Technical Reports on Language Technology
Report No 62, 20020
http://www.njas.helsinki.fi/salama

Can machine translation assist in Bible translation?[1]

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

Abstract

When an unwritten language comes to the process of expressing it in written form, it is most often related somehow to Bible translation. Parts of Bible are often the first written pieces of that language. This means that Bible translators in fact create the writing system of the language. All the initial phases in creating a written language must be done manually. Also first phases of Bible translation are normally manual work. When translation work proceeds to cover the whole Bible, the use of computers in translation becomes worth considering. This review describes the method how a rule-based language analysis system can be used for translating biblical texts via English to a third language. The source language is Swahili, and the target language is Luganda.

Key Words: *machine translation, Bible translation.*

1 Introduction

In an earlier technical report in 2009[2] I demonstrated how Swahili translation system can be made use of in translating Bible text into Luganda. The translation system translates Swahili text into English. By making use of the linguistic information inherited from the source language (Swahili), and the lexical information of the target language (English), it is possible to translate the text into a third language (in this case Luganda). The requirement is that we have a conversion lexicon from English to Luganda and that we know the language structure of Luganda. It was possible to make the test, because I had access to the computerised English-Luganda dictionary and knowledge of the language structure.

The translation demonstration was based strictly on a rule-based approach, where each phase in translation process can be traced. An entirely different approach is the statistical approach, and currently its developed version, neural approach. If we consider translation to a language, which does not have language resources, such as parallel text corpora, statistical and neural methods cannot be used. But what if we have a language, which does not have a dictionary, and not even a written grammar?

---

[1] The report is issued under licence CC BY-NC
[2] http://www.njas.helsinki.fi/salama/bible-translation.pdf

When the Bible translator has such a language at hand, the very first thing is to construct a grammar, or at least a sketch of grammar. Also, vocabulary accumulation is among the first tasks. The vocabularies might evetually develop into dictionaries. The translator must work with these basic language resources in any cae, because even manual translation work cannot be done without basic resources.

The motivation for writing this new report was that since the publication of the first report (2009) much new research and testing has been done. The Bible translation system is in fact only an application of a general machine translation approach, using English as interlingua[3]

If we consider using rule-based machine translation to assist in translation, the very basic resources are the same as in manual translation. One must know the verb structure, the word order rules, and in the case of Bantu languages, the noun class system.

2 The suitability of neural approaches to Bible translation

The current main trend in machine translation is to use the so-called neural methods. In these methods, there is in fact nothing neural. The term is used just because it sounds so mysterious. However, in one respect the term is correct. As we know very little about how the neural system in living beings works, we know equally little how the neural machine translation works. If statistical machine translation was sheer guessing on the basis of likelihoods, neural translation is a more fine-grained form of guessing. If this is the case, it is almost impossible to trace the reasons why translation fails. The difficulty of tracing translation mistakes is an additional reason, why neural methods are hardly optimal for Bible translation.

The more fundamental reason for their unsuitability is that, in order to be successful, neural translation needs large amounts of parallel texts. Tis condition is simply missing with translation tasks that we discuss here.

3 Current use of computers in Bible translation

The leading environment for handling biblical texts in translation is Paratext (paratext.org). It is a multiwindow environment, where the translator can check the translation under work with other translations of the same text. The system itself does not translate. It is just a convenient environment to keep important text varieties on the screen, where corrections to text can be made on the basis of information available in other translations. Even translation from scratch can be done in this environment. No wonder that it is popular among Bible translators.

Paratext, no doubt, speeds up translation and checking, but it has nothing to do with machine translation, except for translation memory, which can be classified as machine translation, and which in some instances can be used for speeding up the work. The

[3] http://www.njas.helsinki.fi/salama/machine-translation-through-interlingua.pdf
http://www.njas.helsinki.fi/salama/multi-channel-approach-to-global-mt.pdf
http://www.njas.helsinki.fi/salama/translation-via-interlingua.pdf
https://researchportal.helsinki.fi/en/publications/toward-global-machine-translation

Paratext environment also can help in translating biblical texts directly from a closely related language, which already has a translation[4].

There are also other tools for helping to bring source information available to the translation and editing environment, such as the alignment of morphemes between Hebrew, Greek, and the target language developed by Anssi Yli-jyrä[5].

It is not known to me that neural methods would have been used for Bible translation. On the other hand, neural methods and the earlier statistical methods are very much dependent of the availability of the human-translated biblical texts. Bible translations constitute the largest source of parallel texts, that are necessary for training the translation systems. Bible texts are also carefully edited, which increases their value. Training the system with poor quality translations is vasted time.

4 Producing a translation in new language as a computer process

We must note very first that it is not feasible to try to achieve ready translation using a computerised translation system. Human input is absolutely necessary. The process can be characterised as a sequence of two phases. The computer makes the translation, and when it encounters a translation problem, it marks such points clearly for human ispection later. The raw translation is then the version, which the human translator checks and corrects. The latter part of work can easily be done in such environments as Paratext, where comparative material can be made available on the same screen.

Below I will describe the phases of rule-based machine translation into a new language. I will use Swahili as source language and Luganda as target language. Luganda does have Bible translation, but I use this language just as an example.

4.1 Analysis of source text

In order to convert the text into a more abstract form, we analyze the text. In this phase each word is inspected, and if it has other morphemes in addition to the stem, those morphemes are identified and marked with specific linguistic tags. As a result, the text is represented in its original surface form, and also in its abstract form. To be precise, there is also a third level between the surface form and abstract form. It is the lexical form, which is void of morphophonological alternations, which appear mostly in morpheme boundaries in surface forms.

The abstract form of prefixes and suffixes is important, because in it such forms that appear as homonyms in surface form are marked uniquely in abstract form. This method has tremendous consequences on translation, and only the rule-based approach can make such distinctions. Statistical and neural methods see the surface strings and have no access to the grammatically tagged representation. In the later part of this report are examples of analysis result.

---

[4] https://www.wycliffe.org/blog/featured/translation-and-technology
[5] https://researchportal.helsinki.fi/fi/datasets/gold-standard-for-the-morpheme-alignment-between-the-hebrew-greek

4.2 Disambiguation and syntactic mapping

The analysed text has ambiguous readings. That is, the word may have more than one legal interpretation when considered in isolation. When we put the word into context, we can decide which one of the alternatives is the correct one in that context. Disambiguation rules, which make use of context, make the selection.

The disambiguated text needs also syntactic labels for showing the role of each word in sentence. The labels hep in identifying the structure of the whole sentence, as well as the structure of smaller constituents, such as the noun phrase and verb phrase. Marking such structures is necessary for constructing correct inflection rules for the target language.

4.3 Multi-word expressions

In machine translation it is normally nexessary to identify and isolate such word clusters, which together constitute one concept. This is especially important, if we translate between structurally very different languages. But now when the source language (Swahili) and target language (Luganda) are both Bantu languages, it is likely that the need for isolating multi-word expressions is less pressing. However, this process cannot be bypassed.

4.4 Lexical transfer

There are at least two alternatives for performing the lexical transfer from Swahili to Luganda. In one method, the transfer is done via English, which functions as a kind of interlingua. This would require a two-step process, which is always more prone to errors than a one-step method.

In my earlier report on the subject I used the two-step method, because I used the resources already available. I had a translation system from Swahili to English. I also had a computational dictionary from English to Luganda. By combining these resources it was possible to perform the transfer beween Swahili and Luganda.

A more secure method would be to perform the transfer directly from Swahili to Luganda. Below I will show how this can be done, although no dictionary from Swahili to Luganda exists.

Because we already have the analysis system of Swahili, which can analyze all words of the Swahili Bible, we extract all lexical words of Bible. Then we map this Swahili lexical list to English, using the conversion lexicon already available. Then we map this English list to the Luganda equivalents available in the dictionary. Now we have a three-language word list of words used in Bible. We go through the list and make corrections where needed. Then we drop the English glosses, and we have a bilingual (Swahili and Luganda) vocabulary of such words, which appear in Bible.

In using this method, one important note must be made. Each word should optimally have only one gloss in another language. More alternative glosses can be added, but the selection must then be done when editing the translation. Another, but less secure, method is to perform semantic disambiguation with CG rules.

Note that when we proceed with the translation process, we all the time work with the abstract (morpheme tags) and lexical (lemmas) representation of the language. This means that for each lexical word there is all morphological and syntactic information attached to it. When we 'translate', that is, map a Swahili lexical word appearing in the context to Luganda, we do that taking into consideration the linguistic information attached to the lexical word. One of the central features is the POS class. A noun must be mapped as noun, adjective as adjective, verb as verb, and so on.

5 Essential features in linguistic transfer

There are several features that must be carefully handled in transfer to target language. I will discuss them one by one.

5.1 Lexical mapping

Care must be taken that a lexical word in source language is mapped to the corresponding lexical word in target language. Linguistic tags available in analysed source language help in correct mapping. The language of Bible is fairly simple, and most words of the same POS category have only one translation, or at least they need only one translation in translating biblical texts.

5.2 Noun class affiliation

Bantu languages have a noun class system, which is a very central feature in constructing word forms. Many nouns have the same noun class affiliation in Swahili and Luganda. However, there are exceptions, and these must be taken care of. One method is that in the bilingual lexical list of words appearing in Bible, for each noun also the noun class affiliation is marked, on both sides. If this is done, the correct information on noun class for each noun is transferred to target text.

There are two methods to mark Bantu noun classes. A much used method is to form classes from pairs, so that a singular and plural form of the noun stem form a class. Using this method, Luganda would have ten classes.

Another method is to use numbers for identifying each noun form, so that singular and plural get different class numbers. Using this method, Luganda has 17 noun classes. This classification method, developed originally as Ur-Bantu, and later Proto-Bantu, classification, is the only sensible method in our case, because it makes the mapping of classes between Bantu languages possible. Therefore, we use numbers for marking Swahili and Luganda noun classes.

Nouns are in key position in syntax, because all members of the noun phrase, such as adjectives, pronouns and part of numbers inflect according to the class of the noun. If the noun is in subject position, it defines the form of the subject prefix in verb according to the class of the subject. Also other verb prefixes, such as relative and object prefix, inflect according to their referent nouns. Therefore, the noun class marking is absolutely necessary for correct translation.

5.3 Word order

Word order rules take care of the correct word order in target language. If we translate from Swahili to English, a large number of word order rules is needed, because the word order in those languages is fundamentally different. The word order of Swahili and Luganda is very similar, and the rule component for controlling the word order is small.

6 What kind of translation we can get with this method?

The resulting translation is not perfect. It has certain strengths and weaknesses.

6.1 Proper names

Bible has 2643 individual proper names and they appear 35,531 times in Bible. The transfer of these names can be done using simple mapping from Swahili to Luganda. The mapping lexicon does the job. And when proper names in these languages do not inflect, the result should not have mistakes.

Consider the difference between this method and the traditional manual translation, where one must remember, or at least check, the correct translation of proper names. The manual method is prone to mistakes, while the transfer method is faultless.

6.2 Nouns

The system translates all the nouns that appear in Bible. The disambiguation system of the source language takes care of the correct choices in analysed text. This disambiguated information is then transferred to target language, taking care also of the possible change of the noun class. When the lexical nouns in target language have the correct noun class information, their singular and plural forms can be produced correctly.

6.3 Noun phrases

When the correct class of the noun is known, all dependent elements, such as adjectives, pronouns, and part of numbers, can be converted to the surface form of that class.

6.4 Verbs

The verb structure of Bantu languages is very complex, and it is a major effort to control the correct surface form, because three of its prefixes are class-sensitive. That is, they must have the form according to the class, which they refer to. The subject prefix gets its form from the subject. The relative prefix may get its form from the subject or object. The object prefix gets its form from the object. Therefore, syntactic information for each word is necessary for finding the correct referent.

How can we produce the correct surface forms of such complex structures? If the language pair would be very different, such as English and Luganda, we would have hard work in adding correct inflection tags for each affix. Now when we have a Bantu language as source text, we already have a lot of information inherited from the source language. The verb prefix structure of Swahili and Luganda is basically similar, with

minor differences. Therefore, instead of going through the long process we can make use of the tags of the source language. When these tags are in language-independent form, they can be converted to the surface form of the target language according to conversion rules. Therefore, even handling the complex werb forms between Swahili and Luganda is not a heavy process.

In the earlier report on this subject I gave an example on how to insert correct inflection tags to verbs. In it, the process required first the insertion of all noun class alternatives of verb morphemes to the reading. Then selection would be made using the context-sensitive selection rules. The method is quite glumsy, and there is now a more elegant solution. For the sake of demonstration, below I have copied a piece from that report (1).

(1)
("<akawaambia>" "ambia" V 1-SG3-SP
[1-SG1-SP]
[1-SG2-SP]
[1-SG3-SP]
[2-PL1-SP]
[2-PL2-SP]
[2-PL3-SP]
[3-SG-SP]
[4-PL-SP]
[5-SG-SP]
[6-PL-SP]
[7-SG-SP]
[8-PL-SP]
[9-SG-SP]
[10-PL-SP]
[11-SG-SP]
[12-SG-SP]
[13-PL-SP]
[14-PL-SP]
[15-SG-SP]
[16-SG-SP]
[17-SG-SP]
[18-SG-SP]
[19-SG-SP]
[20-SG-SP]
[22-PL-SP]
[23-SG-SP]
"<x>" "x" NARR:ka 2-PL3-OBJ
[1-SG1-OBJ]
[1-SG2-OBJ]
[1-SG3-OBJ]
[2-PL1-OBJ]
[2-PL2-OBJ]

[2-PL3-OBJ]
[3-SG-OBJ]
[4-PL-OBJ]
[5-SG-OBJ]
[6-PL-OBJ]
[7-SG-OBJ]
[8-PL-OBJ]
[9-SG-OBJ]
[10-PL-OBJ]
[11-SG-OBJ]
[12-SG-OBJ]
[13-PL-OBJ]
[14-PL-OBJ]
[15-SG-OBJ]
[16-SG-OBJ]
[17-SG-OBJ]
[18-SG-OBJ]
[19-SG-OBJ]
[20-SG-OBJ]
[22-PL-OBJ]
[23-SG-OBJ]
"<x>" "x" { :gambA } @FMAINVintr)

In this example, the Swahili verb form *akawaambia* is analysed and the tags of the subject prefix (1-SG3-SP) and object prefix (2-PL3-OBJ) describe how the prefixes were written in source language. In square brackets are all possible candidates for subject prefix and object prefix of target language. The context-sensitive CG rules then select the correct ones for the target language.

Now, in the new application we do not use selection rules. We add new tags to the end of the reading with CG grammar, and these rules make use of context in the same way as selection rules. In this format, the above description is drastically shorter (2).

(2)
("<akawaambia>" "ambia" V 1-SG3-SP NARR:ka 2-PL3-OBJ { :gambA } @FMAINVintr) [1-SG3-SP] [2-PL3-OBJ]

The tags within square brackets after the original reading were added with adding rules. The tags happen to be identical with the tags inherited from Swahili. In case they were different, the new target language tags would be selected.

The translation proceeds now so that the added prefixes are moved in front of the Luganda verb lemma (3). Note that the TAM marker would come between the subjext prefix and object prefix. In this case it has a zero realization, because it is merged with the subject prefix.

(3)

```
( "<akawaambia>" "ambia" V 1-SG3-SP NARR:ka 2-PL3-OBJ { 1-SG3-
SP+TAM-0+2-PL3-OBJ+:gambA } @FMAINVintr )
```

The surface form of the verb is in (4).

(4)
```
( "<akawaambia>" "ambia" V 1-SG3-SP NARR:ka 2-PL3-OBJ { a+ba+gamba
} @FMAINVintr )
```

7 New implementation of transfer

Below I show the main steps in translating from Swahili to Luganda according to the new implementation. The process is more straightforward and, I hope, the code is more readable than in the earlier implementation.

The example sentence is below (5).

(5)
[MAT28_18/] *Yesu akaja kwao, akasema nao, akawaambia, Nimepewa mamlaka yote mbinguni na duniani.*

The sentence is analysed morphologically, disambiguated, and analysed syntactically using the SALAMA development environment (6). SALAMA uses two-level morphology for morphological analysis and Constraint Grammar (CG) for morphological disambiguation, syntactic mapping, semantic disambiguation, and for adding tags to readings.

(6)
PHASE 1
```
( "<[MAT28_18/]>" LINE-CODE )
( "<*yesu>" "yesu" N PROPNAME AN HUM { *jesus } MALE CAP @SUBJ )
( "<akaja>" "ja" V 1-SG3-SP VFIN NARR:ka [ja] { come } SV MONOSLB
@FMAINVintr )
( "<kwao>" "ao" PRON POSS 17-SG PL3 { to them } @PRON-COMPL )
( "<,>" "," COMMA { , } )
( "<akasema>" "sema" V 1-SG3-SP VFIN NARR:ka [sema] { speak } SVO
@FMAINVtr+OBJ> )
( "<nao>" "nao" PRON PERS CC-PL 2-PL { with them } @OBJ )
( "<,>" "," COMMA { , } )
( "<akawaambia>" "ambia" V 1-SG3-SP VFIN NARR:ka 2-PL3-OBJ OBJ
[amba] { tell } PREFR SVOO @FMAINVintr )
( "<,>" "," COMMA { , } )
( "<*nimepewa>" "pewa" V 1-SG1-SP VFIN PERF:me [pa] { get } PREFR
SVO PASS CAP @FMAINVtr+OBJ> )
( "<mamlaka>" "mamlaka" N 6-PLSG { authority } @OBJ )
( "<yote>" "ote" PRON :OTE 5/6-PL { all } @<PRON )
( "<mbinguni>" "mbingu" N 9/10-SG { heavens } LOC @NLOC )
( "<na>" "na" CC { and } @CC )
( "<duniani>" "dunia" N 9/10-SG { :earth } PLACE LOC @NLOC )
```

```
( "<.$>" ".$" { . } **CLB )
```

The above reading has tags for abstract description of linguistic features. They carry non-ambiguous information about each linguistic feature. The English gloss of the word is within curly braces.

English glosses are converted into Luganda glosses using the conversion lexicon, which contains all lexical words of Bible (7).

(7)
PHASE 2

```
( "<[MAT28_18/]>" LINE-CODE )
( "<*yesu>" "yesu" N PROPNAME { *yesu [1/--SG] } MALE CAP @SUBJ )
( "<akaja>" "ja" V 1-SG3-SP NARR:ka { jja } @FMAINVintr )
( "<kwao>" "ao" PRON POSS 17-SG PL3 { nabo } @PRON-COMPL )
( "<,>" "," COMMA { , } )
( "<akasema>" "sema" V 1-SG3-SP NARR:ka { gambA } @FMAINVtr+OBJ> )
( "<nao>" "nao" PRON PERS CC-PL 2-PL { nabo } @OBJ )
( "<,>" "," COMMA { , } )
( "<akawaambia>" "ambia" V 1-SG3-SP NARR:ka 2-PL3-OBJ { gambA }
@FMAINVintr )
( "<,>" "," COMMA { , } )
( "<*nimepewa>" "pewa" V 1-SG1-SP PERF:me { peeredwA } PASS CAP
@FMAINVtr+OBJ> )
( "<mamlaka>" "mamlaka" N 6-PLSG { yinza [-/14] } @OBJ )
( "<yote>" "ote" PRON :OTE 5/6-PL { onna } @<PRON )
( "<mbinguni>" "mbingu" N 9/10-SG { ggulu [9/10] } LOC @NLOC )
( "<na>" "na" CC { nE } @CC )
( "<duniani>" "dunia" N 9/10-SG { nsi [9/10] } PLACE LOC @NLOC )
( "<.$>" ".$" { . } **CLB )
```

Note that the Luganda nouns have also the noun class affiliation within square brackets. Nouns usually belong to a certain class pair, whereby one class marks singular, and the other marks plural, e.g. [9/10]. If there is a dash '-', it means zero morpheme.

Note that the glosses are in lexical form. For example, many verbs end in capital A. This vowel changes into *e* or *i* in certain inflected forms. Otherwise it realises as *a*

Now we convert the lexical forms into surface forms in stages. First, we select between singular and plural in nouns. This is easy to do, because the information (SG or PL) is in the tag inherited from Swahili. The result is in (8).

(8)
PHASE 3

```
( "<[MAT28_18/]>" LINE-CODE )
( "<*yesu>" "yesu" N PROPNAME { *yesu [1] } MALE CAP @SUBJ )
( "<akaja>" "ja" V 1-SG3-SP NARR:ka { jja } @FMAINVintr )
( "<kwao>" "ao" PRON POSS 17-SG PL3 { nabo } @PRON-COMPL )
( "<,>" "," COMMA { , } )
( "<akasema>" "sema" V 1-SG3-SP NARR:ka { gambA } @FMAINVtr+OBJ> )
( "<nao>" "nao" PRON PERS CC-PL 2-PL { nabo } @OBJ )
( "<,>" "," COMMA { , } )
```

```
( "<akawaambia>" "ambia" V 1-SG3-SP NARR:ka 2-PL3-OBJ { gambA }
@FMAINVintr )
( "<,>" "," COMMA { , } )
( "<*nimepewa>" "pewa" V 1-SG1-SP PERF:me { peeredwA } PASS CAP
@FMAINVtr+OBJ> )
( "<mamlaka>" "mamlaka" N 6-PLSG { yinza [14] } @OBJ )
( "<yote>" "ote" PRON :OTE 5/6-PL { onna } @<PRON )
( "<mbinguni>" "mbingu" N 9/10-SG { ggulu [9] } LOC @NLOC )
( "<na>" "na" CC { nE } @CC )
( "<duniani>" "dunia" N 9/10-SG { nsi [9] } PLACE LOC @NLOC )
( "<.$>" ".$" { . } **CLB )
```

Now we should add inflection tags to those words, which are dependent on nouns and their classes. First we take verbs, which already have inflection codes inherited from Swahili. But it is not always sure whether the same code can be used in Luganda. It depends on whether the referent of the veb prefix belongs to the same noun class in both languages. In the case of human beings, they often do, but not always. Therefore, although it is likely that subject prefixes of Swahili, which refer to a person noun or person pronoun can be accepted as valid tags also in Luganda, because of exceptions we cannot do that. We must check whether the class is the same. We add the class of Luganda subject to the end of the verb reading. This will then be selected as the class of the subject prefix (9).

In the example sentence we have only such werbs that have a human subject. In case the subject would not be human, we would first need to check whether the class of the subject in both languages is the same or not. If the subject is a person pronoun, the class of Swahili can be accepted as such.

(9)
PHASE 4

```
( "<[MAT28_18/]>" LINE-CODE )
( "<*yesu>" "yesu" N PROPNAME { *yesu [1] } MALE CAP @SUBJ )
( "<akaja>" "ja" V NARR:ka 1-SG3-SP { jja } @FMAINVintr ) [1-SG3-
SP]
( "<kwao>" "ao" PRON POSS 17-SG PL3 { nabo } @PRON-COMPL )
( "<,>" "," COMMA { , } )
( "<akasema>" "sema" V NARR:ka 1-SG3-SP { gambA } @FMAINVtr+OBJ> )
[1-SG3-SP]
( "<nao>" "nao" PRON PERS CC-PL 2-PL { nabo } @OBJ )
( "<,>" "," COMMA { , } )
( "<akawaambia>" "ambia" V 1-SG3-SP NARR:ka 2-PL3-OBJ { gambA }
@FMAINVintr ) [1-SG3-SP] [2-PL3-OBJ]
( "<,>" "," COMMA { , } )
( "<*nimepewa>" "pewa" V PERF:me { 1-SG1-SP+peeredwA } PASS CAP
@FMAINVtr+OBJ> )
( "<mamlaka>" "mamlaka" N 6-PLSG { yinza [14] } @OBJ )
( "<yote>" "ote" PRON :OTE 5/6-PL { onna } @<PRON ) P-14
( "<mbinguni>" "mbingu" N 9/10-SG { ggulu [9] } LOC @NLOC )
( "<na>" "na" CC { nE } @CC )
( "<duniani>" "dunia" N 9/10-SG { nsi [9] } PLACE LOC @NLOC )
```

```
( "<.$>" ".$" { . } **CLB )
```

The subject prefix code is now attached to the verb stem when the subject is a person pronoun. To other finite verbs, inflection codes of the target language were added after the verb reading. There are such Swahili TAM codes as NARR:ka and PERF:me, which refer to the narrative prefix *-ka-* and the perfect prefix *-me-*. Narrative in Luganda is merged with subject prefix. Perfect prefix is merged with subject prefix and is realised as *m-*. Therefore, we do not need to move any codes for them.

We move the Luganda inflection codes of verbs in front of the verb stems. We also need to add the inflection code to the pronoun *onna*. This is done using CG rules. The key is the preceding noun *yinza*, which belongs to class 14 (10).

(10)
PHASE 5

```
( "<[MAT28_18/]>" LINE-CODE )
( "<*yesu>" "yesu" N PROPNAME { *yesu [1] } MALE CAP @SUBJ )
( "<akaja>" "ja" V NARR:ka { 1-SG3-SP+jja } @FMAINVintr )
( "<kwao>" "ao" PRON POSS 17-SG PL3 { nabo } @PRON-COMPL )
( "<,>" "," COMMA { , } )
( "<akasema>" "sema" V NARR:ka { 1-SG3-SP+gambA } @FMAINVtr+OBJ> )
( "<nao>" "nao" PRON PERS CC-PL 2-PL { nabo } @OBJ )
( "<,>" "," COMMA { , } )
( "<akawaambia>" "ambia" V NARR:ka { 1-SG3-SP+2-PL3-OBJ+gambA }
@FMAINVintr )
( "<,>" "," COMMA { , } )
( "<*nimepewa>" "pewa" V PERF:me { 1-SG1-SP+peeredwA } PASS CAP
@FMAINVtr+OBJ> )
( "<mamlaka>" "mamlaka" N 6-PLSG { yinza [14] } @OBJ )
( "<yote>" "ote" PRON :OTE 5/6-PL { P-14+onna } @<PRON )
( "<mbinguni>" "mbingu" N 9/10-SG { ggulu [9] } LOC @NLOC )
( "<na>" "na" CC { nE } @CC )
( "<duniani>" "dunia" N 9/10-SG { nsi [9] } PLACE LOC @NLOC )
( "<.$>" ".$" { . } **CLB )
```

Note that it is not enough to add only the class number to the pronoun, because each POS catogory has a separate inflection paradigm for classes. Therefore, P-14 means that the pronoun should have a pronoun prefix of class 14.

When we have all inflection codes attached to lemmas, we can convert them into surface form. Note that in nouns the code that was after the gloss is now moved in front of the stem (11).

While the locative marker *-ni* in Swahili is a suffix, in Luganda it is actually a prefix, but in orthography it is written as separate word. It is also here ambiguous, why both forms are added.

(11)
PHASE 6

```
( "<[MAT28_18/]>" LINE-CODE )
( "<*yesu>" "yesu" N PROPNAME { 1+*yesu } MALE CAP @SUBJ )
```

```
( "<akaja>" "ja" V NARR:ka { 1-SG3-SP+jja } @FMAINVintr )
( "<kwao>" "ao" PRON POSS 17-SG PL3 { nabo } @PRON-COMPL )
( "<,>" "," COMMA { , } )
( "<akasema>" "sema" V NARR:ka { 1-SG3-SP+gambA } @FMAINVtr+OBJ> )
( "<nao>" "nao" PRON PERS CC-PL 2-PL { nabo } @OBJ )
( "<,>" "," COMMA { , } )
( "<akawaambia>" "ambia" V NARR:ka 2-PL3-OBJ { 1-SG3-SP+gambA }
@FMAINVintr )
( "<,>" "," COMMA { , } )
( "<*nimepewa>" "pewa" V PERF:me { 1-SG1-SP+peeredwA } PASS CAP
@FMAINVtr+OBJ> )
( "<mamlaka>" "mamlaka" N 6-PLSG { 14+yinza } @OBJ )
( "<yote>" "ote" PRON :OTE 5/6-PL { P-14+onna } @<PRON )
( "<mbinguni>" "mbingu" N 9/10-SG { LOC 9+ggulu } @NLOC )
( "<na>" "na" CC { nE } @CC )
( "<duniani>" "dunia" N 9/10-SG { LOC 9+nsi } PLACE @NLOC )
( "<.$>" ".$" { . } **CLB )
```

We convert the inflection codes into surface form (12).

(12)
PHASE 7

```
( "<[MAT28_18/]>" LINE-CODE )
( "<*yesu>" "yesu" N PROPNAME { Yesu } MALE CAP @SUBJ )
( "<akaja>" "ja" V NARR:ka { a+jja } @FMAINVintr )
( "<kwao>" "ao" PRON POSS 17-SG PL3 { nabo } @PRON-COMPL )
( "<,>" "," COMMA { , } )
( "<akasema>" "sema" V NARR:ka { a+gamba } @FMAINVtr+OBJ> )
( "<nao>" "nao" PRON PERS CC-PL 2-PL { nabo } @OBJ )
( "<,>" "," COMMA { , } )
( "<akawaambia>" "ambia" V NARR:ka 2-PL3-OBJ { aba+gamba }
@FMAINVintr )
( "<,>" "," COMMA { , } )
( "<*nimepewa>" "pewa" V PERF:me { m+peeredwa } PASS CAP
@FMAINVtr+OBJ> )
( "<mamlaka>" "mamlaka" N 6-PLSG { ama+yinza } @OBJ )
( "<yote>" "ote" PRON :OTE 5/6-PL { g+onna } @<PRON )
( "<mbinguni>" "mbingu" N 9/10-SG {ku/mu e+ggulu } @NLOC )
( "<na>" "na" CC { ne } @CC )
( "<duniani>" "dunia" N 9/10-SG { ku/mu e+nsi } PLACE @NLOC )
( "<.$>" ".$" { . } **CLB )
```

If there would be need to reorder words, it would be done on this point. Now there is no need to do that, why we can retain only the words that are inside curly braces. We also remove morpheme boundary marks (13).

(13)
PHASE 8
[MAT28_18/] *Yesu ajja nabo, agamba nabo, abagamba, mpeeredwa amayinza gonna ku/mu eggulu ne ku/mu ensi.*

8 More examples

To see whether there are more problems to solve, we take two more examples. The examples types taken in (14) are as in PHASE 2 above, where English glosses are replaced with Luganda glosses.

(14)

```
( "<[MAT28_19/]>" LINE-CODE )
( "<*basi>" "basi" ADV { kale } CAP @ADVL )
( "<,>" "," COMMA { , } )
( "<enendeni>" "enenda" V IMP { gendA } IMP-PL2 @FMAINVintr )
( "<,>" "," COMMA { , } )
( "<mkawafanye>" "fanya" V 2-PL2-SP CONSEQ:ka-e 2-PL3-OBJ { fuulA
} @FMAINVtr+OBJ> )
( "<mataifa>" "taifa" N 5/6-PL { wanga [-/14] } @OBJ )
( "<yote>" "ote" PRON :OTE 5/6-PL { onna } @<PRON )
( "<kuwa>" "wa" V INF MOD-CAN { li } AUX-WA @-FMAINV-n )
( "<wanafunzi>" "mwanafunzi" N 2-PL { yigilizwa [2-PL] } @NCOMPL )
( "<,>" "," COMMA { , } )
( "<mkiwabatiza>" "batiza" V 2-PL2-SP COND-IF 2-PL3-OBJ { :batizA
} @FMAINVtr-OBJ> )
( "<kwa>" "kwa" PREP { mu:- } @PREP )
( "<jina>" "jina" N 5/6-SG { linnya [5/6-SG] } @NCOMPL )
( "<la>" "la" GEN-CON 5-SG { -a } @GCON ) G-5
( "<*baba>" "baba" N TITLE { *kitaffe [1/--SG] } @<GN )
( "<,>" "," COMMA { , } )
( "<na>" "na" CC { nE } @CC )
( "<la>" "la" GEN-CON 5-SG { -a } @GCON ) G-5
( "<*mwana>" "mwana" N PROPNAME { *omwana [1/--SG] } MALE CAP @<GN
)
( "<,>" "," COMMA { , } )
( "<na>" "na" CC { nE } @CC )
( "<la>" "la" GEN-CON 5-SG { -a } @GCON ) G-5
( "<*roho_*mtakatifu>" "roho_*mtakatifu" N PROPNAME {
*omwoyo_*omutukuvu [1/--SG] } MALE CAP @<GN )
( "<;>" ";" SEMI-COLON **CLB )
( "<[MAT28_20/]>" LINE-CODE )
( "<na>" "na" CC { nE } @CC )
( "<kuwafundisha>" "fundisha" V INF 2-PL3-OBJ { yigilizA } CAUS @-
FMAINV-n )
( "<kuyashika>" "shika" V INF 6-PL-OBJ { kwatA } @-FMAINV-n )
( "<yote>" "ote" PRON :OTE 6-PL { onna } @OBJ )
( "<niliyowaamuru>" "amuru" V 1-SG1-SP PAST 3/4-PL-REL 2-PL2-OBJ {
lagilA } @FMAINVtr+OBJ> )
( "<ninyi>" "ninyi" PRON PERS PL2 { mmwe } @OBJ )
( "<;>" ";" SEMI-COLON **CLB )
( "<na>" "na" CC { nE } @CC )
( "<tazama>" "tazama" V IMP { labA } @FMAINVtr-OBJ> )
( "<,>" "," COMMA { , } )
```

```
( "<mimi>" "mimi" PRON PERS SG1 { nze } @SUBJ )
( "<nipo>" "nipo" V-BE SG1-SP { ndi } LOC-16 @FMAINVintr )
( "<pamoja>" "pamoja" ADV { wamu } @ADVL )
( "<nanyi>" "nanyi" PRON PERS CC-PL 2-PL2 { nammwe } @PRON-COMPL )
( "<siku>" "siku" N 9/10-PL { nnaku [9/10-PL] } TIME @NCOMPL )
( "<zote>" "ote" PRON :OTE 9/10-PL { onna } @<PRON ) A-10
( "<,>" "," COMMA { , } )
( "<hata>" "hata" ADV { until } @ADVL )
( "<ukamilifu>" "ukamilifu" N 11-SG { perfection } @NCOMPL )
( "<wa>" "wa" GEN-CON 11-SG { -a } @GCON ) G-11
( "<dahari>" "dahari" N 9/10-SG { aeon } @<GN )
( "<.$>" ".$" { . } **CLB )
```

All information for producing surface forms is now there. Subjects in all clauses are person pronouns. We only need to move the information from Swahili as part of the verb. If inflections tags were not on the word, such as pronouns and genitive connectors, the inflection tag was added after the reading.

In (15), all inflection tags are attached to lemmas.

(15)

```
( "<[MAT28_19/]>" LINE-CODE )
( "<*basi>" "basi" ADV { kale } CAP @ADVL )
( "<,>" "," COMMA { , } )
( "<enendeni>" "enenda" V IMP { IMP-PL2+gende } @FMAINVintr )
( "<,>" "," COMMA { , } )
( "<mkawafanye>" "fanya" V CONSEQ:ka-e { 2-PL2-SP+2-PL3-OBJ+fuule
} @FMAINVtr+OBJ> )
( "<mataifa>" "taifa" N 5/6-PL { N-14+wanga } @OBJ )
( "<yote>" "ote" PRON :OTE 5/6-PL { A-6+onna } @<PRON )
( "<kuwa>" "wa" V INF MOD-CAN { INF+li } AUX-WA @-FMAINV-n )
( "<wanafunzi>" "mwanafunzi" N 2-PL { N-2+yigilizwa } @NCOMPL )
( "<,>" "," COMMA { , } )
( "<mkiwabatiza>" "batiza" V COND-IF { 2-PL2-SP+2-PL3-OBJ+batizA }
@FMAINVtr-OBJ> )
( "<kwa>" "kwa" PREP { mu:- } @PREP )
( "<jina>" "jina" N 5/6-SG { N-5+nnya } @NCOMPL )
( "<la>" "la" GEN-CON 5-SG { G-5+a } @GCON ) G-5
( "<*baba>" "baba" N TITLE { N-1+*kitaffe } @<GN )
( "<,>" "," COMMA { , } )
( "<na>" "na" CC { nE } @CC )
( "<la>" "la" GEN-CON 5-SG { G-5+a } @GCON ) G-5
( "<*mwana>" "mwana" N PROPNAME { N-1+*omwana } MALE CAP @<GN )
( "<,>" "," COMMA { , } )
( "<na>" "na" CC { nE } @CC )
( "<la>" "la" GEN-CON 5-SG { G-5+a } @GCON ) G-5
( "<*roho_*mtakatifu>" "roho_*mtakatifu" N PROPNAME {
*omwoyo_*omutukuvu } MALE CAP @<GN )
( "<;>" ";" SEMI-COLON **CLB )
( "<[MAT28_20/]>" LINE-CODE )
( "<na>" "na" CC { nE } @CC )
```

```
( "<kuwafundisha>" "fundisha" V { INF+2-PL3-OBJ+yigilizA } CAUS @-
FMAINV-n )
( "<kuyashika>" "shika" V { INF+6-PL-OBJ+kwatA } @-FMAINV-n )
( "<yote>" "ote" PRON :OTE { A-6+onna } @OBJ )
( "<niliyowaamuru>" "amuru" V { 1-SG1-SP+PAST+4-PL-REL+2-PL2-
OBJ+lagilA } @FMAINVtr+OBJ> )
( "<ninyi>" "ninyi" PRON PERS PL2 { mmwe } @OBJ )
( "<;>" ";" SEMI-COLON **CLB )
( "<na>" "na" CC { nE } @CC )
( "<tazama>" "tazama" V IMP { labe } @FMAINVtr-OBJ> )
( "<,>" "," COMMA { , } )
( "<mimi>" "mimi" PRON PERS SG1 { nze } @SUBJ )
( "<nipo>" "nipo" V-BE { 1-SG1-SP+di } LOC-16 @FMAINVintr )
( "<pamoja>" "pamoja" ADV { wamu } @ADVL )
( "<nanyi>" "nanyi" PRON PERS CC-PL 2-PL2 { nammwe } @PRON-COMPL )
( "<siku>" "siku" N { N-10+nnaku } TIME @NCOMPL )
( "<zote>" "ote" PRON :OTE { A-10+onna } @<PRON ) A-10
( "<,>" "," COMMA { , } )
( "<hata>" "hata" ADV { until } @ADVL )
( "<ukamilifu>" "ukamilifu" N 11-SG { perfection } @NCOMPL )
( "<wa>" "wa" GEN-CON 11-SG { -a } @GCON ) G-11
( "<dahari>" "dahari" N 9/10-SG { aeon } @<GN )
( "<.$>" ".$" { . } **CLB )
```

Luganda words are converted to surface form (16). Note that in imperative, consecutive and subjunctive forms the verb-final A changes to e.

(16)

```
( "<[MAT28_19/]>" LINE-CODE )
( "<*basi>" "basi" ADV { *kale } CAP @ADVL )
( "<,>" "," COMMA { , } )
( "<enendeni>" "enenda" V IMP { mu+gende } @FMAINVintr )
( "<,>" "," COMMA { , } )
( "<mkawafanye>" "fanya" V CONSEQ:ka-e { mu+ba+fuule }
@FMAINVtr+OBJ> )
( "<mataifa>" "taifa" N 5/6-PL { ama+wanga } @OBJ )
( "<yote>" "ote" PRON :OTE 5/6-PL { g+onna } @<PRON )
( "<kuwa>" "wa" V INF MOD-CAN { oku+li } AUX-WA @-FMAINV-n )
( "<wanafunzi>" "mwanafunzi" N 2-PL { aba+yigilizwa } @NCOMPL )
( "<,>" "," COMMA { , } )
( "<mkiwabatiza>" "batiza" V COND-IF { mu+ba+batiza } @FMAINVtr-
OBJ> )
( "<kwa>" "kwa" PREP { mu } @PREP )
( "<jina>" "jina" N 5/6-SG { li+nnya } @NCOMPL )
( "<la>" "la" GEN-CON 5-SG { ly+a } @GCON ) G-5
( "<*baba>" "baba" N TITLE { *kitaffe } @<GN )
( "<,>" "," COMMA { , } )
( "<na>" "na" CC { ne } @CC )
( "<la>" "la" GEN-CON 5-SG { ly++a } @GCON ) G-5
( "<*mwana>" "mwana" N PROPNAME { *omwana } MALE CAP @<GN )
( "<,>" "," COMMA { , } )
```

```
( "<na>" "na" CC { ne } @CC )
( "<la>" "la" GEN-CON 5-SG { ly+a } @GCON ) G-5
( "<*roho_*mtakatifu>" "roho_*mtakatifu" N PROPNAME {
*omwoyo_*omutukuvu } MALE CAP @<GN )
( "<;>" ";" SEMI-COLON **CLB )
( "<[MAT28_20/]>" LINE-CODE )
( "<na>" "na" CC { ne } @CC )
( "<kuwafundisha>" "fundisha" V { oku+ba+yigilizA } CAUS @-FMAINV-
n )
( "<kuyashika>" "shika" V { oku+ga+kwata } @-FMAINV-n )
( "<yote>" "ote" PRON :OTE { g+onna } @OBJ )
( "<niliyowaamuru>" "amuru" V { na+ba+lagila } @FMAINVtr+OBJ> )
( "<ninyi>" "ninyi" PRON PERS { mmwe } @OBJ )
( "<;>" ";" SEMI-COLON **CLB )
( "<na>" "na" CC { ne } @CC )
( "<tazama>" "tazama" V IMP { labe } @FMAINVtr-OBJ> )
( "<,>" "," COMMA { , } )
( "<mimi>" "mimi" PRON PERS SG1 { nze } @SUBJ )
( "<nipo>" "nipo" V-BE { n+di } LOC-16 @FMAINVintr )
( "<pamoja>" "pamoja" ADV { wamu } @ADVL )
( "<nanyi>" "nanyi" PRON PERS CC-PL 2-PL2 { nammwe } @PRON-COMPL )
( "<siku>" "siku" N { e+nnaku } TIME @NCOMPL )
( "<zote>" "ote" PRON :OTE { z+onna } @<PRON ) A-10
( "<,>" "," COMMA { , } )
( "<hata>" "hata" ADV { until } @ADVL )
( "<ukamilifu>" "ukamilifu" N 11-SG { perfection } @NCOMPL )
( "<wa>" "wa" GEN-CON 11-SG { lw+a } @GCON ) G-11
( "<dahari>" "dahari" N 9/10-SG { aeon } @<GN )
( "<.$>" ".$" { . } **CLB )
```

After final pruning we get the translation (17).

(17)
[MAT28_19/] *Kale, mugende, mubafuule amawanga gonna okuli abayigilizwa, mubabatiza mu linnya Kitaffe, ne Omwana, ne Omwoyo Omutukuvu.*
[MAT28_20/] *ne okubayigiliza okugakwata gonna nabalagila mmwe ne labe, nze ndi wamu nammwe ennaku zonna, <until> <perfection> lwa <aeon>.*

If need be, translations can be arranged verse by verse under each other, such as in (18).

(18)
Lug: [MAT28_18/] Yesu ajja nabo, agamba nabo, abagamba, mpeeredwa amayinza gonna ku/mu eggulu ne ku/mu ensi.
Swa: [MAT28_18/] Yesu akaja kwao, akasema nao, akawaambia, Nimepewa mamlaka yote mbinguni na duniani.
Eng: [MAT28_18/] Jesus came to them, he spoke with them, he told them, I have been given all authority in the heavens and on the earth.

Lug: [MAT28_19/] Kale, mugende, mubafuule amawanga gonna okuli abayigilizwa, mubabatiza mu linnya Kitaffe, ne Omwana, ne Omwoyo Omutukuvu
Swa: [MAT28_19/] Basi, enendeni, mkawafanye mataifa yote kuwa wanafunzi, mkiwabatiza kwa jina la Baba, na la Mwana, na la Roho Mtakatifu;
Eng: [MAT28_19/] then, go, and do all nations be the pupils, if/when you baptize them in the name of Father, and of Son, and of Holy Spirit;

Lug: [MAT28_20/] ne okubayigiliza okugakwata gonna nabalagila mmwe ne labe, nze ndi wamu nammwe ennaku zonna, <until> <perfection> <aeon>.
Swa: [MAT28_20/] na kuwafundisha kuyashika yote niliyowaamuru ninyi; na tazama, mimi nipo pamoja nanyi siku zote, hata ukamilifu wa dahari.
Eng: [MAT28_20/] and to teach them to take hold of all which I commanded you; and look, I am together with you all days, until the perfection of the aeon.

9 Discussion

The above description shows how the machine translation system from Swahili to English can be applied also to translation to a third language. English does not play here any other role than to provide disambiguated glosses for each Swahili word. Swahili glosses could also be replaced directly with Luganda glosses. So far, no unsolvable problem was encountered.

The code looks messy, but the large number of tags is there in case they are needed. I have shown in phases how the translation proceeds. By building the translation system as gradually proceeding process, is easy to follow its function in each step, and the correction of code is possible.

In this test we translated only three Bible verses. This sample contains only a fraction of problem types, which occur in translating the whole Bible. They are just examples of how various translation problems can be solved in a systematic way. It is vastly more efficient to solve translation problems on an abstract level than on case level, because the solution applies often to a large number of concrete cases.

We return back to the title of this paper. Is a machine translation system feasible in Bible translation? If the alternatives are ten years of manual translation of the Bible and the raw translation with computer plus manual correction, the answer should be clear.

Advantages of rule-based machine translation include full covarage of vocabulary, error-free handling of proper names, and when errors occur, they occur very systematically. Even correcting modules for post-editing can be constructed, which can be connected directly to the translation system, or they can be used separately according to need.

The bottleneck in this translation method is that where does one get such a reliable analysis and translation system, which could be used as the base module. I know no one for any African language except for SALAMA, which I have developed and used.