Technical Reports in Language Technology
Report No 4, 2009
http://www.njas.helsinki.fi/salama

Enriching text with tone marks: An application to Kinyarwanda language

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

Abstract

The absence of tone marks in text can be considered an instance of defective writing. Yet most writing systems of tone languages in Africa do not mark tone. This adds to ambiguity, which must be resolved on the basis of the context. Using tone-marked lexicons and tone rules the ambiguity can be resolved. This paper deals with a further problem, i.e. how to insert tone marks into a text without tone marking. Such a tool is needed in text-to-speech applications, and whenever a text needs to be provided with tone marks. Three approaches are briefly discussed, and one of them is selected for discussion, implementation and demonstration. The implementation makes use of Finite State Methods in analysis, Constraint Grammar in disambiguation, and regular expressions in writing tone rules. The system was implemented using Kinyarwanda verb morphology for testing and demonstration. It is assumed that the approach is suitable especially for tone languages that have lexical and grammatical tone.

Key Words: tonology, tone marking, language technology, Finite State Methods

1 Introduction

Tones are a characteristic feature in the majority of African languages, including most Bantu languages. The role of tones in a language ranges from a simple pitch/accent type emphasis to elaborate tone patterns with grammatical roles. A specific characteristic of tones is that they are not segmental features, unlike sequences of phonemes that are represented in normal writing with orthographic sequences of characters. In other words, tones cannot be marked into the lexicon in the way they appear in surface realizations.

Despite the central significance of tones in Bantu languages, the tones are often not marked in writing. As a consequence, writing does not indicate in any way how tone behaves in various words. This may cause difficulties in comprehension for readers, although some experimental studies show that the absence of tone marks has very little to do with comprehension, and that adding tone marks in text may in some cases slow down reading speed (Bird 1999a). However, the absence of tone marks is problematic when we consider developing computational language processing tools for such languages. Although there is no uniformity in tone marking, when it is applied, the total absence of marking is even more problematic. General guidelines in tone marking would be beneficial (Bird 1999b), but from the computational viewpoint the absence of such guidelines is not fatal. Text-to-speech applications would greatly benefit if tones would

be appropriately marked in text. In brief, we need a tool that inserts tone marks correctly into the normal orthographic text.

Below I shall describe a method for inserting tone marks into the text that originally is without tone marking. The primary source material of tone description is from Kimenyi (2002). The model was implemented and tested with Kinyarwanda, a Bantu language in the Great Lakes region.[1] Tone behaviour in Kinyarwanda is semi-complex. In this implementation, only high tone is marked when it is present. Whether distinction should be made between low tone and toneless units does not concern us here. The model can be applied also to more complex tone systems.

Tones in Kinyarwanda can be divided into lexical and grammatical tones. On the lexical level, word stems form two groups in regard to tone: toneless stems and tone-bearing stems. Such tones that are associated with lexical segments of word stems are lexical tones. Also other morphemes, for example prefixes, can be considered to be associated with lexical tones. In this implementation, however, tones associated with prefixes and suffixes are ignored in the lexicon, and only the tone associated with the word stem is defined as a default lexical tone. No other types of lexical tone are assumed. In the subsequent examples, the symbol H, placed after the tone-bearing vowel, is used for marking the high tone.

Lexical tones may be used for distinguishing segmentally identical word stems, such as in (1).

(1)
igikoHokoH 'animal'
igikooko 'big basket'

umuryaHango 'door'
umuryaango 'family'

Grammatical tones function as markers of various grammatical features. Also here, a word-form may have more than one tone pattern, depending on its grammatical function. In (2), there are three segmentally identical verb forms, but each of them has a different tone pattern.[2]

(2)

bazaaririimba	+SPba+AFFzaaririimba
bazaaririimba	+SPba+PARTFUTzaHaHriHriimba
bazaaririimba	+SPbaH+RELFUTzaHaHriHriimba

[1] I am greatly indebted to Jackson Muhirwe at Makerere University in Uganda for allowing me to use the lexical and grammatical material on tones in Kinyarwanda. This material was extracted from Kimenyi (2002), to which full credit is given.

[2] In this paper, the examples of morphological analysis were produced using the Xerox Finite State Tools (Beesley and Karttunen 2003).

2 Approach

The assumption is that each word stem in the lexicon has a default tone assignment. It either has a high tone or does not have it. Therefore, the high tone mark (in this implementation 'H') is inserted after the tone-bearing unit of the word stem. The tone bearing unit is a vowel. Other lexical units, such as prefixes and suffixes, are assumed to be without tone in the lexicon.

Note that the assumption described above was made purely from the viewpoint of implementing the system. It has nothing to do with a linguistic tone theory, where also affixes may be assumed to be tone-bearing units in the lexicon. In fact, assuming a default tone on some of the prefixes would make tone rules more simple and more 'credible' than some of the rules described below.

The surface realization of tone patterns is implemented with rules that make use of the grammatical features of the word-form and of the lexical type of the stem (whether tone-bearing or toneless). Such rules are applied after the segmental analysis of the word-form and after disambiguation.

In the current implementation, it is assumed that tone rules apply word-internally, without taking into consideration the word before or after. If needed, the influence of neighbouring words on the realization of the tone pattern of a word can also be taken into account.

3 Alternative implementations

The major problem in implementing the tone insertion system is that, instead of rewriting something to something else, we should write something out of nothing. Toneless text is an instance of defective writing, analogical to Semitic writing systems without vowel marks. The task of inserting tone marks correctly is challenging, because tones are not segmental units with a fixed place within a segmental string.

In theory, there is more than one way to implement the system. Below I shall describe one unsuccessful approach and two successful ones.

(a) It would be tempting to implement the system with finite state methods, so that on the lexical side we would describe the lexical tone, and in addition to that also those grammatical tone-bearing prefix candidates that are either triggered or blocked by rules using grammatical features as criteria. These tone insertion rules would then be mapped on the lower side of the network, thus enriching it with tone marks, while the segmental string would remain surface-like. In this solution, tone marks would be inserted using replace rules in conjunction with the finite state description of the language.

The problem that arises in using this method is that the linguistic tags are lost before ambiguity is resolved. In rule-based disambiguation, linguistic tags are utmost important. Also in the developing phase, it would be difficult to track mistakes in analysis when tags are not available. Therefore, this method cannot be recommended.

(b) In the second alternative, the morphological analyzer of the language is implemented using regular expressions without a rule component that would handle

morpho-phonological variation. The solution that does not make use of alternation rules makes the lexicon quite complex, especially in Bantu languages with rich morphology. On the other hand, it produces directly lexical strings that are segmentally identical with surface strings, enriched with lexical tone marks and grammatical tags. Tone rules for assigning the final tone pattern could then be written using this enriched representation. This is a straightforward approach without complex composition sequences. Disambiguation could be carried out after morphological analysis, using linguistic tags, and tone insertion rules for inserting grammatical tone marks could be run after that. This could be done also in a different order, so that tone rules would be applied first and the disambiguation carried out thereafter.

(c) In the third alternative, finite state methods with morpho-phonological alternation rules are used in implementation. In this solution, the lexicon structure is more simple than in solution (b) above, but it is more complex to implement.

In this study, solution (c) is applied, because finite state methods in language description are widely used and they are efficient in language processing. In the finite state solution, two languages are constructed, one representing the lexical language (upper language), and the other representing a surface-like language (lower language).[3] When word-forms are concatenated from the sub-lexicons of the lexicon system, the concatenated upper side string contains all morphemes of the word-form, the lexical tone marks of the stem, and linguistic tags.[4] The lower side string is different from the upper side string in that it does not have tags. Otherwise it is identical with the upper side string.[5] The lower side string is surface-like in that it has the morphemes of the word. Yet it is different from the orthographic word in two respects. It has lexical tone marks that are missing in the orthographic word. Also the character segments are not necessarily correct, because no morpho-phonological alternation rules have been applied.

In order to map the lower language of the above network on the orthographic words, we need a set of rules that handle the morpho-phonological alternation and map the tone marks to zero on the surface.

Below is shown how this can be done using the Xerox Finite State tool package (Karttunen and Beesley 2003). Within this system, the lexicon is constructed so that it has an upper and lower side representation of the language. The upper side contains all the linguistically relevant information, including tags, while the lower side is designed so that the alternation rules can be applied to it. In other words, the lower side of the lexicon is temporary, because it will be composed with the alternation rules, and the result of the composition is the true surface language on the lower side. An example of analysis with the composed lexicon (Network 1), without rules, is in (3). Zeros are added on the lower side to facilitate character alignment.

---

[3] Examples of the upper and lower representations will be presented and discussed below.
[4] In this implementation, each tag is composed of the plus sign '+' followed by one or more upper case characters. Lower case letters are not used in tags.
[5] Note that, unlike often is the case, here the upper side must have all lexical morphemes properly represented, character by character, because this is the representation that will be modified into the final tone-marked text.

(3)
```
Upper: +MNAMnti+SPbakioHogosha
Lower: 00000nti000bakioHogosha
```

The upper side has the lexical string with a tone mark on the stem and a tag indicating that it is a 'not any more' (NAM) form in main clause (M). The lower side is identical with the upper side, except that the lower side does not have linguistic tags. However, neither of the forms is a surface form, because alternation rules have not been applied.

When the alternation rules have been composed with the lower side of the lexicon, in the network resulting from the composition (Network 2), the upper language remains in the form as it is in the original lexicon, but the lower side represents the surface language. This is the format that is normally used in language analysis systems. An example is in (4).

(4)
```
Upper: +MNAMnti+SPbakioHogosha
Lower: 00000nti000bacyo0ogosha
```

Now we have a surface language (i.e. orthographic form) on the lower side, which is mapped on the lexical language on the upper side. It is this upper side representation that we would need to preserve, because it has all the information that we need for writing the rules for grammatical tones. But as we can see, although the upper side has all the features needed for writing tone rules, it is not segmentally correct. The alternation rules were applied on the lower side of the lexicon (Network 1), and they did not do anything on the upper side of the lexicon. Therefore, we need to apply morpho-phonological rules also to the upper side. Because the upper side has, in addition to the morphemes of the word, also tags, the same rules may not necessarily suit for applying on the upper side. Therefore, another set of rules is needed, where the tags are taken into consideration in defining context constraints for rules. With the help of the second set of rules also the upper side of Network 2 can be converted segmentally surface-like, while the lower side remains as the original surface language. After applying the second set of rules on the upper side of Network 2, we get a network, which produces forms as in (5)

(5)
```
Upper: +MNAMnti+SPbacyoHogosha
Lower: 00000nti000bacyo0ogosha
```

As a result we have a network (Network 3), where the lower side represents the standard orthographic language, and the upper side also represents the orthographic language, enriched with grammatical tags and lexical tone marks. This is the network that we use, and the output is what we need for writing tone rules.

4 Tone rules

Because tone rules operating on verb constructions exhibit several types of tone processes, we restrict the discussion on verbs. In the test material we had six verbs, two of them consonant-initial tone-bearing verbs, two consonant-initial toneless verbs, one vowel-initial tone-bearing verb, and one vowel-initial toneless verb. These verbs

represent the major types of verbs in Kinyarwanda language. The verbs to be tested with are in (6).

(6)
riHriimba 'sing'
tabaara 'defend'
kuHunguuta 'shake'
shuumbuusha 'compensate'
oHogosha 'shave'
oongera 'repeat'

The following types of tone processes were identified in the test material:[6]

(a) no tone change

(b) tone copied to the nearest tone-bearing unit on the left

(c) tone double-copied to the left, that is, to two nearest tone-bearing units on the left

(d) tone triple-copied to the left, that is, to three nearest tone-bearing units on the left

(e) tone quadruple-copied to the left, that is, to four nearest tone-bearing units on the left

(f) tone copied to the nearest tone-bearing unit on the right

(g) tone shifted to the nearest tone-bearing unit on the left and then double-copied to the left.[7]

(h) tone copied to the nearest tone-bearing unit on the left and right

(i) tone neutralisation (tone deletion)

(j) tone shifted over several tone-bearing units to the left and then double-copied to the nearest tone-bearing unit on the left.[8]

Note that some of the rules seem to have little linguistic motivation. These rules should not be taken as examples of 'real' tone processes. They should rather be understood as interpretation on what seems to happen when no lexical tones are assumed to prefixes.

Rules were formulated on the basis of all relevant TAM (tense/aspect/mood) forms (21) of the language. Below are examples of how tone rules operate.

In future subjunctive (7), there is tone loss. In present imperfective, tone is copied to the nearest tone-bearing unit on the left.

---

[6] Please take into consideration that the description of tone processes below sometimes sounds strange, even counter intuitive. The reason is that while the default lexical tone was assumed to be only on the stem, and not on any of the affixes, the tone process has to be 'explained' using the default tone as a starting point. For the sake of simplicity, also the insertion of the double vowel is omitted, as well as some changes in the end of the verb.
[7] This interpretation obviously does not satisfy all tonologists. However, this is what seems to happen if we do not assume any default tone on verb prefixes.
[8] Interpretation sounds utmost weird.

(7)
bararíriimba +SBJNNFUTbara@ririimba
bararíriimba +SPba+IMPPRESraH@riHriimba

In complementless recent past (8), tone is copied to the nearest tone-bearing unit on the left. Note that when the subject prefix *ba* (+SP) is followed by *a*, one of the vowels is deleted. In this description, the rule removes the vowel a in the subject prefix.

(8)
baaríriimbye +SPb+PRPASTa+COMPa@riHriimb+PERFPRESye
baaríriimbye +SPb+RPASTa+COMPaH@riHriimb+PERFPRESye

In affirmative (9), there is tone deletion. In participial future (9), tone is double-copied to the left. In relative future (9), tone is triple-copied to the left.

(9)
bazaaríriimba +SPba+AFFzaa@ririimba
bazaaríriimba +SPba+PARTFUTzaHaH@riHriimba
bazaaríriimba +SPbaH+RELFUTzaHaH@riHriimba

Also in the so-called 'not yet' tense (10), the tone is triple-copied to the left.

(10)
ntibaraaríriimba +NOTYETntibaHraHaH@riHriimba

The same applies to the so-called 'still tense' (11). Tone is triple-copied to the left.

(11)
baracyaaríriimba +SPba+STILLraHcyaHaH@riHriimba

In realis future conditional (12), tone is quadruple-copied to the left. This is the most extensive tone spreading phenomenon found in the test material.

(12)
nibazaaríriimba +REALniHbaH+CONDFUTzaHaH@riHriimba

In several tenses tone is neutralised. This happens in imperative (13).

(13)
ríriimba +IMP@ririimba

It also happens in affirmative (14).

(14)
bazaaríriimba +SPba+AFFzaa@ririimba

It happens in the 'not any more' tense in main clause (15) and subordinate clause (15).

(15)
ntibakíriríimba +MNAMntibaki@ririimba
batakíriríimba +SNAMbatakiki@ririimba

Tone is deleted also in negative recent past (16).

(16)
ntibaaririimbye +NEGRECPASTntibaa@ririimb+PERFPRESye

Also in hortative (17), tone is deleted.

(17)
barakaririimba +HORTbaraka@ririimba

In subjunctive near future (18), tone is deleted as well.

(18)
baraririimbe +SBJNNFUTbara@ririimbe

In consecutive hortative (19), tone is first shifted to left and then double-copied to the left.

(19)
bookaririimba +CONSHORTboHoHkaH@ririimba

When tone rules work as they should, the grammatical tags can be removed and the temporary high tone mark rewritten as a vowel with an acute diacritic. This is a trivial task. Some of the above examples are rewritten below (20-26) with tone-marked orthography.

(20)
baraririimba baraririimba
baraririimba baráríriimba

(21)
bazaaririimba bazaaririimba
bazaaririimba bazááríriimba
bazaaririimba bázááríriimba

(22)
ntibaraaririimba ntibárááríriimba

(23)
baracyaaririimba barácyááríriimba

(24)
nibazaaririimba níbázááríriimba

(25)
baraririimbe baraririimbe

(26)
bookaririimba bóókáririimba

5 Disambiguation

The examples above show that segmentally identical word-forms may have more than one tone pattern, each with separate interpretation. Therefore, in addition to marking

tones, the system needs also a disambiguation component, so that the correct tone pattern can be selected in each context.

The approach described above was selected partly because it provides efficient means for rule-based disambiguation. It must be carried out in the phase, where linguistic tags are present, because they are important criteria in defining disambiguation rules. Disambiguation can be carried out before or after applying the tone rules.

In the experiment discussed here, disambiguation was carried out using rule-based methods of Constraint Grammar (CG). The major rule types in CG are (a) selection rules and (b) removal rules. A selection rule selects a reading, if context constraint tests allow it, and other readings are deleted. A removal rule removes one or more readings, if context tests succeed, and other readings are left intact.

Because the disambiguation parser expects that tags are surrounded by word boundaries (= blank) and that each reading is on its own line, the output of the morphological analyzer has to be modified accordingly. For example the Constraint Grammar parsers CG-2 (Tapanainen 1996) and CG-3 (Bick 2006) expect the format, where each tag, including the base form of the word, has a word boundary on both sides. This can be implemented in various ways.[9]

The output of the morphological parser is in (27).

(27)

```
baaririimbye        +SPb+PRPASTa+COMPa@riHriimb+PERFPRESye
baaririimbye        +SPb+RPASTa+COMPaH@riHriimb+PERFPRESye
```

In one solution (28), all grammatical tags are moved as a continuous sequence and the word-form is concatenated as one string in the end.

(28)

```
baaririimbye
      "riHriimba" +SP +PRPAST +COMP +PERFPRES baariHriimbye
      "riHriimba" +SP +RPAST +COMP +PERFPRES baaHriHriimbye
```

In another solution (29), boundary markers, that is spaces, are added between the grammatical tags and the morphemes.

(29)

```
baaririimbye
      "riHriimba" +SP b +PRPAST a +COMP a riHriimb +PERFPRES ye
      "riHriimba" +SP b +RPAST a +COMP aH riHriimb +PERFPRES ye
```

Both formats suit as input for the CG-2 and CG-3 parsers. When disambiguation has been performed, that is, only one interpretation for a word-form is left, the reading can be pruned so that only the surface form is retained. In the case of (28), it means removing everything else except the last element, which is the surface word-form. In (29), the surface form must be concatenated from the separate morphemes, and other elements

[9] The required post-processing can be carried out with several programming languages that support regular expressions.

must be removed. In both cases the result is the same, that is, the correct surface form with tone marking.

6 Conclusion

The absence of tone marks in writing is an instance of defective writing. This increases ambiquity in morphological analysis. In analysing text, the ambiguity can be resolved using context-based disambiguation rules. However, the absence of tone marks is problematic, for example, in text-to-speech applications, because the correct pronunciation is essential in speech. In this paper we have presented a method for enriching text with tone marks. Three possible approaches were suggested. One of them, however, was found problematic, because it does not provide efficient methods for disambiguation. Out of the two other methods, the one implemented using the Xerox Finite State Tools was selected for discussion.

The guiding idea is that, while the selected tool package produces a two-level implementation, the lower language should represent the orthographic writing of the language and the upper language should have the same segmental form, character by character, as the lower language. In addition to this, the upper language should have the correct tone marking and linguistic tags, the latter for facilitating rule-based disambiguation. The aim was achieved stage by stage, composing first the lower side of the lexicon with a set of morpho-phonological replace rules. In the network that resulted from the composition, the lower side represented the orthographic text, and the upper level had linguistic tags and lexical tone marks (not final tone marks). Another rule set was then composed with the upper side of the current network. This was done for 'converting' the lexical character sequence into surface form. Now in this stage, on one side of the network there was the orthographic language, and on the other side there also was the orthographic language, enriched with lexical (not final) tone marks and linguistic tags.

The orthographic text was then given as input to this version of the network. In this stage, tone rules were applied, after which each reading of the word-form had tone marks attached to the analysed word-form.

Then the output, produced by the Xerox formalism, was reformatted to meet the requirements of the Constraint Grammar parser. The disambiguation was carried out using rules written in the Constraint Grammar environment. The disambiguation included morphological as well as tonal ambiguity. After this, each word-form had only one reading.

When disambiguation had been performed, the tags were removed, and only the surface form, enriched with final tone marks, was left.

The method was implemented and tested using Kinyarwanda language, where only high tone was assumed, and the possible distinction between low tones and toneless vowels was ignored. However, the method discussed here suits also to languages with more complex tone patterns.

References

Beesley K. and Karttunen L. 2003. *Finite State Morphology*. Series: Studies in Computational Linguistics, 3. Stanford: CSLI Publications.

Bick Eckhard, 2006. A Constraint Grammar-Based Parser for Spanish. In: *Proceedings of TIL 2006 - 4th Workshop on Information and Human Language Technology* (Ribeirão Preto, October 27-28, 2006.

Bird Steven, 1999a. When marking tone reduces fluency: an orthography experiment in Cameroon. *Language and Speech* 42, 83-115.

Bird Steven, 1999b. Strategies for representing tone in African writing systems. *Written Language and Literacy* 2, 1-44.

Kimenyi, A. 2002. *A Tonal Grammar of Kinyarwanda: An Autosegmental and Metrical Analysis.* The Edwin Mellen Press.

Tapanainen Pasi, 1996. *The Constraint Grammar Parser CG-2*. Publications, 27. University of Helsinki: Department of General Linguistics.