Lexicon-Driven Approach to the Recognition of Arabic Named Entities

Jack Halpern (春遍雀來)

The CJK Dictionary Institute (日中韓辭典研究所)

34-14, 2-chome, Tohoku, Niiza-shi, Saitama 352-0001, Japan

jack@cjki.org

Abstract

Various factors contribute to the difficulties in processing Arabic personal names and named entities, posing special challenges to developers of NLP applications in the areas of named entity recognition (NER), machine translation (MT), morphological analysis (MA) and information retrieval (IR). These include the complexity of the Arabic orthography, the high level of orthographical and morphological ambiguity, the multitude of highly irregular romanization systems, and the vast number of romanized variants that are difficult to detect and disambiguate.

This paper focuses on the orthographic variation of Arabic personal names, with special emphasis on the ambiguity resulting from transcribing names into the Roman script (romanization). It describes the techniques used to compile the Database of Arabic Names (DAN), a large-scale lexical resource containing millions of Arabic names and their variants in both romanized and fully vocalized Arabic, and argues that linguistic knowledge in the form of a rule-driven lexicon can enhance the accuracy of statistical methods to achieve high accuracy in the recognition of Arabic names.

1 Concepts and Definitions

Much confusion surrounds the terms *transliteration* and *transcription*, with the former often misleadingly used in the sense of the latter even in academic papers (AbdulJaleel and Larkey, 2003). To discuss these concepts in an unambiguous manner it is necessary to understand these and related terms correctly.

Romanization is the representation of a language written in a non-Roman script using the Roman alphabet. This includes both transliteration and various kinds of transcription. There are various official systems for romanizing Arabic.

Transcription is a representation of the source script of a language in the target script in a manner that reflects the pronunciation of the original, often ignoring graphemic correspondence. This includes the following subcategories:

1. A *phonetic transcription*, enclosed in square brackets, represents the actual speech sounds, including allophones. The best known systems are IPA and SAMPA. For example, محمد is transcribed as [muħëmmëd].

2. A *phonemic transcription*, enclosed in slashes, represents the phonemes of the source language (ignoring allophones), ideally on a one-to-one basis. For example, محمد is transcribed as /muHammad/, in which a represents the phoneme /a/, rather than the phone [ë]. Some well known systems include ICS *(*Intelligence Community Standard) and ALC-LA (American Library Association-Library of Congress romanization standard) (Library and Archives Canada, 2006).

3. A *popular transcription*, indicated by italics, is a conventionalized orthography that roughly represents pronunciation. For example, محمد is transcribed in over 100 ways, such as *Mohammed*, *Mohamed*, *Moohammad*, *Moohamad*, *Mohammad*, *Mohamad*, etc.

Transliteration, enclosed in back slashes, is a representation of the script of a source language by using the characters of another script. Ideally, it unambiguously represents the graphemes, rather than the phonemes, of the source language. For example, محمد is transliterated as \mHmd\, in which each Arabic letter is unambiguously represented by one Roman letter, enabling round-trip conversion. A transliteration system widely used in NLP applications is the excellent Buckwalter system (Beesley, 2003). Many aca-

demic papers misleadingly use the term *transliteration* when what they actually mean is transcription (Halpern, 2007).

To summarize, the name محمد is transliterated as \mHm~d\, phonetically transcribed as [muħëmmëd], phonemically transcribed as /muHammad/, and romanized in various popular transcriptions such as *Mohammed, Muhammad* and *Mohamad*, among many others.

In this paper, the adjectives *Arab* and *Arabic*, especially in the context of *Arab name* and *Arabic name*, are not interchangeable. *Arab* refers to names of Arabs (as opposed to names of non-Arabs) regardless of whether they are written in the Arabic or the Roman script, whereas *Arabic* is more general and refers to Arab names written in the Roman script or any name (Arab and non-Arab) written in the Arabic script.

2   Why the Arabic script is ambiguous

The Arabic script is highly ambiguous. A distinguishing feature of written Arabic is that words are represented as a string of consonants with little or no indication of vowels, referred to as *unvocalized Arabic*. For example, the string مو can theoretically represent 40 consonant-vowel permutations, such as *mawa, mawwa, mawi, mawwi*, etc. Even fully vocalized Arabic is not phonemic because of the many one-to-many grapheme-to-phoneme ambiguities (Halpern, 2007 and 2008). The principal factors contributing to orthographical ambiguity are summarized below.

1. The omission of short vowels: e.g., the unvocalized كاتب can represent the following seven wordforms: كَاتِب /kaatib/, كَاتَبَ /kaataba/, كَاتِبِن /kaatibin/, كَاتِبٌ /kaatibun/, كَاتِبَ /kaatiba/, كَاتِبِ /kaatibi/ and كَاتِبُ /kaatibu/.
2. Long /aa/ can be represented various ways, i.e., ا, ى, آ, or zero, as illustrated by سوريا, الوسطى and آسيا, while short /a/ can be represented by zero, ه, ة, or even ا (especially in loanwords), as illustrated in the various Arabic variants for Alexandria: الإسكندريه, الإسكندريا, and الإسكندرية.
3. Consonant gemination is ambiguous since *shadda* is normally omitted, e.g., the unvocalized محمد /muHammad/ has no *shadda* (vocalized as مُحَمَّد).

4. The omission of nunation diacritics for case endings, e.g., in شكرا /shukran/ (vocalized شُكْرًا), the *fatHatayn* is not written.
5. Alternation between ي (*yaa'*) and ى (*'alif maqSuura*), especially in Egypt, causes confusion. For example, ابو ظبي 'Abu Dhabi' is often written as ابو ظبى.
6. Alternation between ه (*haa'*) and ة (*taa' marbuuTa*), resulting in names like عودة /ʕuuda/ being also spelt as عوده.
7. The rules for determining the *hamza* seat are of notorious complexity.
8. Phonological alternation processes such as assimilation modify the phonetic realization, e.g. الرجل الطويل is realized as /'arrajulu-TTawiilu/.
9. Compound names are often written either solid or open, e.g., /ʕabd-urrahiim/ is written either عبد الرحيم or عبدالرحيم. This is true of various other common name elements like أبو /'abuu/ and بن /bin/,
10. Long vowels are often neutralized, a fact which is mostly ignored in the literature. Even such common words as أنا /'ana/ 'I' and هٰذا /haadha/ 'this' are often incorrectly transcribed as *'anaa* and *haadhaa*.

These ambiguities result in names such as عودة /ʕuuda/ also being spelt as عوده, and variation due to common name elements such as عبد /ʕabd/ and أبو /'abuu/ being detached from or attached to the rest of the name. A combination of these factors lead to a name like عبد العزيز /ʕabd-'al-ʕaziiz/ having eight variants in Arabic, including عبدالعزيز, عبد العزيز, and عبدالعزيز. As a result, it is difficult to identify such strings as variants of the same underlying name, leading to lower recall in named entity extraction and recognition.

3   Why romanized Arabic is ambiguous

Arabic also has a high level of romanization ambiguity. That is, an Arabic name can be romanized in a multitude of ways. One reason for this is that many official and unofficial romanization systems are used to transcribe Arabic sounds in a bewildering number of ways. For example, شولوخ /shuuluukh/ is transcribed in several official systems as follows: *shwlwkh* in the ALA-LC system, *Shulukh* in ICS (Intelligence Community Standard), *šūlūḫ* in DIN, *Shūlūkh* in BGN/PCGN and *ʃuːluːx* in IPA. Another reason is that many Arabic speech sounds do not exist in West European languages. For example, ظ is variously romanized as *t, z, th,* or *dh,* and ق is

romanized in some unusual ways as shown in the table below.

Roman	Example	Frequency
q	Qaddafi	00077900
g	Gaddafi	00219000
gh	Ghaddafi	00013100
k	Kaddafi	00068300
kh	Khaddafi	00007380
c	Caddafi	00000034
j	Jaddafi	00000031

Table 1: Examples of ق romanizations

The official romanization systems, which are used rather infrequently, are just a fraction of the plethora of romanizations which for lack of a better name we shall lump together under the label "popular transcriptions". These transcriptions are often inconsistent, irregular and unpredictable, sometimes leading to hundreds or more than a thousand variants for a single name. For example, the names محمد /muHammad/ and القذافي /al-qadhaafii/ are touted as having more than 100 romanized variants. As impressive as this number may seem, it pales in comparison to names like عبدالرحيم /ʿabdurrahiim/ and عبدالرزاق /ʿabdurrazzaaq/, which have over 1100 variants. Those cases are extreme, but names with several dozen variants are very common. For example, the popular name محمود /maHmuud/ has 69 variants, the most frequent of which are shown below along with their frequency of occurrence on the web.

Variant	Frequency
Mahmoud	0020400000
Mahmud	0005770000
Mahmood	0004050000
Mahmut	0003780000
Mehmood	0000685000
Mahmod	0000138000
Machmut	0000006640
Machmud	0000108000
Mehmud	0000082400
Mahmoed	0000052100
Mechmod	0000048000
Makhmud	0000042700

Table 2: Common romanized variants of محمود

The various ambiguity factors described above result in an immense number of romanized variants.

4　　　Lexicon-Driven Approach

An important motivation for using statistical methods in NLP applications has been the poor availability and the high cost of constructing large-scale lexical databases, but statistical methods by themselves are inadequate for dealing with the ambiguities of the Arabic script, especially because of the difficulties in algorithmically recognizing and processing the immense number of romanized name variants. Procedures such as entity recognition and disambiguation cannot be based on probabilistic methods such as using bigram statistics and algorithmic methods alone (Farghaly, 2004) To enhance these methods we have adopted a **lexicon-driven approach**, which exploits a large database of Arab personal names built with the aid of an Orthographical Rule Base compiled by statistical analysis of a large-scale name corpus and by an in-depth linguistic analysis of how Arabic names map to their romanized variants.

The **Database of Arab Names** (DAN), which currently contains over five million entries, is a rapidly expanding repository of romanized personal names and their variants mapped to the original Arabic script. In addition, a database referred to as DANA covers several hundred thousand Arabic-script variants in fully vocalized Arabic (see below). DAN and DANA combined include various attributes useful for NLP applications, such as web occurrence statistics, gender and type codes, canonical forms, variants mapped to their canonical forms, common Arabic spelling errors, and several official romanizations. A snippet of DAN with a subset of its attributes is shown in Table 3.

Variant	Arabic	Type[1]	Gender	Frequency
Mohamed	مُحَمَّد	B	M	06642415
Muhammad	مُحَمَّد	B	M	05300000
Mohammed	مُحَمَّد	B	M	05030000
Mohammad	مُحَمَّد	B	M	03410000
Muhammed	مُحَمَّد	B	M	02342308
Mohamad	مُحَمَّد	B	M	01140395
plus 164 variants				
Al-Qubaisi	اَلْقُبَيْسِيّ	S	N	00007570

---

[1] S: surname, G: given name B: surname and given name

Al-Kubaysi	الْقُبَيْسِيّ	S	N	00001420
Al-Qubaysi	الْقُبَيْسِيّ	S	N	00000267
Al-Koubaisi	الْقُبَيْسِيّ	S	N	00000205
Al-Qubeisi	الْقُبَيْسِيّ	S	N	00000195
Al Qubaisy	الْقُبَيْسِيّ	S	N	00000077
plus 214 variants				
Yasmina	يَاسْمِينَة	G	F	00735141
Yasmeena	يَاسْمِينَة	G	F	00014170
Yasmeenah	يَاسْمِينَة	G	F	00004390
Yasmineh	يَاسْمِينَة	G	F	00002300
Yasmiina	يَاسْمِينَة	G	F	00002236
Yasminah	يَاسْمِينَة	G	F	00000899
plus 43 variants				

Table 3: Snippet of DAN with some attributes

Using algorithmic or statistical methods to identify similar spellings such as *Mahmoud* and *Mehmood* works well, but it is more difficult to match more dissimilar spellings like *Awdah* and *Udeh*, variants of عوده /ʕuuda/. Even if these could be identified by statistical pattern matching, ultimately statistical methods by themselves are inadequate (Kay, 2004) for NER and morphological analysis. As pointed our by Maloney and Niv (1998), morphological analysis is crucially dependent on lexical data.

Combining statistical methods with a lexicon-driven approach often ensures a higher probability of an exact match. Another reason for the limitation of statistical methods is the lack of large-scale bilingual corpora of Arabic names that would be necessary for training statistical generative models such as HMMs (Arbabi et al, 1994).

5 Compilation Methods

DAN was compiled by a team of software engineers and native speaker editors trained in Arabic phonology using the techniques described below.

5.1 Name Corpora

The **Name Component Corpus** (NCC) was created and is maintained by collecting massive quantities of Arabic name data consisting of name components and their variants derived from a large variety of sources, including web-sites, corpora, books, phone directories, dictionaries, encyclopedias and university rosters. As new data is collected, the corpus is augmented by rule-based generation to build an expanded corpus. This expanded corpus currently contains over 50 million entries, mostly unvalidated, and serves as the raw data for building DAN using the techniques described below.

Name components, or simply *names* in this paper, refers to a given name or surname, not to full names; e.g., given names such as محمد /muHammad/ and their romanized variants (over 100 in this case) and surnames like علي /ʕaliyy/ are included in NCC, but full names (of which there are hundreds of millions) such as محمد علي /muHammad ʕaliyy/ have been excluded. The NCC is regularly sanitized, maintained and expanded.

On the other hand, an unsanitized corpus of raw full name data, referred to as **Full Name Corpus** (FNC), is maintained and expanded to serve as a source of data for NCC, for reference and for some validation and proofing tasks. The NCC, not FNC, serves as the central repository for such tasks as name vocalization, romanization, normalization and validation. It is estimated that the total number of full names and their potential variants easily exceeds one billion. If these had been combined with name components to form a single corpus, it would have resulted in a massive amount of repetition and lead to inefficient database management and maintenance (since one name like محمود can appear tens of thousands of times) while bringing no specific benefits.

5.2 Linguistic Tools

A suite of Arabic name processing tools was developed and is constantly maintained for performing the tasks described below.

1. automatic Arabic variant generation
2. automatic Roman variant generation
3. analysis and validation of vocalization integrity
4. input and proofreading interfaces
5. automatic transliteration and transcription
6. acquisition of web occurrence statistics
7. large-scale collection of names from the web and other sources
8. database compilation, management and code conversion
9. inference engine for processing the Orthographic Rule Base.

To develop some of these tools requires ongoing in-depth analysis of such phenomena as how the Arabic orthography maps to the various romanization systems, the variation patterns of both Arabic and romanized names, as well as extensive analysis of the orthographic behavior of dozens of *name elements*, such as أبو و/'abuu/, بن بـ /bin/ and عـ بد /ʿabd/.

5.3 Orthographic Rule Base

One of the central components of the system is the **Orthographic Rule Base** (ORB), based on the analysis of hundreds of thousands of attested variants, and the accompanying inference engine. This consists of a romanization table with hundreds of one-to-many grapheme-to-phoneme mappings, many of which are restricted by contextual constraints. The table below shows some typical one-to-many mapping.

ظ	/T/	d dh t z th
ق	/q/	q k g kh c ck j gh
ك	/k/	k q c ck
ش	/sh/	sh sch ch
ج	/j/	j g dj gh
	/u/	u o uo oo uu
و	/uu/	uu ou oo u o ow oe uw

Table 4: One-to-Many Mappings

Various factors needed to be considered in building the ORB:

1. Dialectical allophones may affect romanization, so that ج can be represented by *g* (Egypt) and *dj*, in addition to the more common *j*.
2. Variation based on the differences in representing the same phone by various graphemes in various European languages; e.g., ش /sh/ is represented by *sch* in German and *ch* in French orthography.
3. Many Arabic speech sounds, such as ق ,ظ and خ (Table 3 above), cannot be easily represented in the West European languages, leading to an explosion of romanized variants.
4. The inference engine is used to expand vocalized Arabic names. For example, حُسَيْن is used to generate *Housseine* (one of over 100 other variants), which is later confirmed to be valid with a web occurrence of 220,000.
5. Two types of constraint and expansion rules are applied. The first is specific to particular mappings, e.g., the rule **double intervocalic s** is applied to generate *Hussain* from *Husain* (even though حُسَيْن has no *shadda*). In addition there are universal constraint/expansion

rules that apply across the board, e.g., **un-double all double letters** is applied to all double letters generated from *shadda*.
6. The validity of hypothetical mappings was tested by searching the web for the co-occurrence of names in original Arabic script along with generated variants. The co-occurrence statistics were used to determine which mappings yield poor results and thus need to be removed.

5.4 Vocalization and Sanitization

A key feature of the system is that every Arabic name is normalized and vocalized to produce a database of error-free, fully sanitized Arabic canonical forms. At this stage variants and spelling errors such as عبدالعزيز ,عبدلعزيز ,عبدإلعزيز for العزيز عبد /ʿabd 'al-ʿaziiz/ are removed, to be later included in a separate database described below (DANA). The vocalization of the unvocalized Arabic is performed by team of editors with the aid of tools and interfaces designed to achieve maximum efficiency.

Stage	Arabic	Description
1	عبدالغنى \EbdAlgnY\	Unvocalized data extracted from source
2	عبد الغنى \Ebd AlgnY\	Normalize algorithmically by adding space
3	عَبْدُ ٱلغنى \Eabodu {lognY\	Automatic vocalization
4	عَبْدُ ٱلْغَنىّ \Eabodu {loganiY\	Manual vocalization
5	عَبْدُ ٱلْغَنيّ \Eabodu {loganiy\	corrected after sanity checking

Table 5: Vocalization Process

The data is analyzed by various sanity check routines to ensure data integrity, to eliminate false positives, and to remove undesirable items like real and pseudo-duplicates, linguistically invalid vocalizations, electronic garbage, non-names, and foreign (non-Arab) names, and is subjected to repeated cycles of manual vocalization and sanity checking until 100% linguistically valid, accurate vocalization is achieved. For example, at the end of the process the name عبد الرحمن /ʿabdurraHmaan/ is vocalized as ٱلرَّحْمٰن عَبْدُ, which includes such rarely used diacritics as dagger 'alif and *wasla*.

5.5 Automatic Romanization

In the next step, the inference engine uses the fully vocalized canonical forms to generate tens

of millions of romanized variants using the rule base while applying constraint rules. Although all the generated variants are linguistically valid and conform to the rule base mappings as well as the constraint rules, at this stage they are only *potential variants* and there is no guarantee that a particular variants is used as a real name. The inference engine also generates romanized variants based on official romanization systems, such as ICS and ALA-LC.

Variant	Status	Frequency
'Abd al- Ḥaqq	ALA-LC	00001072
'Abdulhaq	attested	00047000
Abdul-Haq	attested	00042800
'Abd-al-Haqq	ICS	00018700
Abdul Haqq	attested	00005590
'Abdalhaqq	attested	00002080
Abdul-Hagg	validated	00001980
'Abdlhak	validated	00000689
'Abdul Hack	validated	00000193
Abed-Al Haq	validated	00000099
'Abdil-Haq	validated	00000089
Abdelhac	validated	00000074
'Abdulheq	validated	00000037
'Abd-Ul Haqq	validated	00000034
'Abdool Haq	validated	00000022
Ebdul Haq	validated	00000001
'Abad El-Hekk	unvalidated	00000000
'Abd-Al Heqq	unvalidated	00000000
'Abdl Hagg	unvalidated	00000000
'Abdoul-Hakk	unvalidated	00000000
'Ebdal Heqq	unvalidated	00000000

Table 6: Romanized variants of عَبْدُ آلْحَق

5.6 Validation

To eliminate false positives and ensure high data quality and integrity, several validation techniques are applied:

1. The most important of these is using web frequency acquisition tools to determine web occurrence statistics for each name variant. These techniques are constantly refined to achieve high performance and better precision and recall.
2. Variants of zero or low occurrence are eliminated from the main database (DAN) and placed in a database containing tens of millions of potential variants for future reference. This database is accessed by the generation module to ensure that the same false positives are not generated again.
3. The generated variants are matched against a large database of *attested variants*, which are variants that actually occur in sources such

as phone directories, books, encyclopedias, and the like.
4. Confidence level codes are assigned to the different levels of validation to distinguish between attested variants, official romanizations, high frequency variants, low frequency variants, and other categories.
5. Before integrating the data into DAN the data is subjected to several final sanitization procedures.

5.7 Arabic Variants

Another important component of the system is the **Database of Arab Names in Arabic** (DANA), a sister edition of DAN. This consists of sanitized and fully vocalized Arabic-script canonical forms of Arab names mapped to their variants, including attested variants, the unvocalized version, generated variants, and common spelling errors; e.g., عزيز ع بدا لٔ /ˁabd-ul-`aziiz/ mapped to ز عزي ال بد ع, which serves both for the generation of romanized variants as well as for Arabic name recognition.

Variant	Buckwalter	Canonical	Freq	Status
عبدالله	EbdAllh	عبد الله	59793720	variant
عبد الله	Ebd Allh	عبد الله	29959090	canonical
عبدلله	Ebdllh	عبد الله	00536060	error
عبداللة	EbdAllp	عبد الله	00000506	error
عبد الله	Ebd All~h	عبد الله	00000216	variant
عبد اللة	Ebd Allp	عبد الله	00000188	error
عبد اللاه	Ebd AllAh	عبد الله	00000129	error
عبد لله	Ebd llh	عبد الله	00000121	error
عبدالله	EbdAllAh	عبد الله	00000115	Error
عبدالله	EbdAll~h	عبد الله	00000091	variant
عبد اللا	Ebd AllA	عبد الله	00000021	error
عبداللا	EbdAllA	عبد الله	00000001	error

Table 7. Orthographic variants of عبد الله

6 Future Work

For romanized name candidates that are short or that have low frequency, using the web for validation does not guarantee 100% accuracy in identifying a string as an Arab name, unless the name in question is derived from an attested source; that is, a source known to contain Arab names. In the case of variants derived from rule-based generation, even if they are validated, there is some possibility that the string coinci-

dentally represents an ordinary word (in English or another language) or possibly a different name in Arabic that happens to be romanized identically.

For example, both أ سد and أ س عد are commonly romanized as *Assad*, so that it is not possible to determine which of these two the string *Assad* actually represents. A more extreme example is the name س ع د , which can be romanized as *Sad*, identical to the English word *sad*. This means that the high frequency obtained by web validation could be skewed, sometimes severely. This is an issue for which innovative solutions need to be found, but not so major as to detract form the overall value of the lexicon-driven approach.

At this stage of database development we are focused on Arab names only and have intentionally ignored non-Arab names. Various projects to automatically romanize non-Arab names and loanwords into the Roman script are reported in the literature (Stalls and Knight, 1998). Recently our institute has begun to build databases of non-Arab names similar in structure and scale to DAN and DANA, with the aim of providing a lexicon-based approach to the difficult task of converting foreign names back to the original script , such as converting ك ل ي ن تون to *Clinton* (this is often misleadingly called *back-transliteration*).

The most urgent task is to continually expand DAN and DANA to cover more variants. Even though the coverage is already comprehensive, it is technically challenging to collect or generate every existing variant. Detailed plans are now in place to validate up to seven million variants, and to further refine the validation techniques.

7   Conclusions

As we have seen, the ambiguity of the Arabic script and the multitude of romanization systems give rise to a vast number of romanized variants. To perform such procedures as named entity recognition and disambiguation, Arabic NLP applications require the use not only of statistical modeling techniques such as HMMs, but also of lexical databases. This paper describes comprehensive databases of Arab names and name variants in both Arabic and Roman script, as well as the techniques used to build and validate these databases. Combining a lexicon-driven approach with statistical methods is the key to achieving effective processing of Arabic names. Though some issues remain, such as skewed frequency

statistics due to romanized names accidentally spelled as ordinary words, the lexical-driven approach can significantly enhance statistically based methods.

In view of the many practical applications that require the ability to identify and process Arabic names, including automatic transcription, information retrieval, named entity recognition and security applications, there is a growing need for the continued development of large-scale Arabic lexical resources, especially of named entities.

References

Nasreen Abduljaleel and Leah S. Larkey. 2003. Statistical transliteration for English-Arabic Cross Language Information Retrieval. CIKM 2003: 139-146

Mansur Arbabi, Scott M. Fischthal, Vincent C. Cheng, and Elizabeth Bart. 1994. Algorithms for Arabic Name Transliteration. *IBM Journal of Research and Development.*

Kenneth R. Beesley. 2003. Xerox Arabic Morphological Analyzer Surface-Language (Unicode) Documentation. *Xerox Research Centre*. Europe.

Ali Farghaly. 2004. Computer Processing of Arabic Script-based Languages: Current State and Future Directions. COLING 2004

Jack Halpern. 2007. The Challenges and Pitfalls of Arabic Romanization and Arabization. *Proceedings of Computational Approaches to Arabic Script-Based Languages 2007*, Palo Alto, CA.

Jack Halpern. 2008. Exploiting Lexical Resources for Disambiguating CJK and Arabic Orthographic Variants. *Proceedings of LREC 2008*. Morocco.

Library and Archives Canada. 2006. *Inventory of Romanization Tools*

John Maloney and Michael Niv. 1998. A Fast, Accurate Arabic Name Recognizer Using High-Precision Morphological Analysis. *Proceedings of the Workshop on Computational Approaches to Semitic Languages..*

Martin Kay. 2004. Arabic Script-Based Languages Deserve to be Studied Linguistically. *Proceedings of COLING 2004,* Geneva.

Bonnie Glover Stalls and Kevin Knight. 1998. Translating Names and Technical Terms in Arabic Text. *Proceedings of the COLING / ACL Workshop on Computational Approaches to Semitic Languages..*