston-Burch, 2005), in that we use the BMA to produce multiple glosses of input tokens; whereas, Linear B uses multiple statistical machine translation phrase tables.

### 3.2 Procedure

We originally developed the Buckwalter-based Lookup Tool (BBLT) for in-house use for machine translation (MT) developers and evaluators to rapidly see the meanings of Arabic strings that were not being translated by our Arabic-English MT engines (Voss et al., 2006). We then discovered that we could adapt the software to make it more human-readable and applicable to a variety of tasks. In order to provide a variety of simplified views to the output and make use of information embedded in tags, we built a layer of abstraction on top of the analyzer, in which each solution is captured in an object for further processing. In addition, we enhanced the original algorithm of the analyzer to extract unique meanings out of the original XML output.

#### 3.2.1 Providing an English Equivalent for Each Unique Meaning in the Output

Unique meanings are identified in the output by the lemmaID tag. Each solution containing the same lemmaID is collapsed into a single CollapsedSolution object for presentation purposes, which simplifies the display of the output. Further analysis of this output suggested that a deeper level of processing would help to display both active and passive glosses for verb solutions having the same lemmaID. For example, in this setup, "write" and "be written" are viewed as unique (for display purposes) yet they share the same lemmaID (table 2). If the basis for presenting a solution is the lemma ID, these two unique meanings would be displayed together in the same hypertext markup language (HTML) cell. In order to display two meanings that share the same lemmaID in separate cells, as the previous example demonstrates, we enhanced the original Buckwalter algorithm to include an additional tag for each solution, labeled "<root>" for lack of a better term. This tag includes the English gloss of the centrally analyzed piece, which is then used as the unique identifier rather than the lemmaID, thus allowing the active and passive verb forms to be displayed separately. The <root> tag is then later stripped out when displaying the original output.

Table 2. Example of two meanings that share the same lemmaID.

| Token | Vocalization | lemmaID | Gloss |
|---|---|---|---|
| كتب ktb | kataba | katab-u_1 | write |
| كتب ktb | kutiba | katab-u_1 | be written |

In all other cases, each root corresponds to a unique lemmaID, so the display of each solution is fundamentally lemma-based.

### 3.2.2 Further Processing of <pos> and <gloss> Tags

The <pos> and <gloss> tags contain the clitic and inflection information. This information is extracted out of these tags and saved in the solution object for later use in displaying the enhanced meanings. Subjects and objects of verbs, possessive pronouns on nouns, and other information conveyed thorough clitics are made viewable along with the unique English glosses for each analysis.

### 3.2.3 Words Not Found in the BMA Dictionary

In the original BMA algorithm, when a word is analyzed and the stem of the word is not found in the Buckwalter dictStems file, an <x_solution> is created and the word is glossed as "NOT_IN_LEXICON." In the "meanings" and "meanings with clitics & personal inflections" output views, these NOT_IN_LEXICON glosses tended to clutter up the output, so they have been removed from the display. If the output only contains <x_solution> entries, one cell with "NOT IN LEXICON" is displayed in the output.

### 3.2.4 Functional Description of the BBLT Web Application

The Lookup Tool Web application provides a simple input interface consisting of a text input box, labeled radio buttons and check boxes for selecting output options, and a button labeled "analyze" (figure 6). The user can copy and paste or type Arabic text into the text box, and select the desired variations of the output.

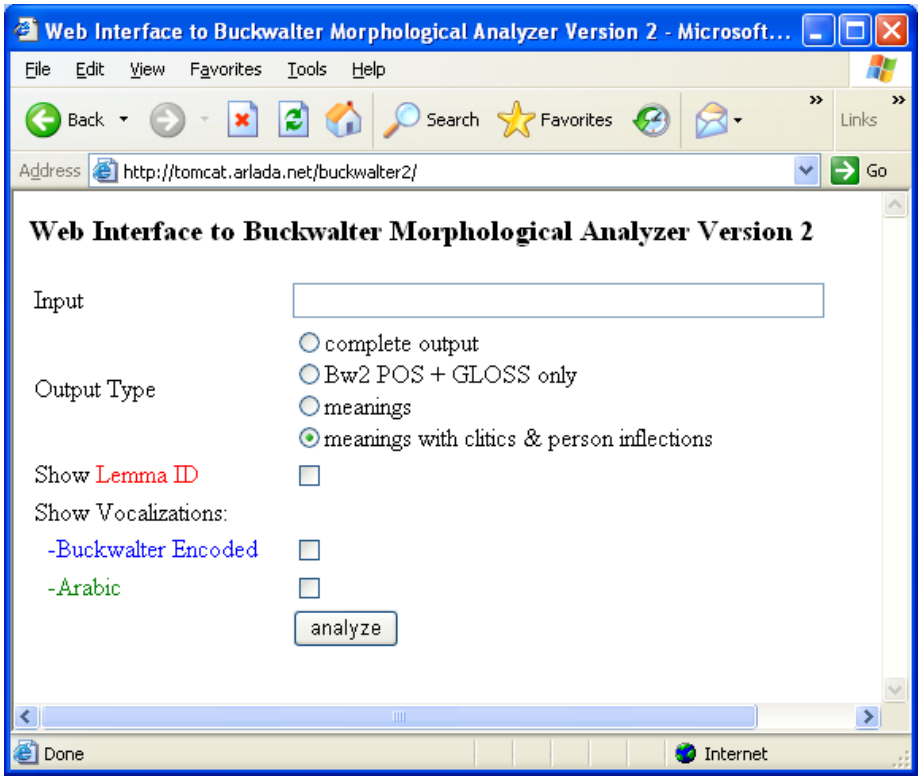


Figure 6. Lookup Tool Web application.

The Lookup Tool Web application has several options for Output Type:

- *Complete output:* This is the original output from the version 2 analyzer.
- *Bw2 POS + GLOSS only:* For each solution, the <pos> and <gloss> tags are extracted and their contents displayed. The output from each input token is separated by a newline. When developing the tool, this option was our first attempt at simplifying the output for human viewing. Although further development proceeded on the output display, we have retained this output type in the system as an option.
- *Meanings:* This output displays each unique meaning identified by the root tag in an HTML table, as explained previously. Each column is headed by the original Arabic token and each unique meaning is listed in the column in the order that it appears in the original output. As such, the rows from column to column have no correspondence other than indicating the order of appearance for each unique meaning. Since the view is designed to present English meanings, the order of the tokens is presented from left to right.
- *Meanings with clitics & person inflections:* This view adds glosses to each unique meaning for any clitics that are present in the analyzed form.

When the Output Type is set to "meanings" or "meanings with clitics & person inflections", the user may choose to see the associated lemmaID (displayed in red) and the Arabic vocalizations displayed in the Buckwalter Encoding (blue) or in Arabic script (green). These views may all be selected at the same time.

Figure 7 shows a sample output page with "meanings with clitics & person inflections" and Show LemmaID selected.

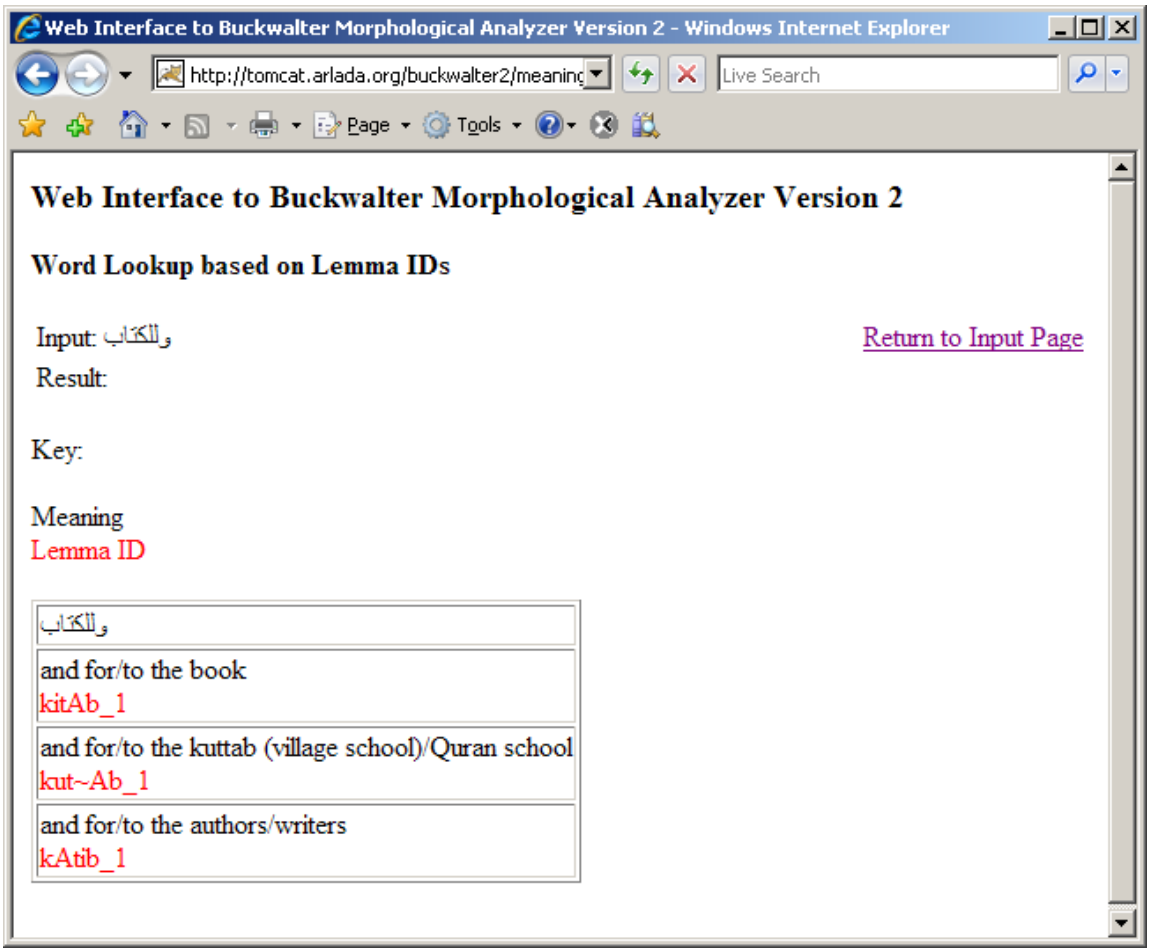


Figure 7. A sample output page with "meanings with clitics & person inflections" and Show LemmaID selected.

### 3.2.5 Design Description of the BBLT Web Application

The BBLT Web application is written as a Java Web application to be run in an Apache Tomcat Web server.* Figure 8 is a diagram that depicts how the various parts of the process interact with each other.† The application makes use of servlets and .jsp pages. There are two servlets, which are written as standalone servlets, extending HttpServlet. One standalone servlet, Buckwalter2Server, handles running the Buckwalter code as a socket-server process in the background and starting and stopping the process when necessary. The other standalone servlet, Buckwalter2Handler, handles requests from the input .jsp page, retrieves the raw output from the Buckwalter2Server, processes the raw output according to parameters passed from the input page, and redirects the output to the appropriate .jsp page for display. (See the diagram in figure 8.)

*The BBLT has also been adapted to work as a Web service. The Web service provides a simple function getOutput() that takes an input string in Arabic as the input. It provides as output the "meanings with clitics & personal inflections" view output in a list format, with the output for each input token being separated by a blank line.

†This formalism in this diagram has been adapted from the Unified Modeling Language (UML) sequence diagram formalism.

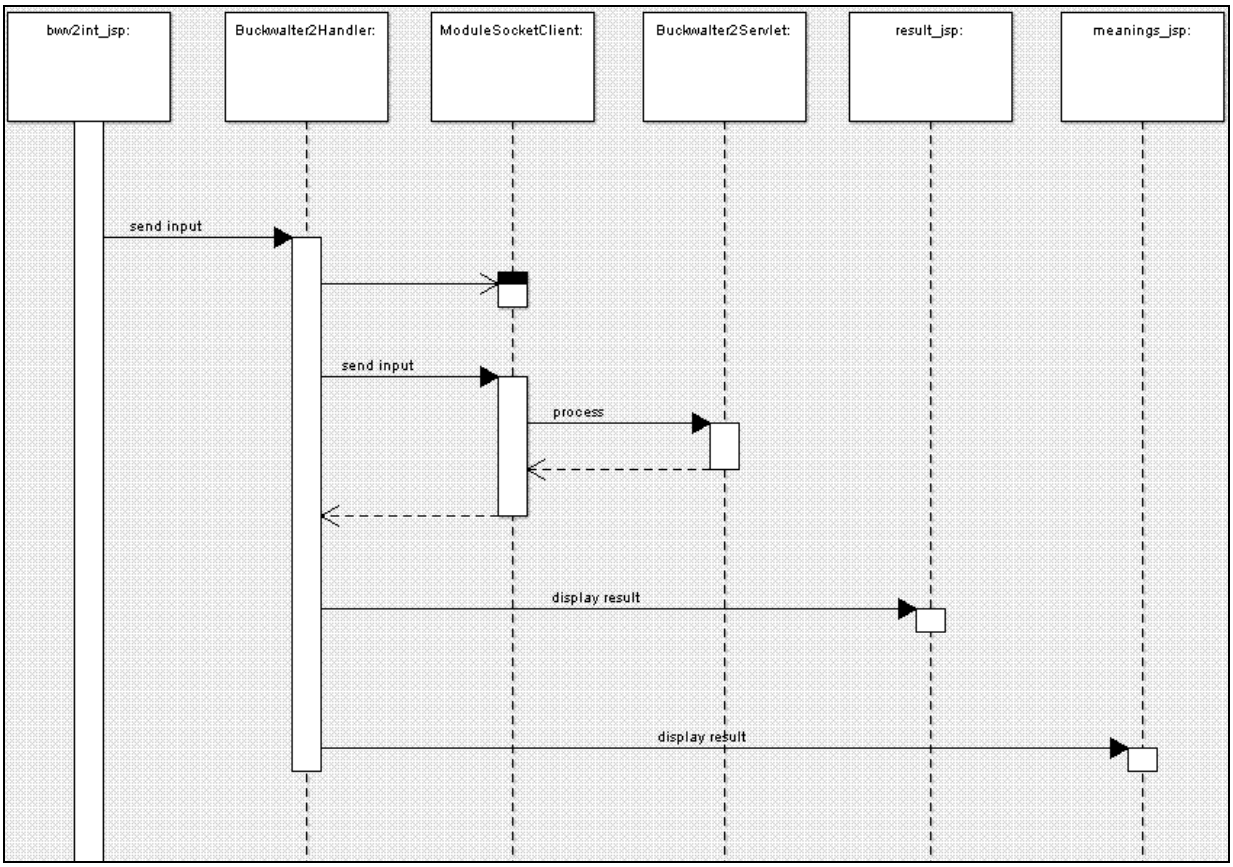


Figure 8. Diagram of the BBLT Web application process.

The bw2int_jsp Web page is the front-end Web page of a Java servlet class. The Web page uses a text field, radio buttons, and check boxes to collect input and output display parameters. Then it passes all of this form data to Buckwalter2Handler, via a GET request. First, Buckwalter2Handler creates a client object, ModuleSocketClient, and then it stores each of the form parameters in the session object as attributes, and calls the process() method of the ModuleSocketClient with the input data. The process() method connects to the socket server maintained by Buckwalter2Servlet. The port and host for the client to socket connection are stored as context parameters in the web.xml file for the Web application. The socket server maintains a running process, Aramorph_server.pl, which is a Perl script that carries out the morphological analysis via a ServerThread object, which can be destroyed by the Buckwalter2Servlet at any time. This script is an enhanced version of Buckwalter's original script. First, it is implemented as a socket server, which allows it to load the data only once when initializing, rather than for every request made to it. Second, the output is slightly modified to include a <root> tag, which stores the central piece of the analysis. This socket server returns the analysis output back to the client in the Buckwalter2Handler.

The Buckwalter2Servlet can be accessed via a browser at context path /buckwalter2/buckwalter2servlet. Access to this servlet is password protected, using BASIC

authentication. Using GET rather than POST for sending requests to the servlet, the socket server can be stopped or restarted at any time by appending the parameter "cmd" to the uniform resource locator (URL) and setting it to "start" or "stop." Also, the socket server can be restarted on any port by appending "port=<portnumber>" as seen in figure 9.

```
/buckwalter2/buckwalter2servlet/?cmd=start
/buckwalter2/buckwalter2servlet/?cmd=stop
/buckwalter2/buckwalter2servlet/?port=<portnumber>
```

Figure 9. An example of starting and stopping Buckwalter2Servlet.

Once the output is back in Buckwalter2Handler, it is further processed based on the original input parameters (which have been stored in the HttpSession object.) and displayed with one of two .jsp pages: result.jsp or meanings.jsp. The result.jsp page provides a text area for the output and is used when options "complete output" or "BW2 POS+GLOSS only" had been selected on the input page. The meanings.jsp page provides the output data in an HTML table when the options "meanings" or "meanings with clitics & person inflections" had been selected during input.

### 3.2.6 Class Diagram of the Abstraction Layer

Figure 10, a Unified Modeling Language (UML) class diagram‡, shows the classes that are used to create the layer of abstraction on top of the original output of the analyzer. It also includes the classes that are used to create a Java Web application to process input requests and display the results.

At the top left are the classes that represent the input and output pages of the Web application: bwv2int_jsp (input), meanings_jsp (output), and result_jsp (output). At the top right of the diagram are the classes that represent the backend processing components of the application: Buckwalter2Handler and Buckwalter2Server. All of the Web application classes inherit from HttpServlet.

At the bottom left of the diagram are classes that handle creating a socket connection and a client to this connection. MTModule is the outer class, which can be used for any type of underlying input-output process. For example, the meanings_jsp page uses this class for running a process to convert the Buckwalter-encoded vocalization back into Arabic text for display purposes. The ServerThread class spawns a new thread to run an underlying process. Buckwalter2Server uses this class to start and stop the underlying "Aramorph_server.pl" process, which is the original Perl process providing the Buckwalter analysis. ModuleSocketClient provides a client that can connect to a socket. Buckwalter2Handler uses this class to connect to the "Aramorph_server.pl" process.

‡UML provides a means for modeling software and documenting its structure and development through the software lifecycle. A class diagram displays class information, how the classes are related to one another (e.g., association, aggregation, composition, inheritance), and multiplicity of the relationships (e.g., many-to-one). The diagram used here demonstrates the relationships between the classes. (Wikipedia, 6 April 2009)

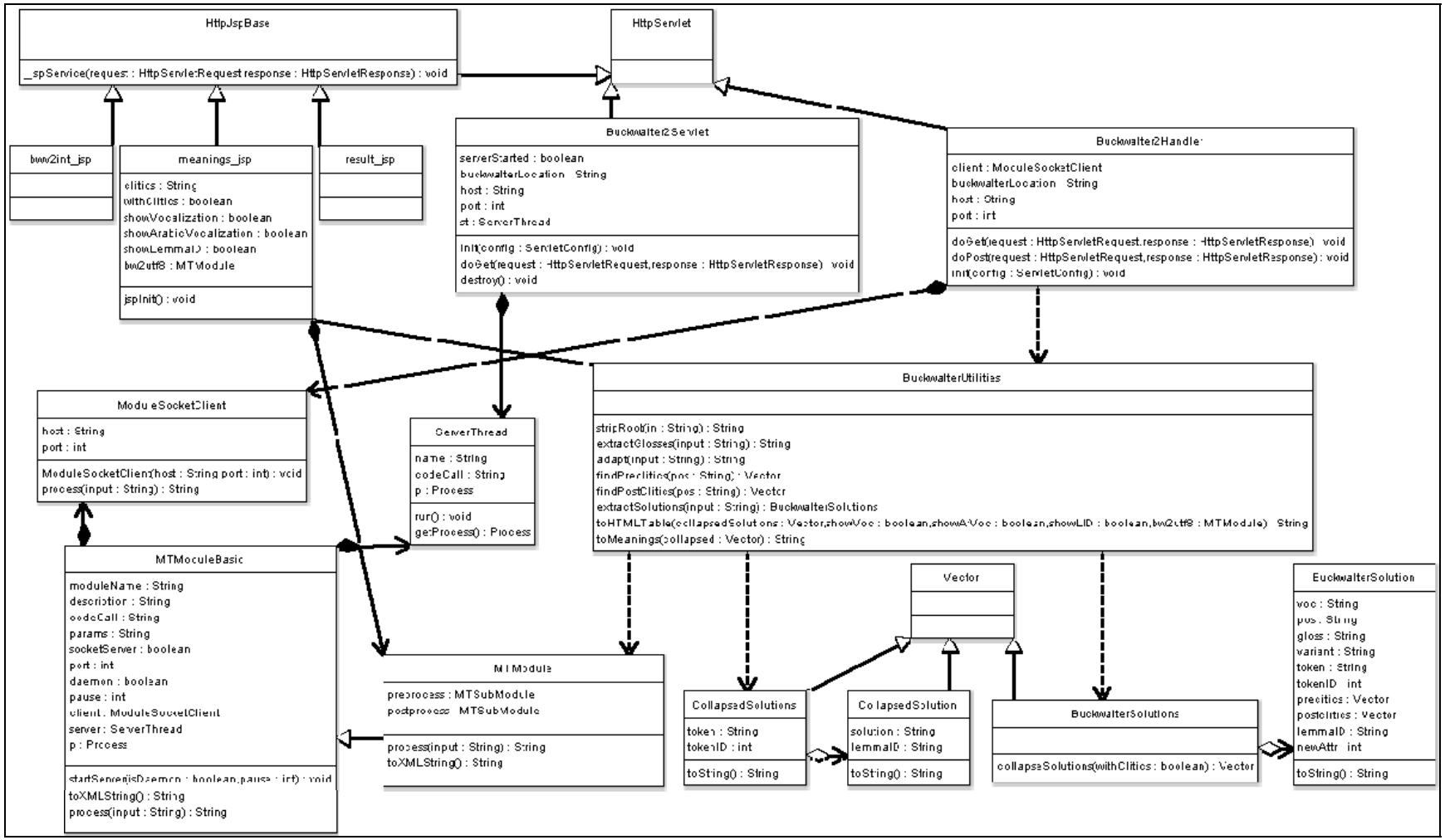


Figure 10. A UML class diagram.

# 4. Additional Features

After the initial development of BBLT with the features described previously, the tool was put to use by research scientists working with Arabic translations. As a result of this initial use of the tool, some additional ideas for useful features were determined and implemented, which are described in the following sections.

## 4.1 Option "Show POS"

Since part of speech (POS) information is available in the BuckwalterSolutions object, it was relatively easy to add the option to display the POS information in the "meanings" and "meanings with clitics & person inflections" outputs.

Figure 11 shows the new input page with a "Show POS" checkbox.

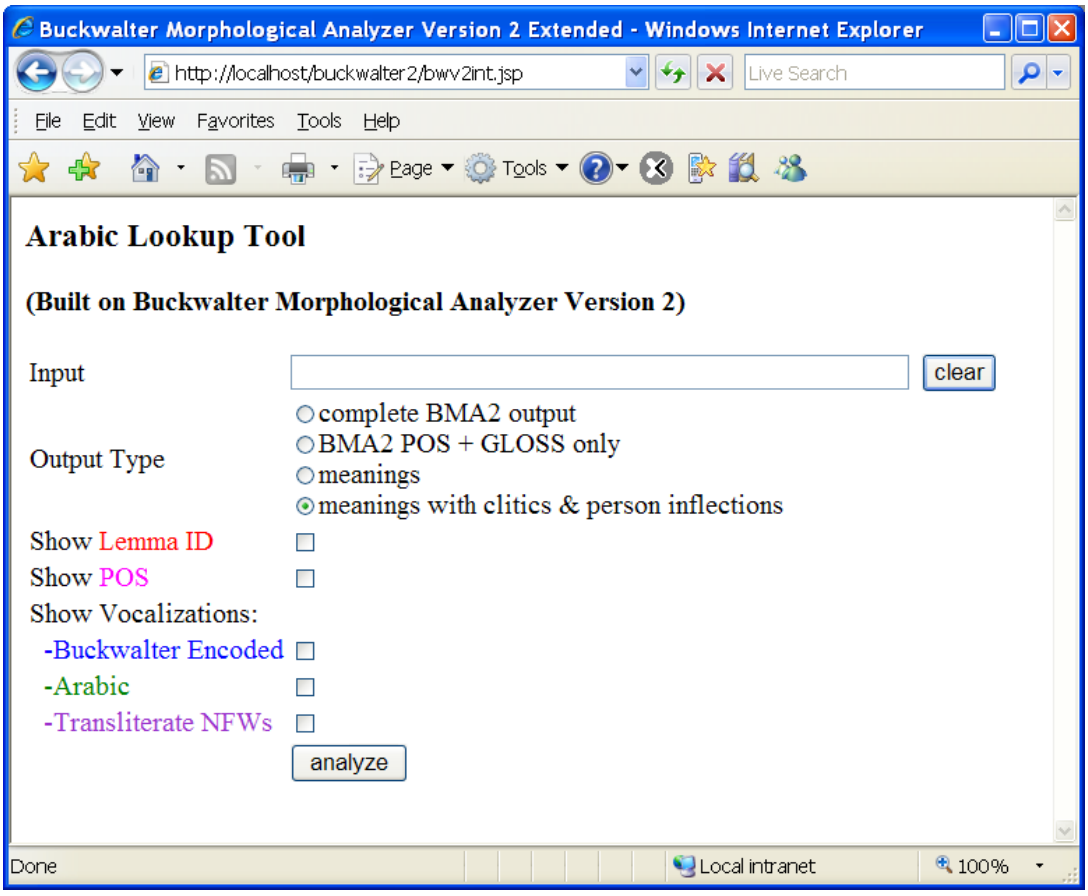


Figure 11. The new BBLT input page with the "Show POS" and "Transliterate NFWs" checkboxes.

### 4.2 Transliteration for "Not Found Words" (NFWs)

Figure 11 also shows a new "Transliterate NFWs" checkbox, which is used to transliterate words that are now found in the dictionary.

Words that are not found in the Buckwalter dictionary are glossed as "NOT_IN_LEXICON" in the original morphological analyzer output. These tokens are analyzed as if they were words with possible clitics attached. The analyzer tries to chunk off any possible clitics and then, if the stem is not found, it glosses the token as "NOT_IN_LEXICON." Figure 12 shows the BMA output of a token whose stem is not found in the Buckwalter dictionary.

```
<token_Arabic>بالنيث
 <variant>bAlnyv
  <x_solution>
   <voc>bAlnyv</voc>
   <pos>bAlnyv/NOUN_PROP</pos>
   <gloss>NOT_IN_LEXICON</gloss>
  </x_solution>
  <x_solution>
   <voc>biAlnyv</voc>
   <pos>bi/PREP+Alnyv/NOUN_PROP</pos>
   <gloss>by/with + NOT_IN_LEXICON</gloss>
  </x_solution>
  <x_solution>
   <voc>biAlnyv</voc>
   <pos>bi/PREP+Al/DET+nyv/NOUN_PROP</pos>
   <gloss>with/by + the + NOT_IN_LEXICON</gloss>
  </x_solution>
 </variant>
</token_Arabic>
```

Figure 12. BMA output of a token whose stem is not found in the Buckwalter dictionary.

Since it is likely that a word that is not found in the dictionary could be a proper noun, the BMA assigns the "NOUN_PROP" POS to these stems. It was thought that if a user could see a transliteration of these stems, they may be able to determine the proper noun that they represent. Therefore, we added an option to show NFWs transliterated to the output options. To do this, we wrote a simple Perl script to perform the transliteration. The Perl script loads a one-to-one list of correspondences between Arabic and Roman characters from an independent correspondence file, which it uses to transliterate Arabic characters into Roman characters. In this way, an administrator can specify exactly how the transliteration should proceed by modifying the correspondence file, with the restriction that one Arabic character corresponds to one Roman character. This script is then encapsulated in an MTModule inside of the meanings.jsp class to transliterate the stem.

Figure 13 shows the output of a particular phrase containing “not found words” with “transliterate” unselected and figure 14 show the same output with “transliterate” selected.

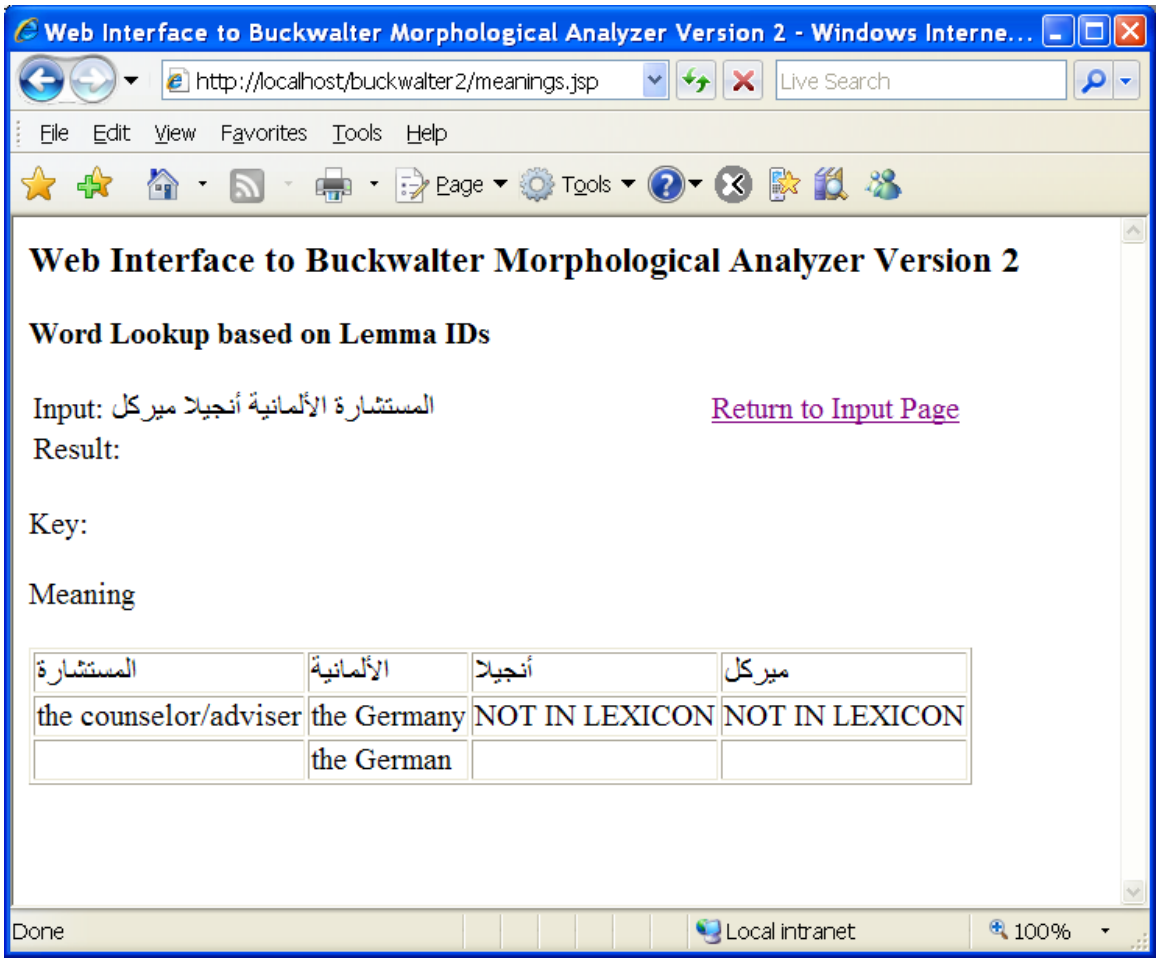


Figure 13. Output example of a phrase showing NFWs with transliterate unselected.

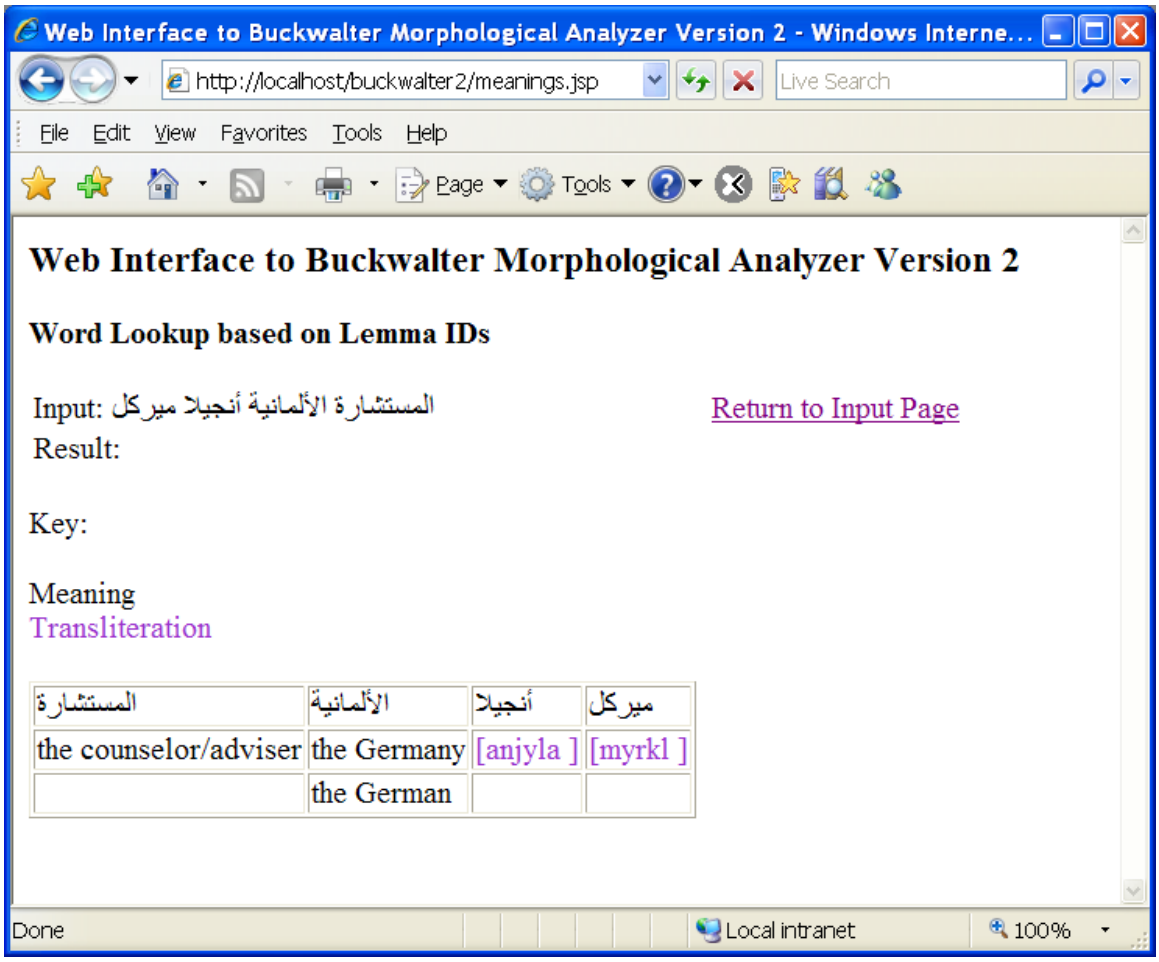


Figure 14. Output example of a phrase showing NFWs with transliterate selected.

### 4.3 Generation Module for Inflecting Glosses of English Verb Forms

The "meanings with clitics & personal inflections" output view was designed with the idea that an English speaker could read across the columns and get a gist of the meaning of a sentence by selecting out the most likely gloss from the list of glosses for each token. The glosses for the clitics and pronouns augment the gisting ability of the user. However, verbs were only displayed with the English root form and not inflected in English according to the tense/aspect and person of the Arabic token. For example, " كتب k-t-b" is glossed as "he/it write." In Arabic, this verb is in the perfect tense, which can be glossed in English with the past tense. So to make this gloss more accurate and readable, it should inflect the root "write" for the past tense, as in "he/it wrote."

Thus, we created a module to do this—a simple Perl script that handles English inflections. The module takes a root plus a POS plus any features relevant to that POS as input. In the case of verbs, a tense/aspect feature plus a person and number feature are possible in the input string. The script converts this input string to the appropriately inflected form, as shown in figure 15.

```
write+v+past        ->      wrote
write+v+pres+3+sg   ->      writes
write+v+ing         ->      writing
write+v+en          ->      written
```

Figure 15. An input string converting using the appropriate inflected form.

Currently, the script only works on verbs, but it can be extended to handle noun plurals as well. Since the BMA already glosses plurals as plurals, the noun plural inflection feature was not needed at this stage of the development of the tool.

Figure 16 shows an example output with the gloss for the second word fully inflected: "(she/it) considers/regards/believes."

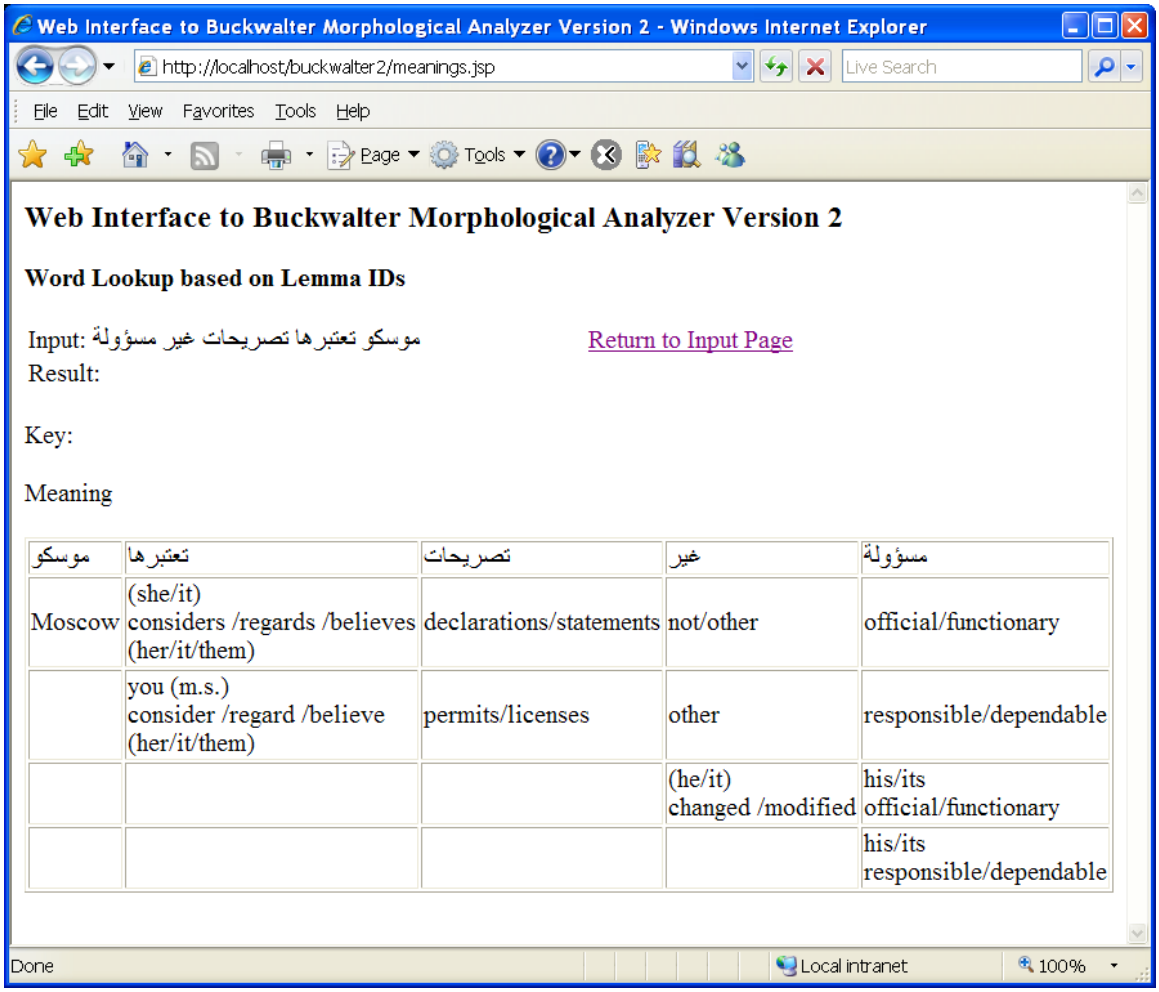


Figure 16. An example output with the gloss for the second word fully inflected: "(she/it) considers/regards/believes."

## 5. Conclusions

The creation of the BBLT serves as an example of how we can extend the functionality of an existing translation resource to apply to a language of interest to the Army. In the case of existing morphological analyzers, adapting the tool to perform word lookups allows the user to exploit the richness of the output of morphological analyzers while rendering that output more usable. This report has documented the adaptation of one notable tool, the BMA, version 2.0. The resulting BBLT is implemented as a Web application on top of the existing BMA algorithms and dictionaries. The variety of output types and options allows the user to see the output in different ways. We expect that using the BBLT will eventually improve the speed and accuracy with which humans perform document triage.

## 6. References


Buckwalter, T. Arabic Morphology Analysis. 2002. www.qamus.org/morphology.htm (accessed June 2005).

Buckwalter, T. *Buckwalter Arabic Morphological Analyzer (BAMA), Version 2.0*, LDC Catalog number LDC2004L02, 15 Dec 2004, www.ldc.upenn.edu/Catalog (accessed April 2006).

Calliston-Burch, C. Linear B System Description for the 2005 NIST MT Evaluation Exercise. *NIST MT Evaluation Workshop*, 2005.

Voss, C.; Micher, J.; Laoudi, J.; Tate, C. Ongoing Machine Translation Evaluation at ARL. *Proceedings of the NIST Machine Translation Workshop*, Washington, DC, 2006.

Wikipedia. Clitic page. Last modified: 12 April 2009. http://en.wikipedia.org/wiki/Enclitic (accessed April 2009).

Wikipedia. Unified Modeling Language page. Last modified: 6 April 2009. http://en.wikipedia.org/wiki/Unified_Modeling_Language (accessed June 2007).

## List of Symbols, Abbreviations, and Acronyms

| | |
|---|---|
| AMTA | Association for Machine Translation in the Americas |
| BBLT | Buckwalter-based Lookup Tool |
| BMA | Buckwalter Arabic Morphological Analyzer |
| DoD | Department of Defense |
| HTML | hypertext markup language |
| MT | machine translation |
| NFWs | not found words |
| NLP | natural language processing |
| POS | part of speech |
| UML | Unified Modeling Language |
| URL | uniform resource locator |
| XML | extensible markup language |

| NO. OF COPIES | ORGANIZATION |
|---|---|
| 1 ELEC | ADMNSTR<br>DEFNS TECHL INFO CTR<br>ATTN DTIC OCP<br>8725 JOHN J KINGMAN RD STE 0944<br>FT BELVOIR VA 22060-6218 |
| 1 | DARPA<br>ATTN IXO S WELBY<br>3701 N FAIRFAX DR<br>ARLINGTON VA 22203-1714 |
| 1 CD | OFC OF THE SECY OF DEFNS<br>ATTN ODDRE (R&AT)<br>THE PENTAGON<br>WASHINGTON DC 20301-3080 |
| 1 | US ARMY RSRCH DEV AND ENGRG CMND<br>ARMAMENT RSRCH DEV AND ENGRG CTR<br>ARMAMENT ENGRG AND TECHNLGY CTR<br>ATTN AMSRD AAR AEF T J MATTS<br>BLDG 305<br>ABERDEEN PROVING GROUND MD 21005-5001 |
| 1 | PM TIMS, PROFILER (MMS-P)<br>AN/TMQ-52<br>ATTN B GRIFFIES<br>BUILDING 563<br>FT MONMOUTH NJ 07703 |
| 1 | US ARMY INFO SYS ENGRG CMND<br>ATTN AMSEL IE TD A RIVERA<br>FT HUACHUCA AZ 85613-5300 |
| 1 | COMMANDER<br>US ARMY RDECOM<br>ATTN AMSRD AMR<br>W C MCCORKLE<br>5400 FOWLER RD<br>REDSTONE ARSENAL AL 35898-5000 |
| 1 | US GOVERNMENT PRINT OFF<br>DEPOSITORY RECEIVING SECTION<br>ATTN MAIL STOP IDAD J TATE<br>732 NORTH CAPITOL ST NW<br>WASHINGTON DC 20402 |

| NO. OF COPIES | ORGANIZATION |
|---|---|
| 1 | US ARMY RSRCH LAB<br>ATTN RDRL CIM G T LANDFRIED<br>BLDG 4600<br>ABERDEEN PROVING GROUND MD 21005-5066 |
| 12 | US ARMY RSRCH LAB<br>ATTN IMNE ALC HRR<br>MAIL & RECORDS MGMT<br>ATTN RDRL CII B BROOME<br>ATTN RDRL CII T C VOSS<br>ATTN RDRL CII T J MICHER<br>(5 COPIES)<br>ATTN RDRL CII T S LAROCCA<br>ATTN RDRL CII T V M HOLLAND<br>ATTN RDRL CIM L TECHL LIB<br>ATTN RDRL CIM P TECHL PUB<br>ADELPHI MD 20783-1197 |
| TOTAL: | 21 (1, ELEC, 1 CDs, 19 HCs) |