Nordic Journal of African Studies 1(1): 87–119 (1992)

A Two-Level Computer Formalism for the Analysis of Bantu Morphology

An Application to Swahili

ARVI HURSKAINEN
University of Helsinki, Finland

ABSTRACT

SWATWOL is a computer program which has been designed to analyze morphologically Standard Swahili texts. It is based on Koskenniemi's (1983) already well-known two-level model. A number of applications of this model on various languages exist already. Some of those are very ambitious and almost complete (Koskenniemi 1983; Karlsson 1992), others being still in various stages of development. The implementations have concerned so far such languages, where inflection and derivation is taken care of primarily by means of suffixation. This is the first effort to apply the two-level model to a primarily prefixing language. SWATWOL includes a full description of Swahili inflectional morphology and morphophonology and contains a lexicon system with more than 25 000 lexical entries. The program identifies and analyzes all correct word-forms as far as the lexical items are listed in the lexicon. The extensive tests on various types of Swahili texts described below indicate that the coverage and precision of the program are close to perfection.

INTRODUCTION

In recent years there has been an increasing interest in the use of computers in the study and teaching of African languages. Many African languages are well suited to computational processing. In computational linguistics, the emphasis has been on ambitious automatic translation projects, on devising syntactic parsers (Sato 1988; Gazdar et al. 1988; Tomita 1987; Milne 1986), on computational lexicology (Ritchie 1987; Zernik and Dyer 1987), and on preparing ready-made packages (Butler 1985). Many African languages offer also other kinds of challenges, particularly those connected with the automatic analysis of word-forms. Bantu languages, for example, offer good testing grounds for programs designed for morphological analysis, because they use extensively both inflection (through prefixation) and derivation (through suffixation).

Much of the discussion on the feasibility of a morphological parser concerns memory and run-time requirements, as well as the possibility of using open lexicons. When related to the types of languages found in Africa and the problems posed by them, some of this discussion seems irrelevant. Jäppinen and Ylilammi (1986), for example, take it as an advantage that the parser developed by them is able to parse forms of lexemes that are not in the lexicon. This is possible, they

claim, because their model processes from right to left. This is true insofar as the language inflects through suffixes, as is the case with most Indo-European languages. In analyzing from right to left, the lemmas are normally the last morphemes in the form, and the model may be made to recognize the correctness of a form even though its lemma is not known in advance. The case is different in many African languages, e.g. all Bantu languages and a number of Nilo-Saharan languages, which inflect through both suffixes and prefixes. Because the large lexicon of lemmas is located in the middle, and not at the beginning or end, of the sequence of morphemes, the direction of processing does not solve the problem.

My purpose here is to describe the implementation of one morphological parser, SWATWOL, on Swahili, which is a Bantu language. This parser processes from left to right.

1.0 Description of the Two-Level Formalism

The general formalism for automatic analysis and production of word-forms and the computer program for implementing it was first designed by Kimmo Koskenniemi (1983). It is a language-independent program, which can be applied to the analysis of any language. Its usefulness in the analysis of morphologically complex languages is apparent.

Theoretically the formalism has some similarities with generative phonology, which has been successfully used for the description and analysis of inflectional morphology and morphophonology. One of the weak points of generative phonology is, however, that its capabilities of describing the dynamic processes of word analysis and generation are limited. This weakness is due to the functioning of the model, where a predetermined order of rewriting rules is essential (Lass 1984, 214-33). The two-level model overcomes these difficulties, because it uses parallel rules in describing morphophonological variation.

The two-level formalism treats strings of characters as sets of correspondences, where each lexical character has a surface representation. This property makes it possible to study morphological phenomena on both levels. The formalism allows both the study of how a lexical form is realized in a surface representation, and also what kind of lexical form lies underneath a surface form. The requirement of exact correspondence between the lexical and surface levels sometimes causes unexpected problems, for example in cases where a lexical phoneme has allophones of a different length in the surface representation (e.g. /k/ > /ch/; /sh/ > /z/). These can be handled, however, as we shall see later.

The system has an invariant central module, termed here TWOL[1], which takes as input a set of transducers (a kind of finite state automata), and a set of lexical

[1] TWOL is a shorthand of 'two-level', and it is used here as the name of the central module, although some other names for it have been used. Karttunen (1983) uses the name KIMMO;

entries. Each transducer is in fact a morphological or phonological rule, which defines the context where the specific (deviant) surface representation takes place. A specific compiler has been developed which automatically transforms the rules into transducers (Karttunen et. al. 1987). The user of TWOL needs to formulate the morphological and/or phonological rules of the deviant surface representations in the language, and also the lexicon system with a set of sub-lexicons.

1.1 THE TWO-LEVEL RULES

Although the purpose of this paper is not to serve as a user guide to the system[2] , some basic description of its structure is necessary for understanding its working.

A rule, as understood in this system, has typically the following format:

(1) CP "op" LC __ RC

where: CP signifies the **correspondence part**, i.e. the lexical character and its surface representation, whose occurrence is restricted by the rule. It may be a concrete or an abstract character pair, whereby in the latter case an abstract character has to be defined elsewhere.

The **operator** ("op") defines the type of restriction the rule contains. There are three types of operators:

(2) => signifies the **context restriction rule**, and it defines the context where such a correspondence between the lexical and surface character is permitted. It does not, however, make the realization necessary in this context.

(3) <= The **surface coercion rule** makes the particular realization of CP necessary in the given context, but it does not prevent such realizations also in other contexts.

(4) <=> This operator is a combination of the above types and it is the most common one in rules. It states that a CP is permitted if and only if it occurs in a given context.

The context where the rule applies may be defined by giving the environment on both sides of the CP. This is not, however, obligatory; either side may also be left undefined. LC means the left context, the underscore the location of the CP, and RC means the right context. In the definition of the context both the lexical and surface representations may be used. This is a very useful facility, which enables one to

---

Dalrymple, Karttunen and Shaio (1987) call it DKIMMO, while TWOL in their use means the compiler which compiles rules into finite-state automata.

[2] Practical information about its use is provided e.g. by Dalrymple, Kaplan et al. (1987).

define the context exactly, so that the rule is neither too restrictive nor too permissive.

Within the present formalism, a colon (:) separates the lexical and surface level. A symbol written to the left side of the colon means the lexical representation, and a symbol on its right side is a surface realization.

The equivalences between the lexical and surface level may be illustrated in this way:

(5) lexical level: #nInasomIshA# #nI/asomIshA#
surface level: 0ninasomesha0 0n00asomesha0

Note that a lexical character may be realized as itself, as another character, or as zero. Any other realization than the default one is accounted for by rules.

An example of a phonological rule written in this formalism:

(6) "I > 0 in front of the present tense marker **-a-**"
I:0 <=> #: [n: | l: | z: | k:] _ /: :a ;

The text between the quote marks is the name of the rule. It is supposed to be a short verbalization of what the rule represents. The above rule can be verbalized more specifically as follows:

The lexical {I} has a zero realization if and only if it occurs in an environment where its left context is a lexical word boundary {#} followed by any of the lexical consonants {n, l, z, or k}, and its right context is the lexical diacritic {/} followed by a surface /a/.

This rule defines explicitly the context where the zero realization of the lexical I is permitted and compulsory.

The same lexical character may have another kind of realization in another environment. The following rule illustrates this:

(7) "I:y in front of the present tense marker -a-"
I:y <=> #: (:V) _ /: :a ;

The rule causes lexical {I} to be realized as surface /y/ in an environment where its left context is a lexical word boundary {#} followed optionally by a surface vowel, and its right context is the lexical diacritic {/} followed by a surface /a/.

Note that by placing a context delimiter within the curved brackets one can expand the range of the cases where the rule applies. The left context of this rule is satisfied if on the left side of the character pair to be defined (I:y) there is nothing but a word boundary. It is also satisfied when the word initial is a surface vowel, but not if it is a consonant, for example.

By means of such phonological rules it is possible to account for phonological irregularities as well as regular permutations of the lexical forms. There are,

however, morphotactic phenomena in Bantu languages, the description of which should not be made with the aid of the rule system. This can be done more effectively by applying a feature mechanism which constructs strings of characters (i.e. words) with the aid of certain features found in certain parts of a word. I shall return to this in (2.1.4.).

Useful guidelines of how to proceed in finding out where the two-level rules are appropriate and where the description may be done within the lexicon system are given in Koskenniemi (1991).

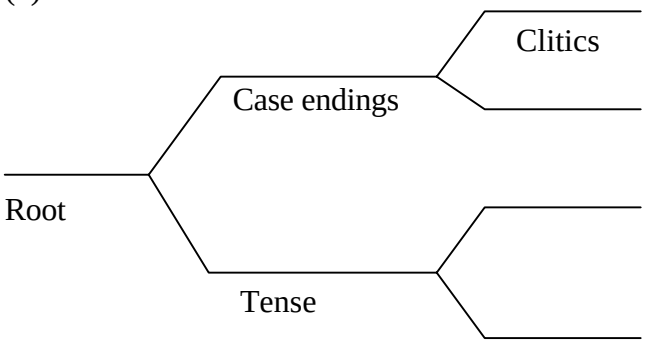
1.2. THE LEXICON SYSTEM

The data for the input of the two-level system are obtained from the lexicon system, which contains all the lexical data of the language needed for the purpose concerned. It may be a restricted lexicon compiled for a specific purpose, or it may be an ambitious comprehensive system containing all significant lexemes of the language.

It depends on the type of language what kind of lexicon system would be most effective in each case. The basic working principle of the program with the lexical data is quite simple. To some extent the structure of the lexicon system resembles a tree. The program always starts from the sub-lexicon termed START, from which the processing continues to all such sub-lexicons where it is allowed to proceed. In a language system which inflects mainly by means of suffixation (instead of prefixes) this START Lexicon would probably be followed by a sub-lexicon containing mainly word roots. From different roots there would then be branches to various sub-lexicons, such as to case endings for nominals, to tense/aspect endings for verbs etc., and to clitics from these[3].

A simple tree would look like this:

(8)



[3] In the implementations of Koskenniemi (1983), Karttunen (1983), and Jäppinen and Ylilammi (1986) the root lexicon is in fact the lexicon of word roots.

In Bantu languages, however, a number of morphemes precede the word root, and the analysis has to go through a complicated network of sub-lexicons in search for strings of characters. This will be illustrated below.

A sub-lexicon has a name and any number of entries. Part of one sub-lexicon of Swahili is given below:

(9) LEXICON P0

si	P2n	"	NEG SG1";
hu	P2n	"	NEG SG2";

(a) (b) (c)

The name of this sub-lexicon is P0 and it has two entries. The lexical representation (a) of the entry is the first part of each entry. This is what the control of the program has to find in order to be able to read the line further. If the string of symbols entered matches with the lexical representation found in the sub-lexicon, the control moves to the second part (b) of the entry. This part defines the range of possible continuations in building up the total morphological form. In the above example the only continuation class is **P2n**, which is a name of another sub-lexicon. It means that if the control encounters the lexical characters {si}, which match with the characters entered, it is allowed to continue only to the sub-lexicon **P2n**. The last part of the entry (c) contains the information to be retrieved in case of match. If the control, for example, encounters the lexical string **si**, the string **NEG SG1** will be retrieved as an output. It means that si is a prefix of a negative verb form in the first person singular.

The control then continues to seek for matching strings of characters in the sub-lexicon **P2n**, and if a string identical with the entered string is found, the search will continue to the sub-lexicons defined in the continuation classes of that particular entry. It should be noted that the continuation classes in all the entries of one sub-lexicon need not be identical. Each entry may have a different continuation according to need. Also the number of continuation alternatives is free. If there is more than one continuation in an entry, they have to be written together as one string, or be represented with an abstract symbol and defined elsewhere (see below 10 and 11) as separate names of sub-lexicons.

1.3. DEFINITIONS

All the characters and diacritics used in the rules and in the lexicon system have to be defined in the rule module. In the present formalism, however, lexical and surface characters need not be defined separately as long as the lexical and surface values are the same. The program takes this as a default case. Only when the surface value is different from the lexical one is there a need for the definition of the values of both levels. Lexical capital letters are typical cases where the values on both levels have to be defined, by using a colon (:) as a separator between lexical

and surface characters, (e.g. **I:i, A:a, N:0**). Note that normally the most common surface realization of the lexical character is defined as a default case, and other realizations are taken care of by rules.

It is often useful to define various sets of characters, such as consonants, vowels, and certain subsets of them. It is more convenient in the rules to refer to a symbol, which represents a group of characters, than to write all of them explicitly into the rules.

The use of a previously defined symbol instead of a string of concrete symbols is a convenient practice also in lexical entries, where one would otherwise have to write a long list of possible alternative continuations from that entry. Verb roots are a case in point, because they have several alternative continuations. Therefore, instead of writing the concrete list of these continuation alternatives, a single symbol may be used. This symbol is then defined in the section where other similar definitions are.

TWOL has undergone quite a number of improvements since 1983 (Koskenniemi 1990, 1991; Karlsson 1990). One change concerns the format of definitions. One simplification was mentioned already, namely the default correspondence of lexical and surface characters. Another improvement is that there is no need to have a separate section for continuations in the beginning of the lexicon system. The definitions are treated as sub-lexicons, where the name of the lexicon is the symbol to be defined, and the entries are the names of the individual sub-lexicons which the symbol stands for.

There are also other modifications to the earlier versions. The lexical entry need not have all three parts (entry, continuation class and output string). If the entry is zero, it may be ignored. If nothing is to be retrieved from that entry, empty quotation marks may be omitted. The program is made to recognize the parts of the entry from the final semicolon backwards recursively. If quote marks are absent, it treats the string preceding the semicolon as a continuation class, and the further preceding string (if found) as a lexical entry. Sub-lexicons containing 'definitions' of symbols used have such one-part entries.

Examples of the types of sub-lexicons where the symbols used in the lexicon system are 'defined':

(10) LEXICON PrepCsStvPsE
Prep; Cs; Stv; Ps; E;

(11) LEXICON Vd
Prep; Cs; Stv; Ps; E;

Two things should be noted in these examples. (a) They represent two alternative ways of using continuation class names. In the first example the names of the sub-lexicons are appended together as a single string and then defined as separate sub-lexicons. In the latter example the short symbol Vd is used to stand for the same set of sub-lexicons. (b) They are examples of sub-lexicons, the entries of which lack

both the lexical representation and the output string (for full three-part entries see 9).

2.0 THE IMPLEMENTATION OF TWOL IN SWAHILI

As described above, the language-specific program is an integration of two modules, the rule component and the lexicon system. The basic principle in constructing the program is that morphemes which do not show morphophonological variation should be handled within the lexicon system alone. Such morphemes are for example the stems of nouns, and also uninflected words, particularly adverbs. The need for rules enters, however, as soon as there is variation in the surface realization of morphemes. There are nevertheless a number of cases where a problem can be solved either by the lexicon system alone, or by a combination of the lexicon system and rules.

Deviations concerning only isolated cases are easiest to handle within the lexicon system. An example of such a case is the infinitive marker {ku}, which is realized as /kw/ only in the verbs *kwenda* and *kwisha*. The problem is handled by directing the continuation from {kw} to the sub-lexicon containing only these two verb roots, while the continuation from {ku} is directed to the sub-lexicon containing the remaining verbs.

An example of a case where rules handle the deviations conveniently is the final {A} of Bantu verb stems. Its three realizations (/a/, /e/ and /i/) can be handled by rules which define the context for each realization. Rules are particularly useful in cases where the number of entries in a sub-lexicon is so large that it is not reasonable to create a duplicate or triplicate lexicon in order to build separate routes for different endings. Rules handle such cases effectively.

There are a large number of cases where the choice between these two possibilities depends on the purpose of the program. If the aim is to build a linguistic two-level description of the language, the system offers excellent scope for describing the lexical representations of surface forms. Various realizations of one lexical form may be effected through applying a set of rules. On the other hand, if the aim is to describe only correct surface forms, the number of rules may be decreased, whereby more use will be made of the possibilities of the lexicon system. Generally speaking, the system facilitates the more linguistically oriented and abstract morphological description and the more transparent Item-and-Arrangement type of description. The description of Blåberg (1984) of Swedish morphology with 35 two-level rules and 13 morphophonemes offers an example of the former type of orientation, while the recent and comprehensive description of Karlsson (1992) with only eight rules is a good example of the latter type.

2.1. THE LEXICON SYSTEM OF SWAHILI

2.1.1. Material of the SWATWOL lexicon

Since machine-readable dictionaries and word-lists on Swahili were not available, the work on compiling the SWATWOL lexicon had to be started from scratch. Fortunately I had already started compiling word lists of recently coined words which were not available in standard dictionaries. The latest Swahili dictionary *Kamusi ya Kiswahili Sanifu* was scanned into computer form. These materials and other available word lists were used in compiling the lexicon system.

The structure of the lexicon, however, was built and tested with a small sample lexicon, which gradually expanded to contain all sub-lexicons needed. After the network of sub-lexicons seemed satisfactory, the actual compilation of word roots followed.

Because Swahili is a prefixing language, and therefore word roots cannot be placed in the beginning of the tree-structure, the roots will be grouped into a large number of sub-lexicons. This process was automated to a large extent by using a number of BETA programs. Dictionaries and word lists were provided with sufficient symbols to be used in excerpting different types of lexical entries. As a result verbs were in one group, nouns of each class in their own groups, adjectives formed one group etc. These groups were, however, only a starting point, and more subdividing and refining was needed.

The danger of entering the same lexeme more than once was avoided by keeping entries in each sub-lexicon always in alphabetical order. There was quite much overlapping in different word lists. The number of lexical entries in *Kamusi ya Kiswahili Sanifu* is about 15.000, and the two compiled word lists of new words contained about 5000 entries each. Double entries with identical readings were eliminated, and also such entries were not accepted which already had been entered although with another semantic meaning. This leaves a total of about 19.000 simplex entries. The verbal derivation and nominal derivation from verbs discussed below increases the number of words to more than 25.000. The number of verbs particularly depends very much on the definition of verb, i.e. whether derived verbal forms are considered as derivations of the base form or as separate verbs. There are 2.530 simplex verbal entries in the lexicon, but with derived forms the number will be several times higher. In fact verbal derivation is open-ended; there is no definite limit to it.

More words into the lexicon system were obtained through the extensive testing process (see below). SWATWOL was made to extract and list unanalyzed word forms from a number of books and newspapers. This work yielded in several hundred new words not found in dictionaries or in lists of recently coined words. This tells about the rapid development of Swahili, and also of deficiencies in word-lists and dictionaries.

2.1.2. General structure of the lexicon

In the implementation described here the lexicon system of Swahili is composed of a rather complicated network of sub-lexicons. The complexity derives from the rich prefixation of almost all word classes, and also from the multiple suffixation of verbs. The system forms a tree, which starts from a root lexicon called START (12), and branches from it into a number of sub-lexicons, most of which in turn have continuations to other sub-lexicons. The idea in constructing this network is to make the program produce and recognize all permitted word forms and prevent it from accepting ungrammatical ones.

(12) LEXICON START
NORM;
#* NORM;
NUM;
#* PROPN;

From START there is a continuation to a sub-lexicon NORM. Access to it is provided also through a capital initial letter, which in a preprocessed text is symbolized by '*'. Continuation is allowed also to NUM (containing numerals) and PROPN (containing proper names, always with the capitalized initial).

(13) LEXICON NORM
VERB; NOUN; ADPRON; KI;

Sub-lexicon NORM has no lexical entries. Its only purpose is to redirect the analysis to four major sub-lexicons, which again redirect the analysis further. The sub-lexicon NOUN is given below.

(14) LEXICON NOUN
M/WA; M/MI; JI/MA; KI/VI; N/N; 0/0; U; Pa; DerN;

These examples show that a sub-lexicon can be used without lexical entries for 'defining' in more detail the contents of the lexicon name. This is particularly useful in lexicons with several entries, each with several alternative continuations. As a continuation may be given only a short symbol, which is then 'defined' in another sub-lexicon concretely. Such defining lexicons have been placed as a separate group before the actual sub-lexicons with lexical entries. After defining lexicons are the sub-lexicons needed for describing verbs. Then follow nouns, adjectives, uninflected words, pronouns, proper names etc.

2.1.3. Verbs

The following example of the Swahili verb structure illustrates how different types of morphemes are located within the string of morphemes. The verb structure may be thought to consist of a definite number of 'slots', which may or may not be filled with appropriate morphemes. One (surface) form of the stem {som} (to read) is segmented below:

(15) *wa-li-cho-tu-som-e-a*

The English gloss of this form is: "that which they read for us". When SWATWOL analyses this surface form (written without dashes) it gives information about each morpheme in the order in which they occur in the string. It is simply a question of convenience how much and what sort of information the program is made to retrieve. In the present implementation the output string looks as follows:

(16) "M/WA 3PL PAST REL KI/VI SG OBJ 1PL VERB PREP"

This is interpreted as follows:

"M/WA 3PL" means that the surface /wa/ is a subject prefix of the noun class called here **M/WA** (less mnemonic class names based on numbers are not used here) and that it is third person plural (**3PL**).

"PAST" means past tense with a marker /li/.

"REL KI/VI SG" indicates that /cho/ is a relative prefix (**REL**), referring to a noun which belongs to the noun class **KI/VI**, and it is singular (**SG**).

"OBJ 1PL" stands for the object prefix of the first person plural.

"VERB" indicates the appropriate part of speech.

"PREP" stands for the prepositional or applicative form, the surface form of which is here /e/. This is then followed by the word-final /a/ with obvious meaning; hence zero output.

The formal structure of Swahili verb with possible slots for morphemes is described below:

(17)

Inflectional prefixes					Root	Derivational suffixes				
NEG	SP	TENSE/ASPECT	REL	OBJ	ROOT	PREP	CAUS	STAT	PAS	REC (etc.)	A

The above schema is appropriate as far as the slots prior to the root are concerned. The order is as described above, although not all of the slots may be filled at any one time. The arrangement of slots after the root, on the other hand, is quite flexible, and some of the derivational suffixes may be applied more than once in the same string.

As described above, the verbal system is rather complicated and requires a number of sub-lexicons. I have grouped the alternative prefixes of the same slot in one sub-lexicon, e.g. subject prefixes comprise one, object prefixes another sub-lexicon etc. In principle the analysis proceeds step by step from the beginning to the end of the given word form, at each point comparing the pair of characters (lexical and surface). If the total process through sub-lexicons is successful, i.e. if all the character pairs match, the string given is accepted and the result is monitored. If the match fails at any point, the string is discarded. Thus the program may be used for checking the correctness of any word form.

The linear processing of verb forms is complicated by the fact that not all tense/aspect forms are allowed to take all types of prefixes. Subjunctive and present tense negative are a case in point. I have solved this problem by creating separate parallel sub-lexicons for facilitating routes through the array of prefixes for subjunctive and present tense negative. All these routes meet in the sub-lexicon of verbal roots, from where a further extensive branching starts to facilitate verbal derivation. There is a total of 22 sub-lexicons of verbal prefixes facilitating various combinations of verbal inflection.

The large number of derived verbal forms, some of which may be applied after an already derived form, and some more than once, requires a network of sub-lexicons, which covers all the possible combinations of derivation. Verbal derivation is in fact one of the instances where this computer formalism shows its power over the traditional ways of description. As Khamisi[4] has shown, it is simply impractical to try to include all the possible combinations of derivation of each verb root in a conventional dictionary. Through a correct structure of the lexicon system each verb root may be given right continuations.

It is clear, however, that it is not useful to handle all verb forms as derivations from the basic verb root, so that only the basic roots are included in the sub-lexicon of the verb roots. Some roots are so productive that verb stems with quite specific meanings are derived from them, e.g.

(18) *enda* 'go' > *endelea* 'proceed' (double applicative)
ona 'see' > *onea* 'bully' (applicative)

[4] I agree with Khamisi (1985, 1988) that the problem of the productivity of different verb roots and the types of combinations allowed to be suffixed after each type of root has not been sufficiently studied.

These are cases where the derived verb stem should be entered as such in the sub-lexicon of verb roots, although in other cases the applicative is handled as a separate suffix (cf. Khamisi 1988, 63-68; Mukama 1978, 26-34).

A similar problem is posed by some derived forms, such as conversive (e.g. **funga** 'tie', 'close' > fungua 'untie', 'open'), contactive (*kama* 'squeeze' > kamata 'take hold of'), and inceptive (*nene* 'thick' > *nenepa* 'get fat' (persons)). A number of regular conversive forms with predictable meanings may be handled as derivatives from the basic root, but those with specific meanings should be entered with the conversive stem.

It must be pointed out that technically it would be possible to get even these problematic derived forms from the basic roots, but the problem concerns the correct output string. The output is composed of the pieces of information given in each lexical entry. Therefore, from the surface form *on-e-a* the output string would be /'to see' PREP/, and it would not convey the idea of 'bully'.

The verbs have a total of 13 different formal derivational patterns. The vast majority of verbs belongs to one of these patterns. The term 'formal' here denotes the morphological criteria alone excluding semantic variables. In the present implementation each verb class has been given its own route through the derivational 'slots', although two or more verb classes might have identical lexemes in the corresponding slot. This is done in order to secure that only grammatical combinations will be realized. As a result a total of 97 sub-lexicons were formed to describe the verbal derivation. More research and testing with experimental material might make it possible to simplify the network of sub-lexicons on the part of verbal derivation to some extent.

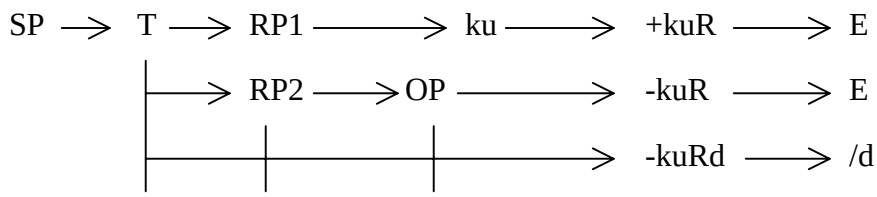
2.1.4. Irregularity of monosyllabic verbs

From the viewpoint of computer implementation the monosyllabic verbs are an interesting case. The rule, verbally expressed, requires that the infinitive marker {ku} precedes the monosyllabic verb stem, if otherwise the stress would fall on a morpheme which cannot take stress. There are no morphological or phonological features, however, in these morphemes to help in formulating the two-level rule. The morphemes that cannot take stress are some (but not all) tense/aspect morphemes {na, li, ta, me, mesha, meisha, nge, ngali}, and relative prefixes.

Because the number of monosyllabic roots is small, a practical solution is to form sub-lexicons of monosyllabic roots and build three separate routes for them, one for the cases with infinitive /ku/ and two for those without. This means establishing three sub-lexicons (19), here termed as +kuR, -kuR and -kuRd, each having the same lexical entries of monosyllabic roots. The forms with /ku/ are made to go via +kuR and terminate with verb-final {A}, because as soon as any suffix is attached, the stem is no longer monosyllabic and the infinitive marker /ku/ may not appear. The forms without /ku/ are directed to the two other root lexicons with identical lexical entries, which, however, have different continuations. One (-kuR) allows only the verb-final vowel (E) as a continuation, and the other (-kuRd) takes

only derived forms (/d) as a continuation. This division is necessary, because derived verb forms (-kuRd) have to have access from the relative prefixes (RP2) and also directly from tense (T). The verb forms (-kuR) without derivation and without /ku/ are directed through the object prefix (OP) sub-lexicon only, because other basic verb forms have to take /ku/.

(19)

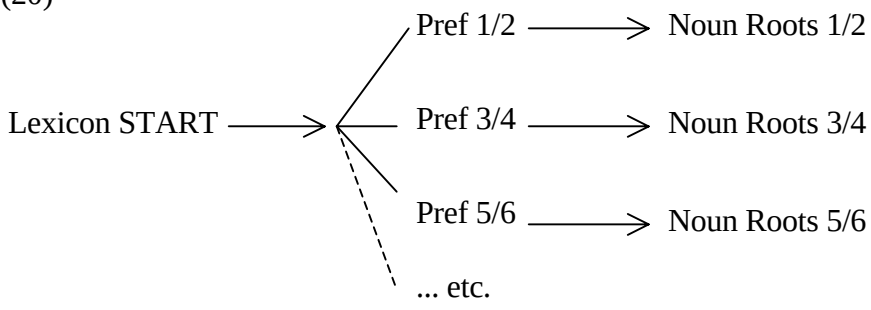


2.1.5. Nouns, adjectives, pronouns and adverbs

Nouns and adjectives are characterized by their noun class prefixes. They are different in that while part of adjectives may have any of the noun prefixes, the nouns may not. Some noun roots (e.g. /ti/ and /tu/) may take prefixes from more than one noun class, with different meanings, and others may be classified alternatively in two noun classes (classes 5/6 [JI/MA] and 9/10 [N/N]) without change of meaning. Inflecting adjectives are given the class prefix of the noun which they qualify.

In the lexicon system it seems practical to form a sub-lexicon for the noun roots of each noun class. Therefore, noun roots which take the prefixes of class 1/2 [M/WA] are located in the same sub-lexicon, and access to it is arranged only from the sub-lexicon consisting of the prefixes of this noun class. The route is directed from the START lexicon to each of the sub-lexicons consisting of the prefixes of respective noun classes, and the route continues from each of these sub-lexicons to the sub-lexicons consisting of respective noun roots. The structure may be illustrated thus:

(20)



In practice, however, most noun classes need more than one sub-lexicon for the roots, some because of irregular morphophonemic variation of the prefix, some because of variations in plural forms. As a result the number of sub-lexicons containing nominal roots is 20 instead of eight.

Adjectives are grouped into three sub-lexicons, because they behave differently. Non-inflecting adjectives may take no prefixes, while access to the other two is arranged from all the noun class prefixes. The prefixes taken by adjectives, although basically identical with noun prefixes, are not subject to the same morphophonological variation as those prefixed to the nouns. Therefore the adjectival prefixes have been listed as a separate sub-lexicon, and access is given to the sub-lexicon of such adjectives which take prefixes. Those numerals which take prefixes have been given a separate sub-lexicon.

The adjectives and numerals of foreign origin which do not take prefixes, are grouped in separate sub-lexicons, as are also adverbs, prepositions and other uninflected words.

Possessive pronouns, the associative /a/, and the special forms /ote/, /enye/ and /enyewe/ have also been given the sub-lexicons of their own. The alternation of the vowel in the demonstrative /h-/ is solved within the lexicon system, because it is a closed class and needs no general rule[5].

2.1.6. The application of a feature mechanism

There are problematic morphemes, which appear as suffixes after a large sub-lexicon of word roots, but which cannot be connected with all possible preceding morphemes found on the route. Such cases cannot be fitted directly into the tree-structure, where each morpheme has a connection to the START lexicon through a series of sub-lexicons.

A typical case is the general relative affix, which is suffixed after the verb-final vowel. It cannot take tense/aspect morphemes or relative prefixes, which appear as optional morphemes in most verbal constructions. In (21) this constraint is illustrated.

(21) a - na - ye - ki - som - a (He who reads it)
a -*na -*ye - ki - som - a - ye (He who reads it)

Another example is the imperatives, which normally take no prefixes, and may allow only the object prefix, in which case with the final {A} realized as /e/.

(22) som - a (Read!)
ki - som - e (Read it!)

[5] It would be possible to implement the stem vowel variation of the demonstrative {h+V} by means of rules. It would require, however, three rules, which is considered here excessive compared with the convenient use of separate sub-lexicons.

Because the number of verb roots is several thousand, it is not practical to build separate routes for each type of case. In handling such cases, a special feature mechanism would be one solution. In it an appropriate and identical feature is attached (a) to an entry, whose presence is conditioned by a feature located after the entry in the lexical string of morphemes, and (b) to an entry whose presence/absence functions as a trigger. For example, the presence of the tense/aspect morpheme is conditioned by the absence of the general relative suffix, while the presence of this suffix triggers the absence of the tense/aspect morpheme.

This may be implemented by attaching a feature to the output strings of the lexical entries of these interdependent sub-lexicons. This feature is given an appropriate value in each context. For example, in the following tense/aspect string:

(23) na P34R "TENSE=PR1";

'TENSE' marks the feature, and 'PR1' is its value. Similarly, in the entries of the relative suffix sub-lexicon there is the same feature, but with a different value, e.g.

(24) ye # "TENSE=NIL RELS";

where the feature 'TENSE' has a value 'NIL'. Because the values of 'TENSE' in (23) and (24) are different, the combination na+ROOT+ye is not permitted.

In a similar way features with appropriate values may be attached to the entries of the sub-lexicon of relative prefixes, e.g.

(25) ye P4R "RELP=M/WA SG";

and to the sub-lexicon of relative suffixes,

(26) ye # "RELP=NIL TENSE=NIL RELS";

The values of RELP are different, and the program discards a string in which both a relative prefix and suffix appear. The feature mechanism does not affect morphemes which do not have features to be tested for matching.

In the present application the feature mechanism described above has not been applied. Instead these special cases have been handled by means of context restriction rules. These rules have been discussed below.

In general terms, this is the structure of the lexicon system. It has to be supplemented, however, with the rule component, so that morphophonological alternation processes may be handled. It is necessary if we want to meet the requirement that all correct forms and none of the ungrammatical ones will be produced.

2.2 SWAHILI RULES

In the implementation discussed here I have included more rules than necessary, for reasons to be described below. Swahili nouns are good illustrations of how the same surface representation may be achieved either through the lexicon system alone or by a combination of the lexicon and rules.

For example in noun class 7/8 [KI/VI] the singular prefix {kI} is realized as /ki/ in front of consonant-initial roots, and otherwise it is supposed to be realized as /ch/. The plural {vI} is realized as /vi/ and /vy/ in similar environments. In practice, however, only in 24 of the vowel-initial noun roots the prefix underwent permutation and 134 vowel-initial roots received the unaltered prefixes /ki/ and /vi/. It would be possible to handle the situation through the lexicon system alone by placing the few vowel initial roots receiving permuted prefixes in a separate sub-lexicon, and by arranging access to them only from the permuted prefixes {ch} and {vy}. What we lose by so doing is the correct lexical description of the prefix. By using rules we may enter only the base forms {kI} and {vI} into the sub-lexicon and handle the alternations through rules.

The above problem applies also to vowel-initial adjectives, some of which take permuted prefixes. Because it is not possible to define exclusively the phonological environment where the permutation takes place, a useful solution is to locate these rule-violating stems (nouns and adjectives) in separate sub-lexicons and give access to them from the lexical prefix forms {ki} and {vi} instead of {kI} and {vI}.

A good example of the effectiveness of the rule system is provided by noun class 9/10 [N/N] , which has the same prefix in singular and plural. The prefix is stated by most authorities to be {N} (Polomé 1967, Ashton 1944, Schadeberg 1984), which is also realized as /m/, /ny/ and /0/. On this assumption, the /ny/ realization in front of the vowel initial stems is, however, problematic. There is no (morpho)phonological reason for such a realization, because in similar environments /n/ may be followed by a vowel. The problem may be solved by assuming that the lexical form of the prefix is {NI}, which is found in Proto-Bantu (Doke 1943, 20; Meinhof 1910, 38), and which is obviously the 'original' Swahili form also (Madan 1884, 18; Gregersen 1977, 79; Bakari 1985, 94-95)[6].

In this formalism all four realizations (/n/, /m/, /ny/ and /0/) have been derived from the lexical form {NI} by using appropriate rules. This is useful in describing phonological processes and it also simplifies the lexicon system, because there is no need to make sub-lexicons for roots taking different realizations of the prefix. The treatment of inflecting adjectives would lead to immense complexity without rules, because they are allowed to be connected with prefixes of any noun class. In this application all class prefixes are allowed to be connected with all inflecting

[6] If we take the Proto-Bantu form as a base form in all Bantu languages, we can note that in none of Meinhof's six test languages is the prefix {*NI} realized as such. In all of them it has dropped {I}, and {N} is realized as /in-/, /im-/, /i-/, /o-/, or /ozo-/ in languages with a pre-prefix, and as /0/ in others (excluding special cases of bilabial consonants and vowels). Guthrie (1967-71 II, 144) gives it the base form {*NY}, indicating its actual surface form in most cases.

adjectives, and deviations are handled with rules. Below (27) are realizations of noun class 9/10 prefixes, with the lexical form above and the surface realization below. Note the character-by-character equivalence.

(27)

```
#NIzige#            #NIdege#            #NIgoma#
0n0zige0 'locust'   0n0dege0 'bird'     0n0goma0 'drum'

#NIbegu#            #NIvi# 'grey        #NIpya#
0m0begu0 'seed'     0m0vi0  beard'      0m0pya0  'new'

#NItaa#             #NIsaa#             #NIkuku#
000taa0 'lamp'      000saa0 'watch'     000kuku0 'hen'

#NIundo#            #NIumba#            #NIeupe#
0nyundo0 'hammer'   0nyumba0 'house'    0nyeupe0 'white'
```

The total elision of the prefix {NI}, when followed by a voiceless consonant, would be explained in generative phonology by applying two subsequent rules, where {I} is first realized as zero, and {N} then elided in this new environment. Although this order may be thought of also in this two-level formalism, the actual working of the rule component takes place as one operation.

The development of the language has led to the expansion of the noun class 9/10, because most of the new nouns are located in this class. They are also generally without noun prefix; i.e. {NI} is realized as /0/. It is becoming increasingly doubtful whether the base form of the prefix of these new nouns, many of which are loan-words, should be postulated as {NI}. In fact, in order to make the description more transparent, and in order to make it represent the actual situation, I have located these new roots in a separate sub-lexicon with 0-prefix.

Often it is useful to capitalize those letters in the lexical representation which are subject to alternation. In so doing it is possible to define precisely the context where the rule applies. It is very unusual in Swahili that {i} is realized as zero when followed by a consonant. This is true, however, in the case of the noun class 9/10, where the almost obsolete {i} is elided except when followed by a vowel. The capital letters **NI** help to specify the context, so that the rule does not destroy {i} in such contexts as 'nina', 'alinipenda'.

Capital letters in the lexical representation have been used whenever there is a danger that otherwise the context conditions for the rule would not be precise enough. Examples of such lexical entries are: {mU, wA, lI, yA, I, kI, vI, kU, pA}. An interesting case is the prefix {mU}, which appears as a nominal and adjectival prefix of the noun classes 1, 3 and 18, but also as a subject prefix of 2PL and of noun class 18. On the surface {U} is normally realized as zero, but when followed by a vowel it is realized as /w/. I do not see any reason why this prefix should be represented as a mere {m}, as is often done, because such a representation does not provide explanation for the appearance of /w/ in front of a vowel. It is more correct,

and again in tune with a number of other Bantu languages, to assume the existence of the lexical {mU}.

In the lexicon system, {mU} appears also as an object concord of 3SG instead of {m}. This interpretation makes it possible to explain why /w/ appears after /m/ in front of vowel-initial verbs. Polomé (1967:75) explains it to be part of the old verbal root in {ona} < {*wona}. This does not explain, however, why /w/ also appears in such constructions as alimwambia, alimwepa, alimwiga, alimwuguza etc. By postulating the lexical form {mU}, we can interpret /w/ as a realization of {U}.

In the present version of SWATWOL there are 18 rules, three of which are double rules with only a partial function. Kimmo Koskenniemi has advised me in several phases in constructing these rules, particularly during the time when the automatic rule compiler was not yet available to me. The uncompiled rules are given in Appendix 1.

2.3. Use of Diacritics in Rules

Somewhat problematic are alternations whose defining features are located at a distance from the place of alternation, or which do not have otherwise enough defining features. Such a case is the verb-final {A}, which has realizations /a/, /e/, and /i/. What are the defining features of the final /e/ in the subjunctive? One characteristic feature is the absence of the tense/aspect marker, but because it is absent also in the habitual, imperative, infinitive and present tense negative, it is not possible to refer to this feature in the rule. Nor is there any phonological feature which could cause this alternation. In such a case I have used a special symbol in the lexicon, to which reference is made in the rule (rule 7 above). For the alternation {A} > /i/ in the present tense negative I have used another symbol (rule 8), and so on. Such diacritics are always realized as zero, but they are useful in defining rules. Examples of the use of diacritics:

(28)
```
#^nIsomA#
00nisome0  " 'I may read' SG1  SBJN "
```

(29)
```
#!hawAsomA#
00hawasomi0 " 'They do not read' PL3  NEG PR "
```

(30)
```
#nI/asomA#
0n00asoma0  " 'I read'  SG1  PR2 "
```

In the last example I have used the diacritic {/} (rule 3) to define the context: /a/ on the right is not sufficient, because in similar environments {I} is realized also as /i/ and /e/ (rule 9, cf. Polomé 1967:84).

In discussing the lexicon system, a need for restricting the combination of certain morphemes was stated. Because there is no feature mechanism in the system, such restrictions are effected by means of context restrictions rules. Such

rules require the use of diacritics in the lexicon to trigger the co-occurrence of desired morphemes. A pair of rules as shown in (31) handles the restrictions of general relative (rules 15 and 16 in Appendix 1).

(31) %+: => _ :* %&: ;
%&: => %+: :* _ ;

The diacritic /&/ placed as part of the morpheme strings of general relative causes the realization of those strings only, which are preceded (in any distance) by the diacritic /+/. As the diacritics are realized as zero on the surface by definition, they do not affect the realization of other permitted forms.

The derivation of certain noun types from verbal roots is also implemented by means of context restriction rules (rules 17 and 18). Verbal roots are allowed to take nominal prefixes of the classes 1, 2 and 11 and a nominal suffix /ji/ after the verb-final /a/. This is the most productive type of nominal derivation. Allowing all verbs ending with /a/ to take these affixes results in slight overgeneration. Yet the formations are grammatical although not presently in common use. This facility decreases the number of entries in lexicon by thousands of lines and decreases memory requirements.

2.4. ALTERNATION PATTERNS

There are alternations in Swahili which do not have phonological causes and which therefore cannot be accounted for by rules. Such alternations are found particularly in the derivation of verb roots of non-Bantu origin. In derivation the verb-final {A} is transposed to the end of the derived form, e.g.

(32)	som+A	>	som-esh+A
	read		make to read

Some non-Bantu verbs retain the verb-final vowel (which is not {A}) in derived forms and add the final {A} at the end of the derived form.

(33)	rud+i	>	rud+i-sh+A
	come/go back		make to come/go back

(34)	kod+i	>	kod+i-sh+A
	rent		rent out

Such verb stems may be entered into the lexicon with the final {i} suffixed to the stem, and an alternative end morpheme given with zero realization. This will allow the basic form to end without final {A}, but forces all derived forms to go through the route where the final {A} is compulsory. For example, we may have the following kind of entries with a continuation to a sub-lexicon which has alternative

continuations: one which allows the string to be terminated, and the other which allows derivational forms.

(35) rudi (i)/V "to come/go back";

```
LEXICON (i)/V
#  #;
CsStvPrepPsRec;
```

There are a number of such irregular derivation patterns, which can be handled in the same way. Also derivational forms have irregularities, such as the occurrence of lexical {l} on the surface (e.g. /tembea/ > /tembe-l-ea/; /kaa/ > /ka-l-ia/ > /ka-l- i-w-a/), and the alternation /sh|z/ in causative. Some of the regular alternations can be handled by rules, but unexpected realizations can be safely dealt with through alternation patterns.

But here too there is often more than one solution. For instance, the derivation of vowel-final verb roots may be implemented in two ways. It is possible to assume a base form

(36) kaL (instead of ka) 'to sit' > kaL-I-A > kaL-I-sh-A > kaL-I-w-A

whereby the alternation is of the type /L:l | L:0/, which can be handled either through a lexical alternation pattern or through a rule. The other way is to assume a base form

(37) ka > ka-l-I-A > ka-l-I-sh-A > ka-l-I-w-A

where word formation is directed through a sub-lexicon, which gives alternative routes:

(38) ka l/V " to sit";

```
LEXICON l/V
E;
l CstPsPrep;
```

This sub-lexicon gives to non-derived verb forms the realization without /l/, and inserts /l/ in derived forms.

3.0 PREPROCESSING AND TEST RESULTS

In order to facilitate the accuracy of processing, the raw Swahili text has to be preprocessed with a program PREPROC.SWA written in BETA. The program

performs such functions as: transforms word-initial capital letters into a sequence of an asterisk and the appropriate lower case letter; dissociates syntactic and other punctuation marks from letters (and from each other if needed), but does not affect such periods which are part of an abbreviation, such as k.mf. ('for instance'); reconnects parts of words divided on separate lines by hyphenization.

The preprocessing program recognizes the distinction between temporary upper case words, such as titles of texts, and such words, which are always written in upper case (several abbreviations of companies and organizations, e.g. OAU, TAZARA, FRELIMO, ANC, FINNIDA).

The program reduces the former type words into lower case and places an asterisk in front of the first letter. This process takes place provided that all the letters of the word are written in upper case. If one or more letters in such words are written in lower case, this indicates a typing error, and the program spots it.

The letters of upper case abbreviations are transformed into lower case and each letter is preceded by an asterisk. SWATWOL identifies such abbreviations written in the lexicon and discards abbreviations which are only partly written in upper case in the original text. An example of preprocessing is given below: (a) original text, (b) processed text.

(a) JANA J. Nyerere, Rais wa Tanzania, alifungua TAZARA.
(b) *jana *j *nyerere , *rais wa *tanzania , alifungua *t*a*z*a*r*a .

PREPROC.SWA presupposes that in the text given for preprocessing, paragraph boundaries are marked with the character /@/.

In the following are results of tests carried out with four kinds of Swahili text. MZAL is a collection of leading articles from the weekly Tanzanian newspaper MZALENDO (1990). CHEZO contains news reports from sports events, excerpted from the same newspaper. ISIM contains scientific text (linguistics) from a conference report published by the Institute of Kiswahili Research, Univ. of Dar-es-Salaam (1983). BIN is part of the book *Binadamu na Maendeleo*, written by Julius K. Nyerere. These were 'new' texts to the program in the sense that they had not been used in the process of testing and correcting. The texts were preprocessed with a BETA program and then transformed as a list with one word per line.

The resulting list of unanalyzed word forms is composed of three kinds of tokens: (1) misspellings; (2) correct Swahili word forms but either too rare to be included into the lexicon, or words of foreign languages (mainly English); (3) words which should be in the lexicon. Below are test results.

MZAL has a total of 5670 word form types. The program left 196 word forms unanalyzed. These were considered to belong to the following groups: group 1: 36 (14%); group 2: 132 (74%); Group 3: 28 (12%).

The rather large number of unanalyzed words (196, 3.5% of the total) is mainly due to several place names and personal names, which were not included into the lexicon. The number of foreign or rare words in this sample is limited. Group 3 contains words which perhaps should be included into the lexicon. There is no single word with high frequency in normal text. The derived root /wekez/ from the verbal root /wek/ is not found in standard dictionaries, neither is /vitega/. The former appears 14 and the latter 7 times. The number of different word forms in group 3 is 9.

CHEZO has 4943 word form types, and the number of unanalyzed forms was 306. These are grouped as follows: group 1: 35 (12%); group 2: 265 (86%); group 3: 6 (2%).

The number of unanalyzed words in this group is high (306, 6.2% of the total), largely due to the excessively large number of names of players and teams. The text deals with sports events, with a considerable number of foreign (English) terms and names of several nationalities. These should not be included into the lexicon. Group 3 contains 6 word form tokens, with four different word forms.

ISIM has 4028 words, and 81 of them were unknown to the program. These were subdivided as follows: group 1: 35 (38%); group 2: 45 (61%); group 3:1 (1%).

In this sample the number of unanalyzed word forms was rather low (81, 2% of the total). There were 35 misspellings and 45 rare word forms, most of which were names and foreign words. Only one word /kujibizana/ was in group 3.

BIN has 5028 word form types, and only 26 (0.52%) of them were unanalyzed. Almost all of them were misspellings, as the following figures show: group 1:23; group 2:2; group 3:1.

Text BIN represents normal prose extracted from *Binadamu na Maendeleo,* written by President Nyerere. As the results show, the program is very well capable of analyzing normal prose text without rare names and exceptional word forms. The high percentage of misspellings in the text is due to the scanned text, which was not properly edited.

As a sum total of the results of these four texts we get the following breakdown:

Total no. of word forms	19669
Unanalyzed total	609 (3.10%)
Misspellings	129 (0.66%)
Rare words	444 (2.26%)
Should be included	36 (0.18%)

As the results show, normal prose texts and scientific (linguistic) texts were analyzed satisfactorily. News texts with rare names and foreign words were more problematic. The number of proper names in the lexicon is 1.300 (plus 500 derived

names), and it should perhaps be increased to some extent. Reduplication, which is quite common in Swahili, deserves more attention. So far the commonly occurring reduplicated forms have been entered into the lexicon as such. Obviously in some prose texts the majority of unrecognized proper word forms will be reduplications. The danger of overproduction is however imminent if reduplication is implemented by simple morpheme combination.

Even without additions made after these tests the recall is above 99.8% in average prose texts. This corresponds closely to the recall of 99.7% of SWETWOL, the morphological analyzer of Swedish (Karlsson 1992).

4.O The Applicability of the Model

Karlsson (1992) has stated a number of potential applications for SWETWOL. Most of them apply also to SWATWOL, although profound differences in language types may give reason to emphasize applications in different ways.

The most ambitious goal is to develop a syntactic parser, where SWATWOL is used as a morphological module. This analysis gives a basis for carrying out disambiguation processes of readings, clause-boundary determination, as well as surface-syntactic analysis, with a program based on the Constraint Grammar Parser (Karlsson 1989, 1990, 1991).

A version of SWATWOL with the lexical entries of *Kamusi ya Kiswahili Sanifu* has been designed. This version is being used for analysing texts, and it filters out words not included in that Swahili dictionary. These wordforms are then extracted from the relevant texts with appropriate context. This version is being used in an automated data finding process for expanding and improving the second edition of the dictionary.

SWATWOL is being used for tagging the DAHE (Dar-es-Salaam - Helsinki) corpus of 1 million words of Standard Swahili text. It will also be used as an analyzer prior to the actual searching operations in normal untagged text. It expands vastly the possibilities of text excerption and increases accuracy.

From the viewpoint of normal text processing, the application of SWATWOL as a spell checker (without output strings of grammatical features) must be considered most useful.

Applications of SWATWOL may be used in language teaching, particularly in training complicated verb forms. Applications covering phrase structures can be used in practicing concordances and correct word orders. This possibility is discussed below.

4.1 Extension to the Analysis of Phrase Structures

So far we have been discussing phenomena within the so-called 'word', which is the traditional domain of morphology. It is obvious, however, that in some cases the word boundary is quite arbitrary. There would be grounds for dividing the verb

construction into two words, whereby the prefixes would be separated from the root and its suffixes, as indeed has been done in some orthographies of Bantu languages. Keeping this possibility in mind, we have actually already crossed the word boundary and combined two 'would-be' words as a single unit.

Although the present formalism does not work as a syntactic parser, it allows for the combination of phrases as single units, which makes their morphological analysis possible. The principle of noun agreement gives good reason for such extensions. In fact I have developed implementations where words in noun phrases are treated in rather the same way as morphemes are treated in words. Such units of analysis may be nouns with their different combinations of qualifiers, or verb phrases. As within the word some morpheme combinations are permitted and others are ungrammatical, so are the combinations of qualifiers within a noun phrase.

The simplest implementation of such a noun phrase parser may be effected by using some kind of connector between words and giving them a lexical representation in appropriate sub-lexicons. The lexicon system with possible continuations from one type of word to another is structured to meet the possible combinations. An example of the lexical and surface strings:

(39) `mUtoto_Uangu_mUzuri_yule`
`m0toto_wangu_m0zuri_yule`
`child my good that`

which on the screen appears as:

(40) *mtoto_wangu_mzuri_yule*

In the mode of analysis, this last string when typed on the keyboard would result in an output stored in the output string of each lexical entry. The output strings of combinations of words are inevitably lengthy, if all morphological information is to be retrieved. It would look like the following:

(41) *mtoto_wangu_mzuri_yule*
"M/WA SG NOUN M/WA SG POSPRON M/WA SG ADJ M/WA SG DEM-L"

It is possible to implement the output with less or more information according to need.

All the permitted combinations of words may be allowed to be realized. The above words could also have the order:

(42) mtoto_wangu_yule_mzuri

which is a permitted order and gives emphasis to the demonstrative 'yule', but the following forms are ungrammatical:

(43) *mtoto_yule_wangu_mzuri
*mtoto_mzuri_yule_wangu, etc.

The control in preventing the realization of ungrammatical forms is effected through the continuations from sub-lexicons.

4.2. FURTHER PERSPECTIVES

1. Special versions of SWATWOL are useful in teaching Swahili word forms and noun phrases, which are sometimes quite complicated. In a computer classroom students are given tasks to form word forms, which gradually become more complicated. With the aid of the noun phrase parser it is possible to drill various types of phrases, whereby the correctness of the concordance can be tested. Unlike most computer-aided language teaching programs, this one allows great flexibility and a lot of space for imagination in its use. It is not restricted to a strict order of procedure, nor to a small set of words and forms. Depending on the size of the computer, the program may have a variable number of lexemes in its lexicon, normally all commonly used words. To help students, I have also created dictionaries (English-Swahili and Finnish-Swahili) within the same lexicon system, for quick access to the Swahili word root in case one does not remember it.

It is also possible to implement the program to function in the generating mode. There are in fact two forms of generation, one that generates the surface form when given the lexical string of characters, and the other that generates the correct word-form when given the morphological information. The former facility is built-in within TWOL. The latter needs an additional module that makes it possible to read the output strings of the lexicon as input and give the surface string as an output. For example, the input

(44) SG1 PR1 OP=KI 'to read' would give an output

ninakisoma

The input string looks somewhat complicated, because it contains the necessary analytical information. For teaching purposes this is, however, useful, provided that the codes used for morphological description are known in advance.

2. It is possible to implement the formalism in the form of a dictionary, which, in addition to giving a gloss in another language and possible synonyms, gives a full morphological analysis of the formative. It is particularly useful as a quick reference dictionary both for retrieving word roots and for revealing the morphological structure of a word form. The problems involved in deciding on the principles of lexicography are well known (Khamisi 1987; 1988). The computer program discussed above overcomes these problems.

3. The two-level formalism makes it possible to study word forms more comprehensively than is the case with the traditional methods. There are innumerable word forms in Swahili which have several morphological interpretations. This is particularly true of verb forms with a set of prefixes and suffixes. Whereas the human mind normally registers only one or two of the possible interpretations at a time, the program registers them all. It also shows how underneath the same surface form there may be different lexical representations.

4.3. Border Problems Between Morphology and Syntax

Although the program is effective in analyzing the underlying lexical forms, it is not well suited to the analysis of such derivational verbal forms, which carry syntactical features. The differences in the kind and degree of productivity of various roots add to the complexity of the lexicon system, but they can be contained within it. Another problem, and a more difficult one, is the variation in the semantic contents of the same derived form, which may have several meanings depending on the syntactic structure. For example, the rule of -I-marking (if we talk in terms of case frame/case role) has at least seven case roles (benefactive, instrumental, goal, locative, reason, time and manner; Mukama 1978), all of them expressed by one derived form normally termed as 'prepositional' or 'applicative'[7]. Formally such forms belong to morphology, but because they semantically are part of syntax (cf. Khamisi 1985, 321-323), their correct semantic representation within the two-level formalism is problematic, and needs further investigation.

* I am grateful to Kimmo Koskenniemi for close cooperation and advise in developing the computer program discussed in this paper. He has also kindly read through this paper and given useful comments on it.

Appendix 1

Alphabet
a b c d e f g h i j k l m n ' o p q r s t u v w x y z
1 2 3 4 5 6 7 8 9 %- A:a I:i L:l N:0 U:u
#:0 %~:0 @:0 %':0 %. %, %*:* ;

Diacritics
%! %/ %^ %$ %+ %& %` %? %{;

[7] Similar problems are found also with some other rules of marking (Mukama 1976; 1978, 35-40). To the same group of forms with morphosyntactic problems could be counted also /kwa/ (Massamba 1985), which is ambiguous both on the morphological and semantic level.

```
Sets
 C = b c d f g h j k l L m n ' p q r s t v w x y Y z ;
 Cn =  c d f g h j k l L  n '  q r s t  w x y Y z ;
 V = a A e i I o u U ;
 Vi = a A e o u U ;
 Vo = a e i u ;

Definitions

Rules

" (1) Devocalization of I when followed by a vowel"
        I:y  <=>  [#: | %*:] (:v) _  [a: | e: | o: | u:] ;
                  [#: | %*: | v:]  _  %/: ;
                              N:  _  :V ;
                   [#: | %*:] :m  _  :* %': ;

" (2) Reciprocal assimilation of A+i > e (partial rule)"
       i:e  <=>  [m   | p | w] A:  _  C: ;

" (3) Elision of lexical I"
      I:0  <=>  [j | k:k | l | m | v]  _  i: ;
                        [l: | n: | z:]  _  %/: ;
                                N:  _  C: ;
                             [z | l]  _  [a | e | o] ;

" (4) Elision of lexical A in subject prefix followed by a vowel"
                           A:0  <=>  _  [a: | %/:] ;
                          [m | w | p]  _  [a | e: | i:] ;

" (5) kI > affricate (partial rule)"
               k:c  <=>  [#: | %*:]  _  I: :Vi ;

" (6) kI > affricate (partial rule)"
          I:h  <=>  [#: | %*:] k:  _  :Vi ;

" (7) A-final > e in subjunctive"
                A:e  <=>  %!: :*  _  [#: | j] ;
                                     _  %$: ;

" (8) A-final > i in negative present"
                A:i  <=>  %^: :*  _  #: ;

" (9) I > e in derived verbal forms with stem vowels e and o"
```

```
  I:e  <=>  @: :* [e | o] (C:) C:  _  [A: | :s :h | :z | :w | k:] ;
                @: :* :e [:k | :s :h]  _  [A: | :w] ;

" (10) Elision of lexical U in concords"
                      U:0  <=>  m  _  ( @: ) C ;
                   [#: | %*:] [m | k]  _  :o ;

" (11) Devocalization of U in front of vowels"
                      U:w  <=>  k  _  :Vo ;
                        [#: | %*:]  _  :V ;
                                  m  _  (@:) :Vo ;
                                  m  _  [@: | %':] :o ;
                                   t  _  %/: a: ;

" (12) Class prefix N > m (partial rule)"
              N:m  <=>  [#: | %*:]  _  :0* [:b | :p | :v] ;

" (13) Class prefix N > n "
              N:n  <=>  [#: | %*:]  _  I: [j: | :d | g: | z: | :V] ;
                                       _  I: Cn:* V: [#: | %$:] ;

" (14) Root initial r > d and l > d in noun class 9/10"
              [r:d | l:d]  <=>  N: I:  _  ;

" (15) General relative 1"
                           %+:  =>  _  :* %&: ;

" (16) General relative 2"
                    %&:  =>  %+: :*  _  ;

" (17) Noun derivation from verbs 1"
                           %{:  =>  _  :* %?: ;

" (18) Noun derivation from verbs 2"
        %?:  =>  %{: [:u | m | w] :* _  ;
```

APPENDIX 2

An example of how SWATWOL analyzes the sentences:

1. *Mwalimu yule aliyekwishatusomesha amekwenda dukani kununua vitu* (That teacher who has already made us to read has gone to the shop to buy things).
2. *Taa zile zote zimezimika* (All those lamps have gone out).

Sentence 1:
mwalimu
"#mUalimu#" M/WA SG
yule
"#yule#" M/WA SG DEM 'that', 'it'
aliyekwishatusomesha
"#Aliyekwishatu@somIshA#" M/WA SG3 PAST M/WA REL SG COMPL M/WA OBJ PL1 CAUS VERB
amekwenda
"#AmekwendA#" M/WA SG3 PERF VERB
dukani
"#duka$ni#" JI/MA SG LOC
kununua
"#ku@nunuA#" INF VERB
vitu
"#vItu#" KI/VI PL

Sentence 2:
taa
"#taa#" UNINFL ADJ
"#taa#" U PL
"#taa#" N/N Class, no prefix
zile
"#zile#" N/N PL DEM
"#!zilA#" SBJN U OBJ PL VERB
"#!zilA#" SBJN N/N OBJ PL VERB
"#!zilA#" SBJN U PL VERB
"#!zilA#" SBJN N/N PL VERB
zote
"#zIote#" U PL
"#zIote#" N/N PL
zimezimika
"#zIme@zimIkA#" N/N PL PERF STAT VERB

REFERENCES

Ashton, E. O., 1944.
Swahili grammar (including intonation). London: Longman.
Bakari, M. 1985.
The Morphophonology of the Kenyan Swahili Dialects. Berlin: Dietrich Reimer Verlag.

Blåberg, O. 1984.
Svensk böjningsmorfologi: en tvånivåbeskrivning. Master's Thesis, Department of General Linguistics, University of Helsinki.
Butler, C. 1985.
Computers in Linguistics. Worcester: Basil Blackwell.
Dalrymple, M., Karttunen, L., and Shaio, S. 1987.
A Morphological Analyzer using Two-Level Rules. In: Dalrymple, Kaplan et al. 1987. Tools for Morphological Analysis. Stanford.
Dalrymple, M., Kaplan, R. M., Karttunen, L., Koskenniemi, K., Shaio, S., and Wescoat, M., 1987.
Tools for Morphological Analysis. Stanford: Center for the Study of Language and Information. Report No. CSLI-87-108.
Doke, C. M. 1943.
Outline grammar of Bantu. Johannesburg.
Gazdar, G., Pullum, G. K., Carpenter, R., Klein, E., Hukari T. E. and Levine R. D. 1988.
Category Structures. **Computational Linguistics** 14:1, 1-19.
Gregersen, E. 1977.
Language in Africa. An Introductory Survey. New York: Gordon and Breah.
Guthrie, M. 1967-1971.
Comparative Bantu I-IV. Amersham: Gregg International Publishers.
Jäppinen, H. and Ylilammi, M. 1986.
Associative Model of Morphological Analysis: An Empirical Inquiry. **Computational Linguistics** 12:4, 257-272.
Karlsson. F. 1989.
Parsing and Constraint Grammar. Unpublished Paper, Department of General Linguistics, University of Helsinki.
1990 *Constraint Grammar as a Framework for Parsing Running Text.* H. Karlgren (ed.), Papers presented to the 13th International Conference on Computational Linguistics. Helsinki. Vol. 3:168-73.
1991 *The Constrain Grammar Parser CGP*. F. Karlsson et al (eds.), Natural Language Processing for Information Retrieval Purposes. SIMPR Document No. SIMPR-RUCL-1990-13.4e. Department of General Linguistics, University of Helsinki.
1992 *SWETWOL: A Comprehensive Morphological Analyzer for Swedish.* (Forthcoming).
Karttunen, L. 1983.
KIMMO: A general morphological processor. **Texas Linguistic Forum** 22, 165-186.
Karttunen, L., Koskenniemi, K., and Kaplan, R. M. 1987.
A Compiler for Two-level Phonological Rules. In: Dalrymple, Kaplan et al 1987. *Tools for Morphological Analysis*. Stanford.

Khamisi, A. M. 1985.
Swahili verb derivation. PhD. Dissertation (University of Hawaii). Mimeo.
1987 *Trends in Swahili Lexicography*. **Kiswahili** 54:1-2, 192-201.
1988 *Relation between Grammar and Lexicon*. **Journal of Asian and African Studies** 35:55-72.
Koskenniemi, K. 1983.
Two-level Morphology: A General Computational Model for Word-Form Recognition and Production. Department of General Linguistics. University of Helsinki. Publication No. 11.
1990 *Finite-State Parsing and disambiguation.* H. Karlgren (ed.), Papers presented to the 13th International Conference on Computational Linguistics. Helsinki. Vol. 2:229-32.
1991 *Discovery Procedure for Two-Level Phonology*. L. Chignoni and C. Peters (eds.), Computational Lexology and Lexicography. Special Issue, dedicated to Bernard Quemada, Vol. I, pp. 451-465.
Lass, R. 1984.
Phonology. Cambridge: Cambridge University Press.
Madan, A. C. 1884.
A handbook of the Swahili language. London: S.P.C.K.
Massamba, D. P. B. 1985.
The semantic and morphological characterization of KWA in Kiswahili. **Kiswahili** 52:1-2, 73-92.
Matthews, P. H. 1974.
Morphology. Cambridge: Cambridge University Press.
Meinhof, C. 1910.
Grundriss einer Lautlehre der Bantusprachen. Berlin: Dietrich Reimer Verlag.
Milne, R. 1986.
Resolving Lexical Ambiguity in a Deterministic Parser. **Computational Linguistics** 12:1, 1-12.
Mukama, R. G. 1976.
Toward the anatomy of the object selection prohibiting rules in Swahili. **Kiswahili** 1976:1, 6-25.
Mukama, R. G. 1978.
On 'prepositionality' and 'causativity' in Swahili. **Kiswahili** 48:1, 26-41.
Polomé, E. C. 1967.
Swahili Language Handbook. Washington, D.C.: Center for Applied Linguistics.
Ritchie, G. D., Pulman S. G., Balck, A. W. and Russell G. J. 1987.
A Computational Framework for Lexical Description. **Computational Linguistics** 13:3-4, 290-307.
Sato, P. T. 1988.
A Common Parsing Scheme For Left- and Right-Branching Languages. **Computational Linguistics** 141, 20-30.

Schadeberg, Th. 1984.
A Sketch of Swahili Morphology. Dordrecht: Foris Publications.
Steere, E. 1870.
A Handbook of the Swahili Language as spoken in Zanzibar. London.
Tomita, M. 1987.
An Efficient Augmented-Context-Free Parsing Algorithm.
Computational Linguistics 13:1-2, 31-46.