

<https://helda.helsinki.fi>

Compound verbs in English to Swahili machine translation

Hurskainen, Arvi

University of Helsinki, Institute for Asian and African Studies
2019

Hurskainen , A 2019 ' Compound verbs in English to Swahili machine translation ' Technical reports on language technology , no. 43 , University of Helsinki, Institute for Asian and African Studies , Helsinki . <

<http://www.njas.helsinki.fi/salama/compound-verbs-in-en-to-swa-mt.pdf> >

<http://hdl.handle.net/10138/310871>

cc_by_nc
publishedVersion

Downloaded from Helda, University of Helsinki institutional repository.

This is an electronic reprint of the original article.

This reprint may differ from the original in pagination and typographic detail.

Please cite the original version.

Compound verbs in English to Swahili machine translation

Arvi Hurskainen
Department of World Cultures, Box 59
FIN-00014 University of Helsinki, Finland
arvi.hurskainen@helsinki.fi

Abstract

In this report we discuss such compound verbs, which require two inflected verbs in Swahili. In English, the constructions contain normally an inflecting auxiliary verb and the main verb in gerund. In negative forms, we find two alternative implementations.

Key Words: *compound verbs, machine translation*

1 Introduction

There are several types of compound verbs in English, and often the best way of translating them is to treat them as multiword expressions (MWE). However, there are also such compound verb types, which apply to a big number of verbs, and treating them as MWE is not feasible. The translation can be constructed using the combination of disambiguation and inflection tags.

The types of compound verbs that we discuss here are such as, *The worker is working*, *The worker was working*, *The worker has been working*, *The worker had been working* etc. The main verb is in gerund, and the auxiliary verb inflects. In Swahili, both verbs inflect, and the main verb may have two alternative TAM-markers.

In negative forms there is still one more alternative. The negation can be placed either to the auxiliary verb or to the main verb.

Let us take a look at the expressions *The worker is working* and *The worker works*. Both are in present tense, but the meaning is slightly different. The former expression means that the worker is in the action of work at the time of saying. The latter expression states that the job of the worker is to work, but that the expression does not imply that the worker works at the time of saying. English uses the gerund form of the main verb to express the difference.

We will study the translation using seven sentences, each in affirmative and negative form (1).

- (1)
 - a. The worker is working.
 - b. The worker was working.
 - c. The worker has been working.
 - d. The worker had been working.
 - e. The worker will be working.
 - f. The worker would be working.
 - g. The worker should be working.

- h. The worker is not working.
- i. The worker was not working.
- j. The worker has not been working.
- k. The worker had not been working.
- l. The worker will not be working.
- m. The worker would not be working.
- n. The worker should not be working.

Let us see the difference in translation between normal present tense and the combination of auxiliary verb and gerund form (2).

(2)

The worker works.

Mfanyakazi anafanya kazi.

The worker is working.

Mfanyakazi anakuwa akifanya kazi.

In the basic present tense, the TAM marker is *-na-*, and it is attached as a prefix to the verb. In the gerund form, the present tense marker *-na-* is attached to the auxiliary verb, and the main verb gets the TAM marker *-ki-*. This marker is also the conditional marker.

It is also possible that the marker *-ki-* is substituted by the present tense marker *-na-*. The translation then would be *Mfanyakazi anakuwa anafanya kazi*. It is not known whether there is any semantic difference between these two expressions.

2 Phases of translation of affirmative forms

Next we take a look at the translation phases. We take first the seven affirmative sentences and see how they should be treated in each phase. After analysis, disambiguation, and semantic mapping, the result is as in (3).

(3)

a.

```
"<The>"
  "the" %DN> CAPINIT DET
"<worker>"
  "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL fanyikazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL tendakazi } %SUBJ N SG NOM DEF
"<is>"
  "be" { INFMARK+wA } MONOSLB %+FAUXV V PRES SG3
  "be" { wA } MONOSLB %+FAUXV V PRES SG3
  "be" { ni } MONOSLB %+FAUXV V PRES SG3
  "be" { si } MONOSLB %+FAUXV V PRES SG3
  "be" { AUX } MONOSLB %+FAUXV V PRES SG3
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  ". "
```

b.

```
"<The>"
  "the" %DN> CAPINIT DET
"<worker>"
  "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL fanyikazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL tendakazi } %SUBJ N SG NOM DEF
"<was>"
  "be" { INFMARK+wA } MONOSLB %+FAUXV V PAST
  "be" { wA } MONOSLB %+FAUXV V PAST
  "be" { ni } MONOSLB %+FAUXV V PAST
  "be" { si } MONOSLB %+FAUXV V PAST
  "be" { AUX } MONOSLB %+FAUXV V PAST
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
```

c.

```
"<The>"
  "the" %DN> CAPINIT DET
"<worker>"
  "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL fanyikazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL tendakazi } %SUBJ N SG NOM DEF
"<has>"
  "have" { INFMARK+wA na } %+FAUXV V PRES SG3
  "have" { AUX } %+FAUXV V PRES SG3
  "have" { -na } %+FAUXV V PRES SG3
  "have" { INFMARK+wA } %+FAUXV V PRES SG3
"<been>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V EN
  "be" { wA } MONOSLB %-FAUXV V EN
  "be" { ni } MONOSLB %-FAUXV V EN
  "be" { si } MONOSLB %-FAUXV V EN
  "be" { AUX } MONOSLB %-FAUXV V EN
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
```

d.

```
"<The>"
  "the" %DN> CAPINIT DET
"<worker>"
  "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL fanyikazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL tendakazi } %SUBJ N SG NOM DEF
"<had>"
  "have" { INFMARK+wA na } %+FAUXV V PAST
  "have" { AUX } %+FAUXV V PAST
  "have" { -na } %+FAUXV V PAST
  "have" { INFMARK+wA } %+FAUXV V PAST
```

```
"<been>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V EN
  "be" { wA } MONOSLB %-FAUXV V EN
  "be" { ni } MONOSLB %-FAUXV V EN
  "be" { si } MONOSLB %-FAUXV V EN
  "be" { AUX } MONOSLB %-FAUXV V EN
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
e.
"<The>"
  "the" %DN> CAPINIT DET
"<worker>"
  "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL fanyikazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL tendakazi } %SUBJ N SG NOM DEF
"<will>"
  "will" { FUT } %+FAUXV V AUXMOD
  "will" { pendA } %+FAUXV V AUXMOD
"<be>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V INF
  "be" { wA } MONOSLB %-FAUXV V INF
  "be" { ni } MONOSLB %-FAUXV V INF
  "be" { si } MONOSLB %-FAUXV V INF
  "be" { AUX } MONOSLB %-FAUXV V INF
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
f.
"<The>"
  "the" %DN> CAPINIT DET
"<worker>"
  "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL fanyikazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL tendakazi } %SUBJ N SG NOM DEF
"<would>"
  "would" { AUX } %+FAUXV V AUXMOD
"<be>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V INF
  "be" { wA } MONOSLB %-FAUXV V INF
  "be" { ni } MONOSLB %-FAUXV V INF
  "be" { si } MONOSLB %-FAUXV V INF
  "be" { AUX } MONOSLB %-FAUXV V INF
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
g.
"<The>"
```

```
"the" %DN> CAPINIT DET
"<worker>"
  "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL fanyikazi } %SUBJ N SG NOM DEF
  "worker" { 1SG 2PL tendakazi } %SUBJ N SG NOM DEF
"<should>"
  "should" { takiwA } %+FAUXV V AUXMOD
"<be>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V INF
  "be" { wA } MONOSLB %-FAUXV V INF
  "be" { ni } MONOSLB %-FAUXV V INF
  "be" { si } MONOSLB %-FAUXV V INF
  "be" { AUX } MONOSLB %-FAUXV V INF
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
```

We see that the auxiliary verb *be* has several interpretations. In our examples we should select the one, which has the infinitive marker. The result is in (4).

(4)

```
a.
"<The>"
  "the" %DN> CAPINIT DET
"<worker>"
  "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
"<is>"
  "be" { INFMARK+wA } MONOSLB %+FAUXV V PRES SG3
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."

b.
"<The>"
  "the" %DN> CAPINIT DET
"<worker>"
  "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
"<was>"
  "be" { INFMARK+wA } MONOSLB %+FAUXV V PAST
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."

c.
"<The>"
  "the" %DN> CAPINIT DET
"<worker>"
  "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
"<has>"
  "have" { AUX } %+FAUXV V PRES SG3
```

"<been>"
 "be" { INFMARK+wA } MONOSLB %-FAUXV V EN
"<working>"
 "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
 "."
d.
"<The>"
 "the" %DN> CAPINIT DET
"<worker>"
 "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
"<had>"
 "have" { AUX } %+FAUXV V PAST
"<been>"
 "be" { INFMARK+wA } MONOSLB %-FAUXV V EN
"<working>"
 "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
 "."
e.
"<The>"
 "the" %DN> CAPINIT DET
"<worker>"
 "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
"<will>"
 "will" { FUT } %+FAUXV V AUXMOD
"<be>"
 "be" { INFMARK+wA } MONOSLB %-FAUXV V INF
"<working>"
 "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
 "."
f.
"<The>"
 "the" %DN> CAPINIT DET
"<worker>"
 "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
"<would>"
 "would" { AUX } %+FAUXV V AUXMOD
"<be>"
 "be" { INFMARK+wA } MONOSLB %-FAUXV V INF
"<working>"
 "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
 "."
g.
"<The>"
 "the" %DN> CAPINIT DET
"<worker>"
 "worker" { 1SG 2PL fanyakazi } %SUBJ N SG NOM DEF
"<should>"

```
"should" { takiwA } %+FAUXV V AUXMOD
"<be>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V INF
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  ". "
```

Inflection tags are added tp readings, so that surface forms can be generated (5).

(5)

a.

```
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<is>"
  "be" { INFMARK+wA } MONOSLB %+FAUXV V PRES SG3 TAM-na SP-1
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
  ". "
```

b.

```
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<was>"
  "be" { INFMARK+wA } MONOSLB %+FAUXV V PAST TAM-li SP-1
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
  ". "
```

c.

```
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<has>"
  "have" { AUX } %+FAUXV ACR V PRES SG3
"<been>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V EN TAM-me SP-1
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
  ". "
```

d.

```
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<had>"
  "have" { INFMARK+wA } %+FAUXV ACR V PAST TAM-li SP-1
"<been>"
  "be" { wA } MONOSLB %-FAUXV V EN SBJN SP-1
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
  ". "
```


e.
"<worker>"
 "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<will>"
 "will" { FUT } %+FAUXV ACR V AUXMOD
"<be>"
 "be" { INFMARK+wA } MONOSLB %-FAUXV V INF TAM-ta SP-1
"<working>"
 "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
 "."

f.
"<worker>"
 "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<would>"
 "would" { AUX } %+FAUXV ACR V AUXMOD
"<be>"
 "be" { INFMARK+wA } MONOSLB %-FAUXV V INF TAM-nge SP-1
"<working>"
 "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
 "."

g.
"<worker>"
 "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<should>"
 "should" { takiwA } %+FAUXV V AUXMOD TAM-nge SP-1
"<be>"
 "be" { wA } MONOSLB %-FAUXV V INF SBJN SP-1
"<working>"
 "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
 "."

We see above that the auxiliary verb and main verb has a tag for TAM and a tag for subject prefix. For TAM, the prefix is different for the auxiliary verb and main verb. For the subject prefix the tag is the same, that is, the tag of the noun class 1 single.

The examples (d) and (g) above are special cases, because they have three inflecting verbs. Note that the verb *be* in the constructions is in subjunctive. Yet all three verbs have the same subject prefix tag, because they refer to the same subject.

When we move the tags to the correct places, we get the result as in (6).

(6)
a.
"<worker>"
 "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<is>"
 "be" { SP-1+TAM-na+INFMARK+wA } MONOSLB %+FAUXV V PRES SG3
"<working>"
 "work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING

```
"<.>"  
  "."  
b.  
"<worker>"  
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF  
"<was>"  
  "be" { SP-1+TAM-li+INFMARK+wA } MONOSLB %+FAUXV V PAST  
"<working>"  
  "work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING  
"<.>"  
  "."  
c.  
"<worker>"  
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF  
"<has>"  
  "have" { AUX } %+FAUXV ACR V PRES SG3  
"<been>"  
  "be" { SP-1+TAM-me+INFMARK+wA } MONOSLB %-FAUXV V EN  
"<working>"  
  "work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING  
"<.>"  
  "."  
d.  
"<worker>"  
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF  
"<had>"  
  "have" { SP-1+TAM-li+INFMARK+wA } %+FAUXV ACR V PAST  
"<been>"  
  "be" { SP-1+w+e } MONOSLB %-FAUXV V EN  
"<working>"  
  "work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING  
"<.>"  
  "."  
e.  
"<worker>"  
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF  
"<will>"  
  "will" { FUT } %+FAUXV ACR V AUXMOD  
"<be>"  
  "be" { SP-1+TAM-ta+INFMARK+wA } MONOSLB %-FAUXV V INF  
"<working>"  
  "work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING  
"<.>"  
  "."  
f.  
"<worker>"  
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF  
"<would>"  
  "would" { AUX } %+FAUXV ACR V AUXMOD  
"<be>"  
  "be" { SP-1+TAM-nge+INFMARK+wA } MONOSLB %-FAUXV V INF
```

```
"<working>"
    "work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
g.
"<worker>"
    "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<should>"
    "should" { SP-1+TAM-nge+takiwA } %+FAUXV V AUXMOD
"<be>"
    "be" { SP-1+w+e } MONOSLB %-FAUXV V INF
"<working>"
    "work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
```

The tags are converted into surface form. The result is in (7).

```
(7)
a.
"<worker>"
    "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
"<is>"
    "be" { a+na+ku+wa } MONOSLB %+FAUXV V PRES SG3
"<working>"
    "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
b.
"<worker>"
    "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
"<was>"
    "be" { a+li+ku+wa } MONOSLB %+FAUXV V PAST
"<working>"
    "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
c.
"<worker>"
    "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
"<has>"
    "have" { AUX } %+FAUXV ACR V PRES SG3
"<been>"
    "be" { a+me+ku+wa } MONOSLB %-FAUXV V EN
"<working>"
    "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
d.
"<worker>"
    "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
```

```
"<had>"
    "have" { a+li+ku+wa } %+FAUXV ACR V PAST
"<been>"
    "be" { a+w+e } MONOSLB %-FAUXV V EN
"<working>"
    "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
e.
"<worker>"
    "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
"<will>"
    "will" { FUT } %+FAUXV ACR V AUXMOD
"<be>"
    "be" { a+ta+ku+wa } MONOSLB %-FAUXV V INF
"<working>"
    "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
f.
"<worker>"
    "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
"<would>"
    "would" { AUX } %+FAUXV ACR V AUXMOD
"<be>"
    "be" { a+nge+ku+wa } MONOSLB %-FAUXV V INF
"<working>"
    "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
g.
"<worker>"
    "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
"<should>"
    "should" { a+nge+takiwa } %+FAUXV V AUXMOD
"<be>"
    "be" { a+w+e } MONOSLB %-FAUXV V INF
"<working>"
    "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
```

The final translation is in (8).

(8)

- a. *Mfanyakazi anakuwa akifanya kazi.*
- b. *Mfanyakazi alikuwa akifanya kazi.*
- c. *Mfanyakazi amekuwa akifanya kazi.*
- d. *Mfanyakazi alikuwa awe akifanya kazi.*
- e. *Mfanyakazi atakuwa akifanya kazi.*

- f. *Mfanyakazi angekuwa akifanya kazi.*
- g. *Mfanyakazi angetakiwa awe akifanya kazi.*

3 Phases of translation of negative forms

The negation of the examples above can be done in two ways. We can put the auxiliary verb into negative form and the main verb into *-ki-* form. Alternatively, we can put the auxiliary verb into affirmative form and the main verb into negative form. The affirmative *-ki-* form does not have a direct corresponding negative form. We can use the present tense negative form for that. Remember that in affirmative forms we can also use the *-na-* tense instead of *-ki-*.

The sentences that we consider here are in (9).

(9)

- a. The worker is not working.
- b. The worker was not working.
- c. The worker has not been working.
- d. The worker had not been working.
- e. The worker will not be working.
- f. The worker would not be working.
- g. The worker should not be working.

3.1 Auxiliary verb in negative form and main verb in *-ki-* form

After analysis, morphological disambiguation, syntactic mapping, and semantic disambiguation, the result is as in (10).

(10)

```
a.
"<worker>"
    "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<is>"
    "be" { wA } MONOSLB %+FAUXV V PRES SG3
"<not>"
    "not" { NOGLOSS } %ADVL ACR NEG-PART
"<working>"
    "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
b.
"<worker>"
    "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<was>"
    "be" { INFMARK+wA } MONOSLB %+FAUXV V PAST
"<not>"
    "not" { NOGLOSS } %ADVL ACR NEG-PART
"<working>"
```

```
"work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
c.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<has>"
  "have" { AUX } %+FAUXV ACR V PRES SG3
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<been>"
  "be" { wA } MONOSLB %-FAUXV V EN
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
d.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<had>"
  "have" { INFMARK+wA } %+FAUXV ACR V PAST
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<been>"
  "be" { wA } MONOSLB %-FAUXV V EN
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
e.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<will>"
  "will" { FUT } %+FAUXV ACR V AUXMOD
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V INF
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
f.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<would>"
  "would" { AUX } %+FAUXV ACR V AUXMOD
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V INF
```

```
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
g.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<should>"
  "should" { takiwA } %+FAUXV V AUXMOD
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { wA } MONOSLB %-FAUXV V INF
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
```

We note that in (c) the verb *have* has been glossed as AUX and in (d) as INFMARK+wA. Also, the verb *be* has sometimes been glossed as wA and in others as INFMARK+wA. This is done to facilitate correct translation.

When inflection tags are added, we get the result as in (11).

```
(11)
a.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<is>"
  "be" { wA } MONOSLB %+FAUXV V PRES SG3 TAM-0 SP-NEG-1
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
  "."
b.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<was>"
  "be" { INFMARK+wA } MONOSLB %+FAUXV V PAST TAM-ku SP-NEG-1
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
  "."
c.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<has>"
```

```
"have" { AUX } %+FAUXV ACR V PRES SG3
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<been>"
  "be" { wA } MONOSLB %-FAUXV V EN TAM-ja SP-NEG-1
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
  "."
d.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<had>"
  "have" { INFMARK+wA } %+FAUXV ACR V PAST TAM-ku SP-NEG-1
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<been>"
  "be" { wA } MONOSLB %-FAUXV V EN SBJN SP-1
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
  "."
e.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<will>"
  "will" { FUT } %+FAUXV ACR V AUXMOD
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V INF TAM-ta SP-NEG-1
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
  "."
f.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<would>"
  "would" { AUX } %+FAUXV ACR V AUXMOD
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V INF TAM-singe SP-1
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
  "."
g.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
```



```
"<should>"
  "should" { takiwA } %+FAUXV V AUXMOD TAM-0 SP-NEG-1
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { wA } MONOSLB %-FAUXV V INF SBJN SP-1
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-ki SP-1
"<.>"
  ". "
```

Note that in (f) above the subject prefix tag of both verbs is the affirmative tag. This is because the TAM marker TAM-singe already contains negation.

Then the tags are moved to correct places. Note that the tag SBJN for subjunctive is realised as the final *e* of the verb. Also, the present tense negative is realised as final *i* (12).

(12)

a.

```
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<is>"
  "be" { SP-NEG-1+TAM-0+w+i } MONOSLB %+FAUXV V PRES SG3
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<working>"
  "work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  ". "
```

b.

```
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<was>"
  "be" { SP-NEG-1+TAM-ku+INFMARK+wA } MONOSLB %+FAUXV V PAST
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<working>"
  "work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  ". "
```

c.

```
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<has>"
  "have" { AUX } %+FAUXV ACR V PRES SG3
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<been>"
  "be" { SP-NEG-1+TAM-ja+wA } MONOSLB %-FAUXV V EN
"<working>"
```

```
"work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
d.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<had>"
  "have" { SP-NEG-1+TAM-ku+INFMARK+wA } %+FAUXV ACR V PAST
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<been>"
  "be" { SP-1+w+e } MONOSLB %-FAUXV V EN
"<working>"
  "work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
e.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<will>"
  "will" { FUT } %+FAUXV ACR V AUXMOD
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { SP-NEG-1+TAM-ta+INFMARK+wA } MONOSLB %-FAUXV V INF
"<working>"
  "work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
f.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<would>"
  "would" { AUX } %+FAUXV ACR V AUXMOD
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { SP-1+TAM-singe+INFMARK+wA } MONOSLB %-FAUXV V INF
"<working>"
  "work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
g.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<should>"
  "should" { SP-NEG-1+TAM-0+takiw+i } %+FAUXV V AUXMOD
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { SP-1+w+e } MONOSLB %-FAUXV V INF
```

```
"<working>"  
    "work" { SP-1+TAM-ki+fanyA kazi } HUM-V %-FMAINV V ING  
"<.>"  
    "."
```

The inflection tags are converted into surface form (13).

(13)

a.

```
"<worker>"  
    "worker" { m+fanyakazi } %SUBJ N SG NOM DEF  
"<is>"  
    "be" { ha+w+i } MONOSLB %+FAUXV V PRES SG3  
"<not>"  
    "not" { NOGLOSS } %ADVL ACR NEG-PART  
"<working>"  
    "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING  
"<.>"  
    "."
```

b.

```
"<worker>"  
    "worker" { m+fanyakazi } %SUBJ N SG NOM DEF  
"<was>"  
    "be" { ha+ku+ku+wa } MONOSLB %+FAUXV V PAST  
"<not>"  
    "not" { NOGLOSS } %ADVL ACR NEG-PART  
"<working>"  
    "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING  
"<.>"  
    "."
```

c.

```
"<worker>"  
    "worker" { m+fanyakazi } %SUBJ N SG NOM DEF  
"<has>"  
    "have" { ha+AUX } %+FAUXV ACR V PRES SG3  
"<not>"  
    "not" { NOGLOSS } %ADVL ACR NEG-PART  
"<been>"  
    "be" { ha+ja+wa } MONOSLB %-FAUXV V EN  
"<working>"  
    "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING  
"<.>"  
    "."
```

d.

```
"<worker>"  
    "worker" { m+fanyakazi } %SUBJ N SG NOM DEF  
"<had>"  
    "have" { ha+ku+ku+wa } %+FAUXV ACR V PAST  
"<not>"  
    "not" { NOGLOSS } %ADVL ACR NEG-PART  
"<been>"
```

```
"be" { a+w+e } MONOSLB %-FAUXV V EN
"<working>"
  "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
e.
"<worker>"
  "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
"<will>"
  "will" { ha+FUT } %+FAUXV ACR V AUXMOD
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { ha+ta+ku+wa } MONOSLB %-FAUXV V INF
"<working>"
  "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
f.
"<worker>"
  "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
"<would>"
  "would" { ha+AUX } %+FAUXV ACR V AUXMOD
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { a+singe+ku+wa } MONOSLB %-FAUXV V INF
"<working>"
  "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
g.
"<worker>"
  "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
"<should>"
  "should" { ha+takiw+i } %+FAUXV V AUXMOD
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { a+w+e } MONOSLB %-FAUXV V INF
"<working>"
  "work" { a+ki+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
```

The final translation can now be produced. There is no need for reordering constituents (14).

(14)
a. Mfanyakazi hawi akifanya kazi.

- b. *Mfanyakazi hakukuwa akifanya kazi.*
- c. *Mfanyakazi hajawa akifanya kazi.*
- d. *Mfanyakazi hakukuwa awe akifanya kazi.*
- e. *Mfanyakazi hatakuwa akifanya kazi.*
- f. *Mfanyakazi asingekuwa akifanya kazi.*
- g. *Mfanyakazi hatakiwi awe akifanya kazi.*

3.2 Auxiliary verb in affirmative form and main verb in negative form

The negative examples above were implemented using the negative form of the auxiliary verb and the *-ki-* form of the main verb. Part of the examples can also be implemented so that the auxiliary verb is in affirmative form and the main verb in negative form. The main verb is in present tense, and the auxiliary verb inflects in various tenses. We will consider the examples in (15).

(15)

- a. The worker has not been working.
- b. The worker had not been working.
- c. The worker will not be working.
- d. The worker would not be working.

When we process the sentences to the point, where inflection tags are added, we get the result as in (16).

(16)

```
a.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<has>"
  "have" { AUX } %+FAUXV ACR V PRES SG3
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<been>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V EN TAM-me SP-1
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-0 SP-NEG-1
"<.>"
  "."
b.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<had>"
  "have" { AUX } %+FAUXV ACR V PAST
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<been>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V EN TAM-li SP-1
"<working>"
```

```
"work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-0 SP-NEG-1
"<.>"
  "."
c.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<will>"
  "will" { FUT } %+FAUXV ACR V AUXMOD
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V INF TAM-ta SP-1
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-0 SP-NEG-1
"<.>"
  "."
d.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<would>"
  "would" { AUX } %+FAUXV ACR V AUXMOD
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
  "be" { INFMARK+wA } MONOSLB %-FAUXV V INF TAM-nge SP-1
"<working>"
  "work" { fanyA kazi } HUM-V %-FMAINV V ING TAM-0 SP-NEG-1
"<.>"
  "."
```

When the inflection tags are moved as part of the verb, the result is as in (17).

(17)

```
a.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<has>"
  "have" { AUX } %+FAUXV ACR V PRES SG3
"<not>"
  "not" { NOGLOSS } %ADVL ACR NEG-PART
"<been>"
  "be" { SP1+TAM-me+INFMARK+wA } MONOSLB %-FAUXV V EN
"<working>"
  "work" { SP-NEG-1+TAM-0+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
  "."
b.
"<worker>"
  "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<had>"
  "have" { AUX } %+FAUXV ACR V PAST
```

```
"<not>"
    "not" { NOGLOSS } %ADVL ACR NEG-PART
"<been>"
    "be" { SP-1+TAM-li+INFMARK+wA } MONOSLB %-FAUXV V EN
"<working>"
    "work" { SP-NEG-1+TAM-0+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
c.
"<worker>"
    "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<will>"
    "will" { FUT } %+FAUXV ACR V AUXMOD
"<not>"
    "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
    "be" { SP-1+TAM-ta+INFMARK+wA } MONOSLB %-FAUXV V INF
"<working>"
    "work" { SP-NEG-1+TAM-0+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
d.
"<worker>"
    "worker" { 1SG fanyakazi } %SUBJ N SG NOM DEF
"<would>"
    "would" { AUX } %+FAUXV ACR V AUXMOD
"<not>"
    "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
    "be" { SP-1+TAM-nge+INFMARK+wA } MONOSLB %-FAUXV V INF
"<working>"
    "work" { SP-NEG-1+TAM-0+fanyA kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
```

When the tags are converted to surface form, the result is as in (18).

(18)

```
a.
"<worker>"
    "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
"<has>"
    "have" { AUX } %+FAUXV ACR V PRES SG3
"<not>"
    "not" { NOGLOSS } %ADVL ACR NEG-PART
"<been>"
    "be" { a+me+ku+wA } MONOSLB %-FAUXV V EN
"<working>"
    "work" { ha+fanyi kazi } HUM-V %-FMAINV V ING
"<.>"
    "."
```

b.
"<worker>"
 "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
"<had>"
 "have" { AUX } %+FAUXV ACR V PAST
"<not>"
 "not" { NOGLOSS } %ADVL ACR NEG-PART
"<been>"
 "be" { a+li+ku+wa } MONOSLB %-FAUXV V EN
"<working>"
 "work" { ha+fanyi kazi } HUM-V %-FMAINV V ING
"<.>"
 "."

c.
"<worker>"
 "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
"<will>"
 "will" { FUT } %+FAUXV ACR V AUXMOD
"<not>"
 "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
 "be" { a+ta+ku+wa } MONOSLB %-FAUXV V INF
"<working>"
 "work" { ha+fanyi kazi } HUM-V %-FMAINV V ING
"<.>"
 "."

d.
"<worker>"
 "worker" { m+fanyakazi } %SUBJ N SG NOM DEF
"<would>"
 "would" { AUX } %+FAUXV ACR V AUXMOD
"<not>"
 "not" { NOGLOSS } %ADVL ACR NEG-PART
"<be>"
 "be" { a+nge+ku+wa } MONOSLB %-FAUXV V INF
"<working>"
 "work" { ha+fanyi kazi } HUM-V %-FMAINV V ING
"<.>"
 "."

Note that the form of the main verb is the same regardless the form of the auxiliary verb. The final translation is in (19).

- (19)
- a. *Mfanyakazi amekuwa hafanyi kazi.*
 - b. *Mfanyakazi alikuwa hafanyi kazi.*
 - c. *Mfanyakazi atakuwa hafanyi kazi.*
 - d. *Mfanyakazi angekuwa hafanyi kazi.*

4 Conclusion

We have discussed the translation of complex compound verbs from English to Swahili. We have gone through various translation possibilities and found out that it is possible to implement any of the alternatives. However, in practice one should make a decision between alternative translations, especially if there are no clear criteria in text for making a decision.