Technical Reports in Language Technology
Report No 6, 2009
http://www.njas.helsinki.fi/salama

Two-phase Implementation of Morphological Analysis

Arvi Hurskainen
Institute for Asian and African Studies, Box 59
FIN-00014 University of Helsinki, Finland
arvi.hurskainen@helsinki.fi

Abstract

SALAMA (Swahili Language Manager) is a comprehensive computational system for manipulating Swahili text. It has currently two implementations. The older implementation is based on two-level description with a rule module for handling morpho-phonological alternations. The newer implementation makes use of regular expressions, and in it the morphological description is achieved in two phases. This report describes how the latter approach was implemented.

Keywords: language technology, finite state methods, two-phase method, regular expressions, machine translation

Abbreviations

ADJ	adjective
A-INFL	inflecting adjective
CAP	word with capital initial letter
DEM	demonstrative pronoun
IDIOM-V>	verb member in idiomatic expression
<IDIOM	member in idiomatic expression
MT	machine translation
MWE	multi-word expression
MW-N>>	member in multi-word expression (noun) with two other members on the right
<<MW	member in multi-word expression with two other members on the left
<MW>	member in multi-word expression with one member on right and left
N	noun
OBJ	object prefix
PAST	past tense
PERF:me	perfective tense with the marker *me*
POSS	possessive pronoun
PR:na	present tense with the marker *na*
PRON	pronoun
REDUP	reduplicated form
REL	relative prefix
SALAMA	Swahili Language Manager
SP	subject prefix

SVO	transitive verb
TAM	tense/aspect/mood marker
VE	verb end
VFIN	finite verb form
@FMAINVtr+OBJ>	finite transitive main verb with overt object
@FMAINVtr-OBJ>	finite transitive main verb without overt object
@OBJ	object of the clause
@SUBJ	subject of the clause

1 Introduction

SALAMA (Hurskainen 2004b) was designed for manipulating unrestricted Swahili text as it occurs in written form in books, news media, and currently increasingly in the Web. No manual preprocessing is assumed for the text given as input to SALAMA. The system itself, however, assumes a strict format for the text. Therefore, the input text is first preprocessed to meet the requirements of the analysis system.

The initial steps for a system, later to be developed and termed as SALAMA, were taken in 1985. The first module was the morphological analyzer implemented using finite state methods in the two-level framework (Koskenniemi 1983, Hurskainen 1992). Later on a disambiguation module, based on the Constraint Grammar (CG) formalism (Karlsson 1995; Tapanainen 1996, 1999; Hurskainen 1996, 2004a), was added. In addition to handling disambiguation and syntactic mapping with CG, also multi-word expressions (MWE) were implemented using it (Hurskainen 2006).

SALAMA contains several additional modules including the module for transferring Swahili lexical items to English, a module for controlling word order in target language (English), and a module for converting lexical words to surface forms.

In this paper, the emphasis is in describing how the module for morphological analysis can be implemented using the two-phase method. The motivation for an alternative solution was that, although finite state methods have many advantages, for example simplicity of description and excellent processing speed, so far there has been no open source compiler available. The two-phase method can be implemented without using proprietary tools and environments. Because an open source CG parser already exists[1], and other modules of SALAMA can be programmed using non-proprietary methods, the use of the two-phase method makes SALAMA independent from commercial tools. The system was implemented to Swahili, but it is a general approach applicable to other languages as well.

2 Features of the two-phase method

As the name of the method indicates, language analysis takes place in two phases in the two-phase method. Below is a description of both phases and how they were implemented.

[1] http://beta.visl.sdu.dk/cg3.html

The aim in morphological analysis is to get a detailed morphological description of each word-form. In (1) is an example of the verb-form *anayekisoma* and how it should be analyzed.

(1)
```
anayekisoma
       "soma" V 1-SG3-SP VFIN { he } PR:na 1-SG3-REL { who } 7-SG-OBJ OBJ
{ it } z [soma] { read } SVO
```

The verb-form is composed of the following morphemes:
a = subject prefix 3rd person singular of noun glass 1 (1-SG3-SP)
na = tense marker indicating present tense (PR:na)
ye = relative prefix 3rd person singular of noun class 1 (1-SG3-REL)
ki = object prefix singular noun class 7 (7-SG-OBJ)
som = verb root
a = verb-final vowel

In addition, we are informed that the verb stem is soma ("soma"), the basic stem is also soma [soma], its POS category is verb (V), it is a finite verb (VFIN), and it is a transitive verb (SVO). Glosses in English are also included within curly braces.

In the finite state implementation, all this can be described in a single phase, because a detailed description can be included for each lexical entry. Also in the two-phase method this would be possible, but hardly feasible. It is more convenient to carry out the description in two phases, whereby in the first phase we carry out a meta-level analysis, and on the basis of this analysis we expand the analysis into final form in the second phase.

2.1 Phase one

In (2) is the analysis result of the verb-form *anayekisoma*, using the first phase of the analysis system.

(2)
```
anayekisoma
       "soma" [soma] V aSP naTAM yeREL kiOBJ z[2] aVE REDUP
```

We see that some tags have two sections, the left part in lower case and the right part in upper case. In the left part, the affix morphemes of the verb have been copied, and the corresponding right part indicates the grammatical category of the morpheme. For example, naTAM indicates that the morpheme *na* is a tense/aspect/mood marker, but it does not tell precisely the correct category, that is, the present tense in this case.

---

[2] The character ‘z’ is a temporary anchor for locating the gloss of the verb. It also serves as a ‘marking post’ when surface forms of verbs are produced. When a rule applies, the ‘z’ is marked as ‘:z’, so that another rule would not try to modify again the gloss.

Why cannot we tell directly that it is present tense? This cannot be done, or more precisely, this is not feasible, because in the first phase we only identify morphemes and give to each morpheme a category label. The limitation derives from the decision to describe word-forms using regular expressions. In the implementation, each morpheme slot is identified and marked accordingly, for example SP, TAM, REL, OBJ. VE, and REDUP. These are grammatical tags, but as such insufficient for describing the word-form in sufficient detail. Each of the morpheme slots can have several different realizations, some of which are ambiguous, as we shall see later. Therefore, here we can only identify the morpheme slots (written in upper case) and show what is in each of the slots (written in lower case).

We should also note that the tag REDUP is without content. This is a tag indicating that the verb stem can be reduplicated. Also, when we leave out the object prefix, as in (3), we get the empty tag OBJ.

(3)
```
anayesoma
       "soma" [soma] V aSP naTAM yeREL OBJ z aVE REDUP
```

By default any tag without prefix will be removed (4).

(4)
```
anayesoma
       "soma" [soma] V aSP naTAM yeREL z aVE
```

When the word-form has the object prefix and the verb stem is reduplicated, we get all slots filled and analyzed (5).

(5)
```
anayekisomasoma
       "soma" [soma] V aSP naTAM yeREL kiOBJ z aVE somaREDUP
```

2.2 Phase two

In the second phase we expand the meta-level description, so that all necessary information is overtly described. Recall that in phase one we got the kind of description, where each morphological item was given a grammatical category label. In the second phase we specify the full content of each morpheme. The verb-form in (2) is fully analyzed in (6).

(6)
```
anayekisoma
       "soma" [soma] V 1-SG3-SP VFIN PR:na 1-SG-REL 7-SG-OBJ z
```

Each morpheme is now fully analyzed. The added information contains the noun class marking, the person, and the singular/plural distinction. We also see that the tag VFIN has been inserted, indicating that it is a finite verb-form.

The example we have been dealing with so far is quite simple, because it does not contain ambiguity. Consider the verb-form in (7), which we first pass through all the phases of analysis, including disambiguation.

(7)
```
unaouleta
       "leta" V [leta] SVO 11-SG-SP VFIN { it } PR:na 11-SG-REL { which }
11-SG-OBJ { it } z { bring }
```

The analysis result of phase one is in (8). It has only one reading without ambiguity

(8)
```
unaouleta
       "leta" [leta] V uSP naTAM oREL uOBJ z aVE REDUP
```

When we pass the analysis through phase two, we get several readings as in (9).

(9)
```
unaouleta
       "leta" [leta] V 1-SG2-SP VFIN PR:na 2-PL-REL 3-SG-OBJ z
       "leta" [leta] V 1-SG2-SP VFIN PR:na 2-PL-REL 11-SG-OBJ z
       "leta" [leta] V 1-SG2-SP VFIN PR:na 3-SG-REL 3-SG-OBJ z
       "leta" [leta] V 1-SG2-SP VFIN PR:na 3-SG-REL 11-SG-OBJ z
       "leta" [leta] V 1-SG2-SP VFIN PR:na 11-SG-REL 3-SG-OBJ z
       "leta" [leta] V 1-SG2-SP VFIN PR:na 11-SG-REL 11-SG-OBJ z
       "leta" [leta] V 3-SG-SP VFIN PR:na 2-PL-REL 3-SG-OBJ z
       "leta" [leta] V 3-SG-SP VFIN PR:na 2-PL-REL 11-SG-OBJ z
       "leta" [leta] V 3-SG-SP VFIN PR:na 3-SG-REL 3-SG-OBJ z
       "leta" [leta] V 3-SG-SP VFIN PR:na 3-SG-REL 11-SG-OBJ z
       "leta" [leta] V 3-SG-SP VFIN PR:na 11-SG-REL 3-SG-OBJ z
       "leta" [leta] V 3-SG-SP VFIN PR:na 11-SG-REL 11-SG-OBJ z
       "leta" [leta] V 11-SG-SP VFIN PR:na 2-PL-REL 3-SG-OBJ z
       "leta" [leta] V 11-SG-SP VFIN PR:na 2-PL-REL 11-SG-OBJ z
       "leta" [leta] V 11-SG-SP VFIN PR:na 3-SG-REL 3-SG-OBJ z
       "leta" [leta] V 11-SG-SP VFIN PR:na 3-SG-REL 11-SG-OBJ z
       "leta" [leta] V 11-SG-SP VFIN PR:na 11-SG-REL 3-SG-OBJ z
       "leta" [leta] V 11-SG-SP VFIN PR:na 11-SG-REL 11-SG-OBJ z
```

The reason for the ambiguity in (9) is that the subject prefix (SP) and relative prefix (REL) have three interpretations each, and the object prefix (OBJ) two interpretations. As a result there are eighteen grammatically correct interpretations for the word-form. This example shows the importance of the second phase in analysis, because without detailed morphological description the further phases in language processing will inevitably fail.

3 Introducing semantic information

The morphological analysis described above is based strictly on the morphological information available in the surface form of the word. No 'hidden' non-morphemic information is yet available. For example, there is no information whether the verb is transitive or not.

In the next step we introduce more lexical and semantic information to the analysis system. We get this information from the morphological lexicon, which also includes glosses in English for the morphemes. The example in (9) is enriched with glosses of prefixes and the word stem in (10).

(10)

```
unaouleta
       "leta" V [leta] SVO 1/2-SG2-SP VFIN { you } PR:na 1/2-PL-REL { who
} 3/4-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 1/2-SG2-SP VFIN { you } PR:na 1/2-PL-REL { who
} 11-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 1/2-SG2-SP VFIN { you } PR:na 3/4-SG-REL {
which } 3/4-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 1/2-SG2-SP VFIN { you } PR:na 3/4-SG-REL {
which } 11-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 1/2-SG2-SP VFIN { you } PR:na 11-SG-REL {
which } 3/4-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 1/2-SG2-SP VFIN { you } PR:na 11-SG-REL {
which } 11-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 3/4-SG-SP VFIN { it } PR:na 1/2-PL-REL { who }
3/4-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 3/4-SG-SP VFIN { it } PR:na 1/2-PL-REL { who }
11-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 3/4-SG-SP VFIN { it } PR:na 3/4-SG-REL { which
} 3/4-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 3/4-SG-SP VFIN { it } PR:na 3/4-SG-REL { which
} 11-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 3/4-SG-SP VFIN { it } PR:na 11-SG-REL { which
} 3/4-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 3/4-SG-SP VFIN { it } PR:na 11-SG-REL { which
} 11-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 11-SG-SP VFIN { it } PR:na 1/2-PL-REL { who }
3/4-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 11-SG-SP VFIN { it } PR:na 1/2-PL-REL { who }
11-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 11-SG-SP VFIN { it } PR:na 3/4-SG-REL { which
} 3/4-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 11-SG-SP VFIN { it } PR:na 3/4-SG-REL { which
} 11-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 11-SG-SP VFIN { it } PR:na 11-SG-REL { which }
3/4-SG-OBJ { it } z { bring }
       "leta" V [leta] SVO 11-SG-SP VFIN { it } PR:na 11-SG-REL { which }
11-SG-OBJ { it } z { bring }
```

Here we have a fairly complete description of the word-form *unaouleta*, and disambiguation rules can be written using this information. Of course, rule-based disambiguation relies heavily on the context, and disambiguation does not succeed without access to it. Therefore, we expand the example to make it a short subordinate sentence, *Umoja unaouleta uhuru*, (11).

(11)

```
*umoja
       "umoja" N 11-SG { unity }
unaouleta
```

```
       "leta" V [leta] SVO 11-SG-SP VFIN { it } PR:na 11-SG-REL { which }
11-SG-OBJ { it } z { bring }
uhuru
       "uhuru" N 11-SG { freedom }
,
       "," PUNC { , }
```

The word-form *unaouleta* is disambiguated in (11), but it still does not have syntactic tags. Syntactic mapping is done using the CG formalism. An example is in (12).

(12)
```
*umoja
       "umoja" N 11-SG { unity } @SUBJ
unaouleta
       "leta" V [leta] SVO 11-SG-SP VFIN { it } PR:na 11-SG-REL { which }
11-SG-OBJ { it } z { bring } @FMAINVtr+OBJ>
uhuru
       "uhuru" N 11-SG { freedom } @OBJ
,
       "," PUNC { , }
```

Syntactic tags have @ as a prefix. The interpretation of tags for the subject (@SUBJ) and object (@OBJ) is obvious. The tag for the verb is more cumbersome. It (@FMAINVtr+OBJ>) says that it is a finite main verb, inherently transitive, and has on the right an overt object in addition to the object pronoun prefixed to the verb. If the overt object is missing the verb tag would be as in (13).

(13)
```
*umoja
       "umoja" N 11-SG { unity } @SUBJ
unaouleta
       "leta" V [leta] SVO 11-SG-SP VFIN { it } PR:na 11-SG-REL { which }
11-SG-OBJ { it } z { bring } @FMAINVtr-OBJ>
,
       "," PUNC { , }
```

The verb tag (@FMAINVtr-OBJ>) tells that the verb is transitive, but the overt object is missing.

4 Describing multi-word expressions

In machine translation (MT), multi-word expressions (MWE) are difficult to handle for a number of reasons. Because a cluster of words may in one environment be a MWE and in another environment a normal sequence of words, in defining MWEs we need a sentence-wide context. The isolation of MWEs can be done using regular expressions. However, a convenient environment for doing this is a CG parser. In the current implementation this method was used.

The isolation of MWEs is done in phases. In (14) there is an example of a simple idiom, where a verb with an object constitutes a MWE. When each word is analyzed individually, the result is as in (14)

(14)
```
*alipiga
       "piga" V [piga] SVO ACT 1/2-SG3-SP VFIN { he/she } PAST z { hit }
picha
       "picha" N ENG 9/10-SG { picture }
```

First we mark the members of the MWE and attach the lexical gloss on the last member of the MWE (15).

(15)
```
*alipiga
       "piga" V [piga] SVO ACT 1/2-SG3-SP VFIN { he/she } PAST z { hit }
IDIOM-V>[3]
picha
       "picha" <IDIOM { photograph }
```

In the second phase we remove the original lexical gloss from the verb. The new gloss stands on the last member of the construction (16).

(16)
```
*alipiga
       "piga" V [piga] SVO ACT 1/2-SG3-SP VFIN { he/she } PAST z IDIOM-V>
@FMAINVtr-OBJ>
picha
       "picha" <IDIOM { photograph }
```

Be cause the source language is not needed any more, the token and the lemma are removed, and the whole construction is surrounded with parentheses (17)

(17)
```
( V SVO ACT 1/2-SG3-SP VFIN { he/she } PAST z IDIOM-V> @FMAINVtr-OBJ>
<IDIOM { photograph } )
```

A little bit more complicated MWE is in (18), which has three members.

(18)
```
*umoja
       "umoja" N 11-SG { unity }
wa
       "wa" GEN-CON 11-SG x { of }
*mataifa
       "taifa" N AR 5/6-PL { nation }
```

[3] The tag IDIOM-V> means that this is a member in an idiomatic expression, and that it is a verb, and that it is attached to the next token on the right. The tag <IDIOM tells that this is a member in an idiomatic expression, and that it is attached to the next token on the left.

This is quite a common genitive construction, which cannot be translated as a corresponding structure in the target language. Therefore, we mark the members of the MWE and attach the new gloss on the last member of the MWE (19).

(19)
```
*umoja
       "umoja" N 11-SG { unity } MW-N>>[4]
wa
       "wa" MW<>
*mataifa
       "taifa" <<MW { *united *nations }
```

Finally we remove the now redundant elements and enclose the MWE within parentheses (20).

(20)
```
( N 11-SG MW-N>>  <<MW { *united *nations } )[5]
```

5 Adding and deleting words in target language

In addition to MWEs, the source and target language may have also other types of discrepancies in mapping words from the source language to the target language. Verb constructions are typical examples of this. While Swahili marks the subject in the verb also in cases when there is an overt subject, English does not. When the overt subject is missing in Swahili, the SP takes the role of a subject pronoun. Therefore, in translating the construction into English, the subject prefix (SP) is in some cases translated and in other cases deleted.

The same applies also to the object prefix (OP). The presence of the OP in Swahili is not mandatory if the overt object follows. In fact, double marking is found mainly in referring to animates, while non-animate overt objects cause often the OP in verb to be deleted. In the absence of such a strict rule we have to be prepared to handle also this problem in translation.

The rule is: If the sentence contains an overt subject or object, remove the gloss of the prefix. In (21) is an example of a sentence with redundant SP and OP glosses.

---

4 The tag MW-N>> means that this is a multi-word expression in the category of nouns, and that it is attached to the second token on the right. The tag MW<> means that this is a member in a multi-word expression, and that it is attached to the first token on the left and the first token on the right. The tag <<MW means that this is a member in a multi-word expression, and that the expression has two other members immediately on the left.

5 Note that the first and last member of the MWE have been retained, because they contain linguistic information needed in later processing. The middle member is redundant and has been removed.

(21)
```
*mtoto
       "mtoto" N 1/2-SG { child } @SUBJ
anakisoma
       "soma" V [soma] SVO 1/2-SG3-SP VFIN { he/she } PR:na 7/8-SG-OBJ {
it } z { read } @FMAINVtr+OBJ>
kitabu
       "kitabu" N 7/8-SG { book } @OBJ
```

Now when the reading in (21) contains sufficient information for handling the glosses in the SP and OP, the presence of the gloss can be controlled. This can be done in two ways. In the first method we first produce glosses for the prefixes in the verb. Then, with a rule, we remove them depending on whether an overt subject or object is present. In (22) the reading in (21) is transformed into another format for easy manipulation with rules written using regular expressions.

(22)
```
( N 1/2-SG { the } { child } @SUBJ ) ( V SVO 1/2-SG3-SP VFIN PR:na 7/8-
SG-OBJ { it } z { read } @FMAINVtr+OBJ> ) ( N 7/8-SG { the } { book }
@OBJ )
```

In the second method we introduce two types of readings for the verb-form, one with glosses and another without glosses. An example of this implementation is in (23).

(23)
```
*mtoto
       "mtoto" N 1/2-SG { the } { child } CAP @SUBJ
anakisoma
       "soma" V 1/2-SG3-SP VFIN { he } PR:na 7/8-SG-OBJ OBJ { it } z
[soma] { read } SVO @FMAINVtr+OBJ>
       "soma" V 1/2-SG3-SP VFIN { he } PR:na 7/8-SG-OBJ OBJ NO-OBJ-GLOSS
z [soma] { read } SVO @FMAINVtr+OBJ>
       "soma" V 1/2-SG3-SP VFIN NO-SP-GLOSS PR:na 7/8-SG-OBJ OBJ { it } z
[soma] { read } SVO @FMAINVtr+OBJ>
       "soma" V 1/2-SG3-SP VFIN NO-SP-GLOSS PR:na 7/8-SG-OBJ OBJ NO-OBJ-
GLOSS z [soma] { read } SVO @FMAINVtr+OBJ>
kitabu
       "kitabu" N 7/8-SG { the } { book } @OBJ
```

We see in (23) that in fact there are four different readings, that is, one reading with glosses for the SP and OP, one reading with the SP gloss only, one reading with the OP gloss only, and one reading with no SP or OP glosses.
Then we select the appropriate reading (24).

(24)
```
*mtoto
       "mtoto" N 1/2-SG { the } { child } CAP @SUBJ
anakisoma
       "soma" V 1/2-SG3-SP VFIN NO-SP-GLOSS PR:na 7/8-SG-OBJ OBJ NO-OBJ-
GLOSS z [soma] { read } SVO @FMAINVtr+OBJ>
kitabu
       "kitabu" N 7/8-SG { the } { book } @OBJ
```

The advantage in using the second method above is that no string manipulation is needed, because all alternatives are present. The rule formalism of CG can be used for selecting the correct reading in each context.

6 Controlling word order

The order of words in Swahili and English is different. The main difference is in the order of dependent members of a noun phrase. In English the noun follows the dependent members, while in Swahili the noun precedes other members of the phrase. An example is in (25).

(25)
```
*watoto
       "mtoto" N 1/2-PL { the } { child } CAP
wangu
       "angu" PRON POSS 1/2-PL SG1 { my }
wazuri
       "zuri" ADJ A-INFL 1/2-PL { good }
hawa
       "hawa" PRON DEM :hV 1/2-PL { these }
wamesoma
       "soma" V 1/2-PL3-SP VFIN NO-SP-GLOSS PERF:me z [soma] { read } SVO
```

We first put the noun phrase on one line, each member within brackets (26).

(26)
```
( N 1/2-PL { the } { child } CAP ) ( PRON POSS 1/2-PL SG1 { my } ) ( ADJ
A-INFL 1/2-PL { good } ) ( PRON DEM :hV 1/2-PL { these } ) ( V 1/2-PL3-SP
VFIN NO-SP-GLOSS PERF:me z { read } SVO )
```

Then we re-order the constituents to meet the word order of English (27).

(27)
```
( PRON DEM :hV 1/2-PL { these } ) ( PRON POSS 1/2-PL SG1 { my } ) ( ADJ
A-INFL 1/2-PL { good } ) ( N 1/2-PL { child } CAP ) ( V 1/2-PL3-SP VFIN
NO-SP-GLOSS PERF:me z { read } SVO )
```

7 Converting lexical words to surface forms of the target language

In the next phase we translate the glosses into surface forms using the linguistic information inherited from the source language. Because English has an impoverished morphology, for some POS categories this is fairly easy. By default, nouns are in singular, and when plural is needed, a rule produces a plural form for the noun (28).

(28)
```
( PRON DEM :hV 1/2-PL { these } )
( PRON POSS 1/2-PL SG1 { my } )
( ADJ A-INFL 1/2-PL { good } )
```

```
( N 1/2-PL { children } CAP )
( V 1/2-PL3-SP VFIN NO-SP-GLOSS PERF:me :z { have } { read } SVO )
```

Then we leave only glosses and remove everything else and put the result on one line (29).

(29)
these my good children have read

Verb forms are particularly complex to implement, because there is no one-to-one correspondence between the verb structure of the source language and target language. Consider the verb-form in (30), where the English equivalents are still in lexical form, although the order of constituents has already been changed.

(30)
```
( V 1/2-SG3-SP VFIN { he/she } COND-NEG:singali z { implement } PREFR SVO
3/4-PL-OBJ OBJ { them } )
```

In the basis of the linguistic information we then convert the constituents into surface form. This is shown in (31).

(31)
```
( V 1/2-SG3-SP VFIN { he/she } COND-NEG:singali :z { would not have } {
implemented } PREFR SVO 3/4-PL-OBJ OBJ { them } )
```

After some pruning, the translation of the verb-form is in (32).

(32)
he/she would not have implemented them

8 Conclusion

The two-phase method for carrying out morphological analysis is a viable alternative for finite state methods. The first phase in the system was implemented using substitution rules defined by regular expressions. The second phase was implemented using Beta re-writing language, currently available as an open source implementation.

The later phases in processing, starting from disambiguation, are identical with the finite state implementation.

The two-phase method makes it possible to describe reduplication, which is cumbersome in using finite state methods. On the other hand, finite state implementation is several times faster than the two-phase method. The difference in processing speed is important in some applications, but in other applications it is hardly relevant.

References

Hurskainen A. 1992. "A Two-Level Computer Formalism for the Analysis of Bantu Morphology. An Application to Swahili". *Nordic Journal of African Studies* 1(1): 87-122. http://www.njas.helsinki.fi

Hurskainen A. 1996. "Disambiguation of morphological analysis in Bantu languages". *Proceedings of COLING-96*, pp. 568-573.

Hurskainen A. 2004a. "Optimizing Disambiguation in Swahili". In *Proceedings of COLING-04, The 20th International Conference on Computational Linguistics*, Geneva 23-27.8. 2004. Pp. 254-260.

Hurskainen, A. 2004b. "Swahili Language Manager: A Storehouse for Developing Multiple Computational Applications". *Nordic Journal of African Studies* 13(3): 363–397. http://www.njas.helsinki.fi

Hurskainen A. 2006. "Constraint Grammar in unconventional use: Handling complex Swahili idioms and proverbs". In A Man of Measure. Festschrift in Honour of Fred Karlsson on his 60th Birthday. Michael Suominen, Antti Arppe, Anu Airola, Orvokki Heinämäki, Matti Miestamo, Urho Määttä, Jussi Niemi, Kari K. Pitkänen and Kaius Sinnemäki (Editors). A Special Supplement to SKY Journal of linguistics. Turku: The Linguistic Association of Finland. Pp. 397- 406.

Karlsson, F. 1995. "Designing a parser for unrestricted text". Karlsson, F. et al (Eds.), *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*: 1-40. Berlin: Mouton de Gryuter.

Koskenniemi, K. 1983. *Two-level morphology: a general computational model for word-form recognition and production*, Publications No. 11. University of Helsinki: Department of General Linguistics.

Tapanainen, P. 1996. *The Constraint Grammar Parser CG-2.* Publications No. 27. Department of General Linguistics, University of Helsinki.

Tapanainen, P. 1999. *Parsing in two frameworks: finite-state and functional dependency grammar*. Ph.D. thesis, Department of General Linguistics, University of Helsinki.