(This paper was presented at a conference in Tunis in 2002. Processing of Arabic Texts, hosted by the Faculty of Science and Arts at Manouba)

Title: What is a word?

Author: Andrew Freeman

Affiliation: Ph.D. candidate, Dept of Near Eastern Studies, University of Michigan, Ann Arbor, Michigan, USA

Contact Address: Andrew Freeman

3615 Burbank Dr.

Ann Arbor, MI 48105

USA

Email: andyf@umich.edu

Fax number: None

Phone Number: 1-734-222-8512

WHAT IS A WORD?

I. Introduction

The question of what constitutes a word is of interest to anyone wanting to create a lexicon. This question takes on added significance in computational linguistics when "real-world" considerations such as storage requirements, processing time and reusability of resources come into play.

This paper presents and explores a word-segmenting scheme, originally designed for Modern Standard Arabic that has been adapted for segmenting and then part-of-speech (POS) tagging a relatively small corpus of spoken Yemeni Arabic. The rule-based tagger, based on Brill's public domain rule-based POS tagger can be used with either variety of Arabic once it has been trained on both types of corpora.

In particular extending this word segmenter to other varieties of Arabic can be accomplished by adding to the list of closed-class items, such as the verbal aspect and modal particles, prepositions, pronouns, pronominal affixes and conjunctions. A potentially controversial feature of this word-segmenter is that it counts as separate lexical items the pronominal verbal subject, and object affixes, as well as the nominal possessive pronominal affixes, and the alif-lam of the definite article.

Segmenting the input word stream in this way greatly facilitates the creation of a lexicon used in tagging the text. It has the further advantage of helping the tagger train on the context. Tzoukerman, Radev and Gale present a similar result for a POS tagger for French. However, segmenting the text in this way is more than just a convenience for computerized language processing. There is theoretical support for viewing the verbal affixes as "embedded pronouns" from other linguistic traditions including Government and Binding (Fassi-Fehri, 1993; Ouhalla, 1999), traditional Arabic grammarians and modern pedagogy.

These are promising results, 1) from the standpoint of creating portable tools for Arabic language processing, 2) from a theoretical perspective of identifying the separable lexical units of any particular variety of Arabic, and 3) objectively measuring how close any particular variety of Arabic is to another.

II. The Segmenter

The segmenter was introduced at the Arabic workshop at Association of Computational Linguistics-2001 in Toulouse (Freeman, 2001). Of the 29 talks given at the above-named workshop at least six of them were devoted to morphological analysis of written Arabic. The original insight and motivation for the segmenter was that the following Arabic string "فسيكتبونها", can be decomposed into 6 contributing elements. These are: (ف, س, ي, كتب, ون, ها). Arguably, one can translate this into the following six English meanings: (and so, will, 3rd person, write, plural, her). Obviously, the two segments "3rd person" and "plural" have exactly the same meaning as the single English word "they."

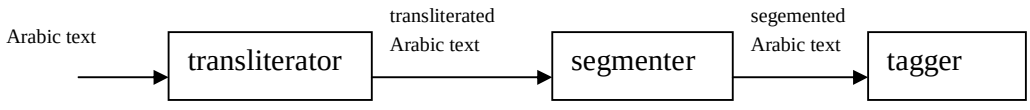
One can choose not to segment the Arabic text before doing part-of-speech tagging (POS) or any other kind of lexically based processing. But unless the system developer chooses to store every stem completely spelled for every possible combination of affixes, enclitics and one-letter prepositions and particles, it will be necessary to perform some kind of morphological analysis. Beesley (1996, 2001) has described a system built on Finite-State technology for parsing and analyzing Arabic morphology. This system is available online at http://www.xrce.xerox.com/research/mltt/arabic. It parses an input string trying to identify all possible analyses of every affix, part-of-speech, and root and pattern while simultaneously vocalizing each different analysis. The transliteration scheme is listed in Appendix A. With the following sentence "لم أبال بنداءات السائق", the system produces six analyses for the first string "لم", more than thirty analyses for the string "أبال" and because of orthographic irregularities, it cannot analyze the string "بنداءات" at all.

None of the above is in any way intended to discount the value of this morphological analysis tool. Indeed this researcher is extremely grateful that Xerox has generously made this tool available to computational linguists working with Modern Standard Arabic. It bears mentioning that Xerox's morphological analysis tool is analyzing each word in isolation, and that almost all successful linguistic *annotation* tools use some sort of n-gram statistical model to disambiguate the grammatical categories of any particular word.

The point is that the original motivation for building the segmenter described here was to the need to create a list of taggable items with their associated tags (tagger lexicon) to give it to an implementation of Brill's transformation-based error-driven machine learning trainable tagger. The claim and hope is that Brill's tagger will learn the parts-of-speech increasingly more correctly as the size of the annotated corpus grows. The first few iterations need to be tagged mostly by hand, but as the tagger is trained on the ever-growing annotated corpus it learns how to correctly tag all word combinations found in the training corpus. In other words, rather than hand crafting a large number of disambiguation rules, the software learns the disambiguation rules as the annotated corpus grows. It is hoped that eventually this annotated corpus will serve as input to a shallow-parser, in order to start building a parse-tree corpus.

It also bears mentioning that Brill's tagger will only take latin-based character sets, so the first stage of the process is to transliterate the text using Xerox-Buckwalters' transliteration scheme which allows for a one-to-one correspondence between the Arabic character set and the latin character set. A schema for the entire system is shown in figure one.

Figure 1: Data flow of entire system



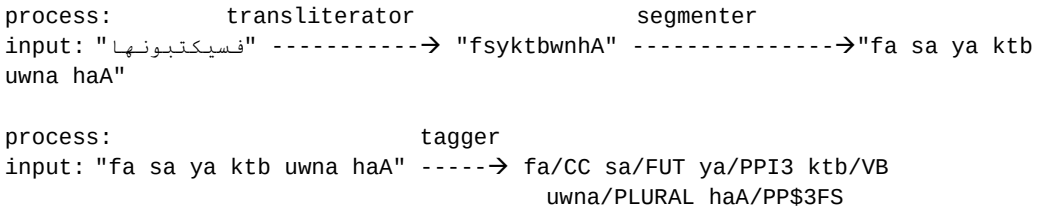
The transliterator is trivial and amounts to using the input as an index to an array cell containing the numeric value corresponding to the correct transliteration character. The tagger is described in Brill (1994, 1995). The segmenter is implemented as a finite-state transducer and can therefore be described using regular expressions. The segmenter was written in c++ and implements the six regular expressions in table 1.

Table 1: Six lexeme recognizing regular expressions in written Arabic

```
Legend:
   Items in parentheses are optional.  The "?" means zero or one
         occurrences.
   conj is any item from the set {f, w, >}
   det == Al
   part_1char_noun is any item from the set {b, k, l}
   part_1char_verb is any item from the set {l, s}
   noun_stem == any item in the segmenter stem dictionary
   verb_stem == any item in the segmenter stem dictionary
   prep_stem == any item in the segmenter preposition dictionary
   inn_wuxt is any item from the set {>n, <n, k>n, l>n, lEl, lkn,
         wlkn}
   imp_verb_pfx is any item from the set {>a, ta, ya, na}
   imp_verb_sfx is any item from the set {wA, wn, yn, y, An, A, n}
   perf_verb_sfx is any item from the set {nA, t, wA, tmA, tmw, tA,
         A, w}
   noun_affix is any item from the set {w, y, wn, yn, At, A, An}
   poss_pron is any item from the set {y, k, h, hA, kmA, hmA, nA, km,
         kn, hm, hn}
   obj_pron is any item from the set {ny, k, h, hA, kmA, hmA, nA, km,
         kn, hm, hn}
The regular expressions are:
   regular expression for indefinite noun
         (conj)? (part_1char_noun)? noun_stem (noun_affix)?
         (poss_pron)?
   regular expression for definite noun
         (conj)? (part_1char_noun)? det noun_stem (noun_affix)?
   regular expression for preposition
         (conj)? prep_stem (poss_pron)?
   regular expression for inna and her sisters
         (conj)? inn_wuxt (obj_pron)?
   regular expression for imperfect verb
         (conj)? (part_1char_verb)? imp_verb_pfx verb_stem
         (imp_verb_sfx)? (obj_pron)?
   regular expression for perfect verb
         (conj)? (l)? verb_stem (perf_verb_sfx)? (obj_pron)?
```

The transliterator will transform its input into a string of transliterated characters. The output from the transliterator is passed to the segmenter, which hands its output to the tagger.

Figure 2: Example showing transforms on sample input



In the example in figure 2, the transliterator will transform the string of Arabic chars "فسيكتبونها" into the transliterated string "fsyktbwnhA". The segmenter upon receiving the input string "fsyktbwnhA" from the transliterator will output the six segments (fa sa ya ktb uwna haA). These segments are what the tagger receives as it input. The tagger will treat each one of these strings as a separate lexical item and produce the following tagged output "fa/CC sa/FUT ya/PPI3 ktb/VB uwna/PLURAL haA/PP$3FS". The tagset is listed in Appendix B.

There is a fair amount of non-determinacy in these regular expressions owing to some overlap of the morphology, such as some of the imperfect plural markers with the sound plurals of nouns. This leads to a fair amount of backtracking. Also there are some segments that cannot be determined without considering contexts larger than individual character strings. For instance the string "smEthA" (سمعتها) can be segmented as either of smEp haA (سمعة ها), "her reputation" or smE t haA (سمع ت ها), I/you/she heard her/it. There is no principled way of determining the correct output for this string without looking at the strings in the surrounding context. There is now a sufficiently large segmented and tagged corpus (30,000 segments), to consider improving the performance of the segmenter by creating and then using a tri-gram statistical language model. This will add some statistically guided decision capabilities when there is more than one possible way to segment the input.

III. Verb affixes as separate lexemes: Computational motivations

Counting the verbal affixes as separate lexemes was originally motivated by a desire to get a system "up and running" as quickly as possible.

Brill's tagger has no resources for lemmatizing the input. Any lumping together of disparate strings into a single category needs to be accomplished prior to giving the input to the tagger.

In this case, lemmatizing would require building an elaborate database with roots marked for all of the derived forms and rules for how to generate every possible verb-form. This amounts to storing all of the verb forms in shorthand form. To do this for every verb form would require an incredible amount of storage and/or processing. The imperfect verbs are 12* 12 forms just for the subject markers and every combination of object pronoun. If we consider the conjunctions and the particles, we would need to store or generate at least (2*2*12*12 = 576) forms for every imperfect verb form, without even taking into account the subjunctive, jussive or perfective forms.

Discounting storage issues for the time being, there is theoretical support in information theory for explaining why separating the person agreement markers from the verb forms would help a machine-learning algorithm learn a task with less input. The string "yaktub" has embedded in it information for three sets of features. These are: 3^{rd} person for the person feature, imperfect for the tense feature, and "write" for the semantic/action feature. In contrast, the string "ya" only carries information for two features 3^{rd} person and imperfect, while "ktub" has only one piece of information "write". In general when dealing with input that has a random distribution the number of bits required to store that information $\log_{\text{base two}}$ of the number of features being differentiated. To differentiate between all of the valid verb stems is $\log_{\text{base two}}$ (count of valid verb stems, perhaps 10000). Obviously the number of bits needed to differentiate between all imperfect verb forms is $\log_{\text{base two}}$((count of valid verb stems) * 576), which is a much larger number. So, separating off the subject affixes from the imperfect verb form divides the string recognition problem into two much smaller problems: the four subject prefixes (>a, ta, ya, na), which needs two bits and the count of valid verb

stems, which has to be less than 10,000, which then needs less than 15 bits. The original problem was forcing us to differentiate between perhaps 5,000,000 verb forms, which needs more than 32 bits or branches in our decision tree.

Furthermore, language is not completely random. Once we have correctly identified a valid imperfect verb person suffix, the only thing that can follow that person agreement marker is, in fact, a verb stem. One of the rules learned by Brill's tagger when training on a correctly tagged corpus was the following rule: "NP VB PREVTAG PPI1S". Translated into plain English this rule says: "change a proper noun into a verb stem if the previous tag is a first person singular marker." The tagger will tag any word not in its lexicon of taggable items as a proper noun if it begins with a capital letter, then this rule will come along during a later pass an correct the tag because the preceding tag was the first person subject imperfect verb marker.

In closing this section, there are two major points. The first is that calling all of the affixes "separate lexemes" or words makes the size of the computerized word list with which we are working much more manageable. In the second place, separating off all of the affixes (definite article, object pronouns, subject pronouns, verbal subject markers) means that our machine learning algorithm only has to train on a sufficiently large number of instances of the stem in question, and then a sufficiently large enough instances of the separated affixes, treated as indepndent events. Not separating off these items necessitates training on the same number of instances of the stem as before, times the number of all possible combination of affixes. Finally, we can use some of the affixes to correctly identify the part-of-speech deterministically which is the current foreground task. For instance, in Arabic, a word preceded by the definite article can only be a noun or an adjective.

IV. Verb affixes as separate lexemes: Support from other paradigms

IV.1 Pedogogy

All of *Yemeni Arabic -1-, Elementary Modern Standard Arabic, and Al-Kitaab fii Tacallum al-cArabiyya* are textbooks are used in the United States to teach Arabic to college students. All of these textbooks give something very similar to Table 2 to teach Arabic's imperfect verb paradigm.

Table 2. Imperfect verb paradigm

singular pronoun	prefix	verb stem goes here	suffix (if any)	plural pronoun	prefix	verb stem goes here	Suffix (if any)
'anaa	'a	_____		naHnu	na	_____	
'anta 'anti	ta ta	_____ _____	iina	antum antunna	ta ta	_____ _____	Uuna Na
huwa hiya	ya ta	_____ _____		hum hunna	ya ya	_____ _____	uuna na

The stem goes into the blank between the suffix and the prefix and the affixes remain the same regardless of any of the stem's features. Also, the independent pronoun is completely redundant. In fact, all three of these textbooks stress that the preferred style is to not use the independent pronoun. The point here is that the affixes can take the place of the independent pronoun. It is true that they do not have an independent existence, since they cannot appear without the verb and when an overt noun phrase appears as the subject they are still required to appear on the verb. But they do take the place of the pronoun in the presence of the imperfect verb. Another thing worth noticing is that the plural agreement is the same regardless of what the number agreement is. Given that all three of the above-named textbooks have at least one author who is a native speaker of Arabic, it seems safe to say that many educated Arabs have in their minds a structure very similar to table 2 that helps them generate and decode Arabic's imperfect verbs.

The distribution of these verbal subject "pronouns" is that they can only appear in these very specified slots before and after the verb stem. However, their structure and meaning is completely independent of the verb stem in every respect.

IV.2 Traditional Arabic grammar

Let us examine the analysis of Arabic grammar as performed according to the traditional system. The traditional grammatical analysis for sentence 1 is taken from *A Dictionary of Grammatical Analysis of the Holy Qur-an* (Librairie du Liban, 1995). The following text is verse 3 of sura 6, Al-An^{c}am. I beg the readers forgiveness for using Quranic texts here, but the analysis of the i^{c}raab is obviously impeccable. أعوذ بالله من الشيطان الرجيم

(1)
fهُوَ اللهُ فِي السَّمَوَاتِ وَفِي الأرْضِ يَعْلَمُ سِرَّكُمْ وَجَهْرَكُمْ وَيَعْلَمُ مَا تَكْسِبُونَe

A. Yusuf Ali's translation = e He is God in the heavens and on earth. He knoweth what ye hide, and what ye reveal, and He knoweth the (recompense) which ye earn (by your deeds).f

The traditional analysis of the "i^{c}raab" for the verb "يَعْلَمُ" (he knows) according to *A Dictionary of Grammatical Analysis of the Holy Qur-an,* is "فعل مضارع مرفوع بالضمة الظاهرة", which I will translate to mean "a present tense verb inflected for the active voice with a visible dhamma." The i^{c}raab for the subject is "فاعله ضمير مستتر تقديره هو" which I translate as "a hidden (or implied) pronoun considered to be 'he.'" Similarly, the analysis for the verb "تَكْسِبُونَ" is "فعل مضارع مرفوع بثبوت النون" or "a present tense verb inflected for the active voice with the 'n' remaining in place." The subject is analyzed as "الواو ضمير متصل في محل رفع فعل" or "the 'w' is an attached pronoun in place of putting the verb in the active voice."

It is clear that both of these analyses go beyond mere person and number agreement, and strongly suggest that this morphology is equivalent to the presence of a pronoun.

IV.3 Perspectives from Modern Linguistics

Fehri (1993) makes an argument for the strong claim that the subject morphology on the verb is an "embedded pronoun." His argument is in the now mostly highly revamped Government and Binding framework. Without going into great detail, some of the data on the distribution of verb agreement morphology that Fehri uses are:

1) when the verb comes first there is no number agreement between the verb and the post-posed subject.
2) even though the preferred position for the subject noun is after the verb, when the independent pronoun follows the verb the independent pronoun does not block the agreement.

(2) (* indicates ungrammaticality, question mark means questionable)

a. ji?-na
came-3.pl.f.
They came.

b. jaa?-uu
came-3.pl.m.
They came.

c. jaa?-at al-banaat
came-3.s.f. the-girls
The girls came.

d. *al-banaat jaa?-at
the-girls came-3.s.f.
The girls came.

e. al-banaat ji?-na
the-girls came-3.p.f.
They came.

f. hunna ji?-na
they-f. came-3.p.f.
They came. (This is a topic fronting sentence)

g. ? ji?-na hunna
came-3.p.f. they-f.
They came.

h. *jaa?-at hunna
came-3.s.f. they-f.
They came.

The shortest hand wave I can do for the argument in Fehri (1993) is that it hinges on the fact that when the noun comes after the verb, the noun can

carry the number agreement features which are in essence governed by the licensing verb. However, when the noun does not follow verb, the verb needs to have overt morphology that it can govern in order to carry the number feature. Owing to c-command constraints the verb cannot govern any item that precedes it. The number morphology on the verb in the case when the noun comes first counts as an embedded pronoun that refers back to the first instance of the referring noun phrase (NP).

In any case, there is some interaction between the verb morphology and the subject NP. In Modern Standard Arabic the NP can block the number feature on the verb when it follows the verb. This number feature works almost like a pronoun in that it apparently replaces a referential noun phrase. A replacement pronoun needs to follow the NP that it refers back to, is the very non-technical insight. A "regular" pronoun does not interact with this number feature on the verb in quite the same way. Apparently, the person and number features on the verb carry some of the functional load of a pronoun.

Niemi, Laine and Tuominen (1994) studied the morphological processing of nouns in Finnish by human subjects. They came to the conclusion that the nominative singular form of the noun is the psychologically real base form or stem of the Finnish noun. Inflected but not derived Finnish nouns are parsed into stems and affixes in word recognition. Niemi, Laine, Tuominen found that inflection and affixation of function particles was a different process from word derivation in Finnish. Admittedly, Arabic's affixation of conjunctions, determiners and possessive pronouns to the noun in the writing system is nowhere near as complex as Finnish's nominal inflectional system. However, it is hard not to imagine that the mental processing of a speaker of Arabic does in fact parse the noun string "bsayyaaratihi" into the individual morphological units "bi", "sayyaara" and "hi."

I will point out that the segmenter system outlined here is only interested in separating open-class noun, adjective, multi-character prepositions and verb stems

from closed-class functional affixes, such as prepositions, object, subject and possessive pronouns and conjunctions.

No effort has been expended in trying to get the segmenter to decompose the stems into a root and pattern. This root and pattern processing is not necessary for the functioning of Brill's tagger. I will note here that Beesley (1996, 2001) and Kiraz (1998, 2000) both of whom are working with Arabic and Semitic languages have adopted many of the finite-state tools for modeling two-level morphology that Koskenniemi originally developed for Finnish's complex morphology.

V. Adapting the segmenter to a spoken corpus

In the course of doing the research for my Ph.D. dissertation in sociolinguistics I recorded and transcribed a fair amount of spoken Arabic in the various Yemeni vernaculars. Roughly a third of this data is currently in machine-readable form and amounts to a little bit more than 50,000 words. It turns out that adapting the segmenter and consequently the tagger to handle Yemeni Arabic is relatively easy. Since my tagged corpus of Standard Arabic is still only 30,000 segmented tokens corresponding to roughly 8,000 strings of unsegmented Arabic, I need to add to the segmenter's and the tagger's stem lists every time I increase the corpus. So adding open class items to the word-lists is not an extra chore. To deal with segmenting the Yemeni Arabic I need to add to some of the closed class items, such as the pre-verbal future markers (ᶜad, sha, ᶜa, ba), the present continuous markers (bayn, bi) and some of the prepositions, conjunctions and question words (laysh, lilmah, 'aysh). The point is that adding these items to the segmenter, can be accomplished without changing any of the procedural code for the segmenter, simply by editing the lists for those classes of items. I will claim that this means that the segmenter has captured an important generalization about the structure of Arabic and how to process the morphology of its affix system.

I also claim that this gives us a metric for measuring how close any of the dialects are to Modern Standard Arabic and to each other by measuring how much work needs to be done when trying to tag texts from a dialect that the tagger has not trained on yet.

VI. Prospects for the Future

Currently, there are segments that cannot be disambiguated by examining a single string of text in isolation. The corpus has finally grown to the point where it makes sense to build a statistical language model in order to use the context of the preceding two segments to choice the most likely segmentation for segments that have more than one allowable segmentation.

I also believe that I am ready to start editing Brill's tagger code to take advantage of the information uncovered by the segmenter. For instance any string with verb agreement/pronoun morphology affixed to it can be called a verb regardless of whether or not the stem exists in the tagger's word list.

I also want to predict that segmenting the text in this way will help with parallel corpus-based statistical translation of Arabic. This is not idle speculation. The Center for Language and Speech Processing at John Hopkins University developed a statistical machine translation toolkit called "Egypt" during a 1999 summer workshop on statistical machine translation. They have very generously made this toolkit available on the web at the URL http://www.clsp.jhu.edu/ws99/projects/mt/ . This toolkit compiles and runs under unix and linux. It comes complete with a parallel corpus of the qur'aan, in English and Arabic. There are some minor problems with the code and the parallel corpus. For instance, some of the verses in the Arabic text are longer than the maximum allowed sentence length of 41 words. Furthermore, the training stage of building the statistical model requires that the parallel corpora contain the same number of sentences, and typically English translations of the qur'aan break each verse into several sentences. So there is a minor amount of

mindless work needed for aligning the corpora, before any tests can be run. However, based on remarks by Kenji Yamada from the Information Sciences Institute when he presented his paper "A Syntax-based Statistical Translation Model" at ACL-2001 in Toulouse I understand that they have not had good luck building a statistical translation model for translating between English and Arabic. I am convinced that segmenting the text into stems and affixes will noticeably improve the performance of the statistical translation model for Arabic. Basically because the entropy will be less, i.e. the second segment of the segmented strings "ya ktub" and "ta ktub" will both produce the same tokens for the English "wrote", whereas "yaktub" and "taktub" will need a corpus larger by some function of the relative distributions of "yaktub" to "taktub" in order to provide the same statistical coverage.

VII. Conclusion

I have documented a scheme for segmenting Arabic text into stems and affixes using the well understood and efficient finite-state/regular expression formalism. For certain kinds of lexeme lookup and corpus-based machine learning tasks it seems like the shortest path to bootstrapping one's way into a working system in order to acquire a part-of-speech annotated corpus. Arguably, acquiring a part-of-speech annotated corpus is the first step in building more interesting language resources such as syntax tree banks and statistical machine translation models.

Additionally, separating the tokens into stems and affixes is more than just an expedient way to avoid spelling out all possible combinations of stem and affix to save on computer resources. It has theoretic support from pedagogy, traditional Arabic grammar and modern linguistic theory and research. It would seem that the definite article, the verb affixes, the possessive and object pronouns, the verbal and the prepositional particles all have a separate semantic and syntactic existence independent of the semantics

of the word to which they attach. There is evidence that this is the way that our minds process these items. I will argue that if we can reduce the number of permutations for any item, it will help the task of building a compact representation. This leads right into the argument that for lexical lookup in general we want to only store the stem in our lexicon.

In any event, the segmentation scheme offered here is not meant to be definitive. It does seem like this is an area that could lend itself to being the subject of an agreed upon public standard. What is an appropriate level of morphology parsing for Arabic on which most lexicographers can agree? What constitutes a stem and a separable affix? In answer to the question in the title of this paper, there is no clear-cut linguistic definition the word "word". However, the verb affixes that my segmenter separates from the verb stem probably do not qualify as words, if the meaning of "word" is restricted to those items that have a semi-independent existence. The verb affixes and object pronouns have a very limited set of environments in which they can occur before and after the verb or noun stem. On the other hand, it is abundantly clear that separating them from the stem makes lexicon building, lexeme lookup and matching strings with linguistic part-of-speech tags many times easier.

Appendix A Transliteration scheme

Figure 1. Transliteration Scheme

A = ا
b = ب
t = ت
v = ث
j = ج
H = ح
x= خ
d = د
* = ذ
r = ر
z= ز
s = س
$ = ش
S = ص
D = ض
T = ط
Z = ظ
E = ع
g = غ
f = ف
q = ق
k = ك
l = ل
m = م
n = ن
h = ه
w = و
y = ي
Y = ى
p = ة
a = ـَ , i = ـِ , u = ـُ
N = ـً ,K = ـٍ ,F = ـٌ
o = ـْ , ~ = ـّ
aA = ـَا, iy = ـِي
uw = ـُو ,aw = ـَوْ, ay = ـَيْ,
‘= ء ,>= أ , < = إ , } = ئ
& = ؤ, | = آ

Appendix B Tagset

. sentence closer . ; ? !
(left paren
) right paren
-- dash , comma
: colon
ABL pre-qualifier quite, rather
ABN pre-quantifier half, all
ABX pre-quantifier both
ABBR abbreviation
AP post-determiner many, several, next
AT article a, the, no
CC coordinating conjunction and, or
CD cardinal numeral one, two, 2, etc.
CS subordinating conjunction if, although
DT singular determiner this, that
DTI singular or plural determiner/quantifier
DTD singular or plural determiner/quantifier
DTS plural determiner
DTX determiner/double conjunction either
EX existential there
FUT future marker; imperfective conjugation
FW foreign word
HL headline (hyphenated after regular tag)
IN preposition
JJFS adjective, fem singular
JJFP adjective, fem plural
JJMS adjective, masc singular
JJMP adjective, masc plural
JJR comparative adjective
JJS semantically superlative adjective
JJT morphologically superlative adjective
MD modal auxiliary can, should, will or other
NC cited word (hyphenated after regular tag)
NNF singular or mass noun, fem
NNM singular or mass noun, masc
NNFA singular noun, fem, acc. case
NNMA singular noun, masc, acc. case
NNSF plural noun, fem
NNSM plural noun, masc
NNMS verbal noun, gerund
NNMSA verbal noun, gerund, acc. case
NN$ possessive singular noun
NNS$ possessive plural noun
NP proper noun or part of name phrase
NP$ possessive proper noun
NPS plural proper noun

NPS$ possessive plural proper noun
NR adverbial noun home, today, west
NRS plural adverbial noun
OD ordinal numeral first, 2nd
PIND indefinite pronoun
PN nominal pro everybody, nothing
PN$ possessive nominal pro
PP$1S 1st person singular poss. pro suffix
PP$2S 2nd person singular poss. pro suffix
PP$3MS 3rd person masc singular poss. pro suffix
PP$3FS 3rd person fem singular poss. pro suffix
PP$2D 2nd person dual poss. pro suffix
PP$3D 1st person poss. pro suffix
PP$1P 1st person plural poss. pronoun suffix
PP$2PM 2nd person masc plural poss. pronoun suffix
PP$2PF 2nd person fem plural poss. Pro suffix
PP$3PM 3rd person masc plural poss. pro suffix
PP$3PF 3rd person fem plural poss. pro suffix
PPI1S 1st person singular imp. subj pro prefix
PPI2FSFX 2nd person fem. singular imp. suffix
PPI2S3F 2nd & 3rd fem. imp. subj pro prefix
PPI3 3rd person imp. subj pro prefix
PPI1P 1st person plural imp. subj pro prefix
PPPR123FS 1st, 2nd and 3rd fem perf subj pro suffix
PPPR2S3F 2nd person & 3rd fem imp. subj pro prefix
PPPR3 3rd person imp. subj pro prefix
PPPR1P 1st person singular imp. subj pro prefix
PPPR2PM 2nd plural perf masc subj pro suffix
PPPR2PF 2nd plural perf fem subj pro suffix
PPPR2D 2nd dual perf subj pro suffix
PLRFIP feminine plural marker for perfect and imperfect conjugations
DUAL ending for dual nouns in construct, imperfect verbs in subjunctive or perfect verbs
PLURAL_VB plural suffix
PLURAL_OBL plural suffix in construct
PLNMF plural marker for fem nouns
PPS1 1st singular nominative personal pro
PPP1 1st plural nominative personal pro
PPS2 2nd singular nominative personal pro
PPS2D 2nd dual nominative personal pron
PPPM2 2nd plural nominative personal pro
PPPF2 2nd plural feminive personal pro
PPPM3 3rd. plural masc. nominative pro
PPPF3 3rd. plural feminine nominative pro
PPSF3 3rd. Singular feminine nominative pro
PPSM3 3rd. singular masculine nominative pro
PPP3D 3rd. dual nominative pro
PP$ possessive personal pro
PP$$ second (nominal) possessive pro
PPL singular reflexive/intensive personal pro

PPLS plural reflexive/intensive personal pro
PPO objective personal pro
QL qualifier very, fairly
QLP post-qualifier enough, indeed
QM question marker
RB adverb
RBNEG negating adverb
RBR comparative adverb
RBT superlative adverb
RN nominal adverb here then, indoors
RP adverb/particle about, off, up
TL title (hyphenated after regular tag)
UH interjection, exclamation
VB verb, base form
WDT wh- determiner what, which
WPIND indefinite relative pronoun
WPMS relative pronoun, masc singular
WPFS relative pronoun, fem singular
WPMP relative pronoun, masc plural
WPFP relative pronoun, fem plural
WPMD relative pronoun, masc dual
WPFD relative pronoun, fem dual
WQL wh- qualifier how
WRB wh- adverb how, where, when

Bibliography

Beesley, Kenneth (1996). "Arabic finite-state morphological analysis and generation." in *Coling 96*, volume 1, pages 89-94, Copenhagen, August 5-9. 16th International Conference on Computational Linguistics.

Beesley, Kenneth (2001). "Finite-state Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plans in 2001." in *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Workshop proceedings, Arabic Language Processing: Status and Prospects*. Pages 1-8.

Brill, Eric (1994). "Some advances in rule-based part of speech tagging." In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, Wa.

Brill, Eric (1995). "Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging." *Computation Linguistics*, 21(4):543-565.

Fassi Fehri, Abdelkader (1993). *Issues in the Structure of Arabic Clauses and Words*. Dordrecht, The Netherlands. Kluwer Academic Publishers.

Kiraz, G and E. Grimley-Evans. (1998). "Multi-tape automata for speech and language systems: A Prolog implementation." In Derek Wood and Sheng Yu, editors, *Automata Implementation.* Lecture Notes in Computer Science, Number 1436. Springer Verlag, pages 87-103.

Kiraz, George. (2000). "Multitiered Nonlinear Morphology Using Multitape Finite Automata: A Case Study on Syriac and Arabic." *Computation Linguistics*, 26(1):77-105.

McCarthy, J. 1979. *Formal Problems in Semitic Phonology and Morphology*. Ph.D. thesis, MIT, Cambridge, MA.

Niemi, J., Laine, M., Tuominen, J. (1994). "Cognitive Morphology in Finnish: Foundations of a New Model." In Sandra, D., and Taft, M. editors *Morphological Structure, Lexical Representaion and Lexical Access*, Hilsdale, USA. Lawrence Erlbaum Associates, Publishers, pages 423-446.

Ouhalla, Jamal (1999). *Introducing Transformational Grammar, from principles and parameters to minimalism, second edition.* London. Arnold

Tzoukerman, E., Radev, D., Gale, W. (1999). "Tagging French without lexical probabilities – combining linguistic knowledge and statistical learning." In Susan Armstrong, Kenneth Ward Church, Pierre Isabelle, Sandra Manzi, Evelyne Tzoukermann, and David Yarowsky, editors, *Natural Language Processing Using Very Large Corpora*. Kluwer Academic Publishers.

Watson, Janet. *A Syntax of San^{c}ani Arabic* , Weissbaden,. Harrassowitz. 1993

Watson, Janet. *Wasf Ṣanᶜāni, Texts in Ṣanᶜāni Arabic*, Weissbaden,. Harrassowitz. 1996

Wehr, Hans. Ed. By J Milton Cowan *A Dictionary of Modern Written Arabic.* Ithaca, NY. Spoken Language Services, Inc. 1994

Wright, W. *A Grammar of the Arabic Language*. Cambridge. Cambridge University Press., 1967.

Yamada, Kenji and Knight, Kevin (2001). "A Syntax-based Statistical Translation Model." in *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference* 523-530.