Machine Translation (2003) 18:191–212
DOI 10.1007/s10590-004-2477-4

Finite-State Computational Morphology: An Analyzer Prototype For Zulu

LAURETTE PRETORIUS[1] and SONJA E. BOSCH[2]
[1]*School of Computing;* [2]*Department of African Languages, University of South Africa, PO Box 392, UNISA 0003, Pretoria, South Africa, E-mail:* {*pretol,boschse*}*@unisa.ac.za*

Abstract. As one of the largest of the 11 official languages of South Africa, Zulu is spoken by approximately 9 million people. It forms part of a language family which is characterized by rich agglutinating morphological structures. This paper discusses a prototype of a computational morphological analyzer for Zulu, built by means of the Xerox finite state tools, in particular `lexc` and `xfst`. In addition to considering both the morphotactics and the morphophonological alternation rules that apply, the focus is on implementation and other issues that need to be resolved in order to produce a useful software artefact for automated morphological analysis. The current status of the prototype is alluded to by providing morphological scope, that is the various word categories (parts of speech) that may be handled, and the lexical coverage in terms of the number of different Zulu roots that are included in the embedded lexicon of the analyzer. Preliminary testing and validation procedures are briefly discussed.

Key words: agglutinating morphological structures, analyzer prototype, finite state morphology, Xerox finite state tools, Zulu

1. Introduction

The African continent is known for its cultural and linguistic diversity. In the area extending from the southern point of the African continent to just north of the Equator alone, over 400 languages belonging to the Bantu language family are spoken. Zulu is classified under the South-Eastern zone of the Bantu language family and, as one of the 11 official languages of South Africa, is spoken by approximately 9 million mother-tongue speakers. In terms of natural language processing (NLP), particularly computational morphology, the Bantu languages, including Zulu, certainly belong to the lesser-studied languages of the world. The only Bantu language for which a computational morphological analyzer has been fully developed so far, is Swahili (Hurskainen, 1992).

Two essential components of most NLP applications are a computational morphological analyzer/generator as an enabling technology, and

some form of a machine-readable lexicon as a basic resource. This is particularly true for a language such as Zulu with its complex morphological structure. However, a complete computational morphological analyzer/generator for Zulu does not yet exist, and a machine-readable lexicon for Zulu is not readily available. In this paper, the focus is on the development of a computational morphological analyzer for Zulu, but the role that such a software artefact or tool can play in addressing the latter issue is also briefly mentioned.

In Section 2, certain aspects of the morphological structure of Zulu nouns and verbs are given, while general comments regarding computational morphology and its challenges in the Zulu context are discussed in Section 3. Section 4 is devoted to issues of implementation, such as the finite state approach followed, designing for accuracy and correctness and decisions regarding the analyzer's interface with its environment, its usage and selected computational results. In Section 5, the scope and coverage of the prototype are addressed. The word categories and certain complex linguistic phenomena that are included, as well as others that still need to be addressed, are given. In addition, the representative nature of the selected embedded lexicon used for development, is discussed. Section 6 concerns appropriate preliminary testing, validation and maintenance issues. In particular, the use of available word lists, the role of Zulu natural language corpora, particularly in the absence of a comprehensive machine-readable lexicon for Zulu, as well as the concurrent development of a machine-readable lexicon for Zulu, are briefly addressed.

2. Aspects of Zulu Morphology

In this section, selected aspects of Zulu morphology, which are of significance for the discussion on implementation issues later on, are explained. The emphasis is on the two basic morphological systems which characterize the morphological structure of Zulu, namely the noun classification system, and the ensuing system of concordial agreement.

The noun classification system categorizes nouns into a number of noun classes, as determined by prefixal morphemes also known as noun prefixes. These noun prefixes have, for ease of analysis, been divided into classes identified by numbers by scholars who have worked within the field of the Bantu language family. In Table I, a representation of Meinhof's numbering system of the noun class prefixes (Meinhof, 1932, p. 48) is given.

In general, noun prefixes indicate number, with the uneven class numbers designating singular and the corresponding even class numbers designating plural. However, this is not always the case, since some nouns in so-called plural classes do not have a singular form, some nouns such as

Table I. Numbering system of noun class prefixes

Prefix	Class	Plural	Plural class	Example
umu-	1	*aba-*	2	*umuntu / abantu* 'person / persons'
u-	1a	*o-*	2a	*udokotela / odokotela* 'doctor / doctors'
umu-	3	*imi-*	4	*umuthi / imithi* 'tree / trees'
i(li)-	5	*ama-*	6	*ikati / amakati* 'cat / cats'
isi-	7	*izi-*	8	*isitsha / izitsha* 'dish / dishes'
in-	9	*izin-*	10	*inja / izinja* 'dog / dogs'
i-	9a	*ama-*	6	*ibhasi / amabhasi* 'bus / buses'
u(lu)-	11	*izin-*	10	*uthi / izinti* 'stick / sticks'
u(bu)-	14			*ubuntu* 'humanity'
uku-	15			*ukuzwa* 'to hear / feel'

adoptives in class 9a take their plural in class 6, class 11 nouns take their plurals in class 10, while classes such as 14 and 15 are not associated with number. Although classes 12 and 13 do not occur in Zulu, they do occur in some of the other languages in the Bantu language family.

The noun prefix typically constitutes two parts, namely a preprefix (the initial vowel) and a basic prefix. This division is significant for the analysis in the sense that some classes such as 1a and its plural class 2a do not have a basic prefix at all. In other instances such as classes 5, 11 and 14, the basic prefixes are often discarded, with the result that only the preprefix appears in the surface form. The embedded basic prefix nevertheless needs to be recognized by a morphological analyzer since the whole concordial system of the grammar is based on the noun prefixes.

The significance of noun prefixes is not limited to the fact that they indicate the classes to which the different nouns belong. In fact, noun prefixes play a further important role in the morphological structure of Zulu in that they link the noun to other words in the sentence. This linking is manifested by a concordial morpheme, which is derived from the noun prefix and usually bears a close resemblance to the noun prefix. This system of concordial agreement, which is the pivotal constituent of the whole sentence structure of the Zulu language, governs grammatical correlation in verbs, adjectives, possessives, pronouns and so forth, as illustrated by the bold printed morphemes in example (1).

(1) *Umama u**zi**thenge **zo**nke **izin**gubo **za**mi **ezin**hle*
MOTHER SHE-THEM-BOUGHT ALL CLOTHES OF-ME WHICH-ARE-BEAUTIFUL
'Mother bought all my beautiful clothes.'

In this sentence, the class 1a noun *umama* (***u****-mama*) 'mother' governs the subject concord *u-* in the verb *uzithenge* 'she bought them', while the class 10 noun *izingubo* (***i-zin****-ngubo*) 'clothes' determines concordial agreement of the object concord *-zi-* in *uzithenge*, of the quantitative pronoun *-o-* in *zonke* 'all', of the possessive *za-* in *zami* 'my', and of the adjective *ezin-* in *ezinhle* 'beautiful'.

The predominantly agglutinating nature of the Zulu language is clearly illustrated in the above examples, which all consist of more than one morpheme. Each Zulu example represents a linguistic word, which consists of a number of bound parts or morphemes which cannot occur independently as separate words. This complex morphological structure will be discussed very briefly by referring to two of the most complex word types, namely nouns and verbs.

Nouns as well as verbs in Zulu are constructed by means of the two generally recognized types of morphemes namely roots and affixes consisting of prefixes and suffixes. The majority of roots are bound morphemes since they do not constitute words by themselves, but require one or more affixes to complete the word. The root is generally regarded to be "the core element of a word, the part which carries the basic meaning of a word" (Poulos and Msimang, 1998, p. 170). For instance in (1) *-ngubo-* is the root that conveys the semantic significance of the word *izingubo* 'clothes'. The morphemes *i-* and *-zin-* are prefixes of the root *-ngubo*.

A morphological distinction may be made between two types of nouns:

- Nouns formed from roots that do not require suffixes. Such roots are not derived from other word categories, and cannot be reduced to any simpler form. The noun root *-ntu* which occurs in the noun *umuntu* 'person', is a complete root.
- Nouns formed from roots that do require suffixes. Such roots are derived from other word categories, such as verb roots, adjective stems and ideophones. The verb root *-theng-* occurs in the deverbative noun *umthengi* 'buyer' and needs the suffix *-i* for completeness. Nouns may also be derived from verb roots with extensions, also known as extended roots. In the example *umthengisi* 'seller' the extended root *-thengis-* 'sell' incorporates the causative extension *-is-* and also needs the suffix *-i* for completeness.

In the formation of nouns it should be noted that roots, which do not require any suffixes for completeness, as well as roots to which final suffixes have been added, may both be termed "noun stems" (Poulos and Msimang, 1998, p. 170).

In the case of the verb, the core element which expresses the basic meaning of the word is the verb root. The essential morphemes of a Zulu

verb are a subject concord (except in the imperative and infinitive), a verb root and a terminative, as illustrated by *bafunda* 'they learn' (2).

(2) *ba* ***fund*** *a*
subject-concord root terminative

Over and above the subject concord, the form of which is determined by the class of the subject noun, a number of other morphemes may be prefixed to a verb root, as in *bazolifunda* 'they will learn it' (3).

(3) *ba* *zo* *li* ***fund*** *a*
subjconc future-tense object-concord root termin

It should be noted that whereas object concords also show concordial agreement with the class of the object noun, all other verbal affixes are class independent. Furthermore verbal affixes have a fixed order in the construction of verbs, with the object concord prefixed directly to the verb root.

Suffixes such as verbal extensions may be inserted between the verb root and the terminative. In the example of *abafundisi* 'they do not learn' (4), it will be noted that the terminative has changed to the negative *i* in accordance with the negative prefix *a*.

(4) *a* *ba* ***fund*** *is* *i*
negative subjconc root extension terminitive-negative

Against the background of this brief outline of Zulu morphology, certain issues of computational morphology and the challenges involved in building a computational morphological analyzer for Zulu will be addressed in the next section.

3. Computational Morphology

Computational morphology deals with automatic word-form recognition and generation, that is, the study of the computational analysis and synthesis of word forms, a prerequisite of all other rule-based techniques of automatic language analysis. The general challenges posed by a computational morphological analyzer are twofold. First, morphemes (i.e. roots and affixes) that make up words cannot combine at random, but are restricted to certain combinations and orders. A morphological analyzer needs to know which combinations of morphemes (morphotactics) are

valid. Second, in the construction of words, morphemes may be realised in different ways depending on their context. A morphological analyzer needs to recognize the morphophonological changes between lexical and surface forms (morphophonological alternations).

Some of the implications of morphotactics and morphophonological alternations in Zulu are illustrated below.

There are numerous possibilities of morpheme concatenations in Zulu word categories for which possible valid combinations need to be determined. In the case of nouns, stems are preceded by a noun prefix which is subdivided into a preprefix and a basic prefix, and may be followed by nominal suffixes such as the diminutive, feminine and augmentative, or even deverbative suffixes, should the noun be formed from a verb root, as in *indlovukazi* 'elephant cow' and *umufundisi* 'teacher' (5).

(5) a. *i* *n* ***dlovu*** *kazi*
preprefix basic-prefix root feminine-suffix
b. *u* *mu* ***fund*** *isi*
preprefix basic-prefix root deverbative-suffix

Morpheme combinations and legal orders that need to be recognized for the verb word category are even more complex since the verb root may be preceded by a variety of prefixes, while various extensions and a terminative may be suffixed to the verb, e.g. *azizubaphekela* 'they will not cook for them' (6).

(6) *a* *zi* *zu* *ba* ***phek*** *el* *a*
neg subjconc future obj-conc root extension terminative

Words are just concatenations of morphemes, but the mere concatenation of morphemes in Zulu would generally result in abstract "morphophonemic" words. Alternations occur between the raw concatenations on the abstract level and the correct final words on the surface level. These are best illustrated by the examples in (7) which show the abstract level, surface level and gloss.

(7) a. `nga + u + mu + ntu` *ngomuntu* 'with the person'
b. `a + i + bamb + w + a` *ayibanjwa* 'it is not being caught'
c. `ba + ya + yi + os + a` *bayayosa* 'they are roasting it'

A morphological analyzer needs to take into account morphophonological processes where changes in the sounds of morphemes are based on surrounding phonemes. For instance, in (7a) we find vowel coalescence ($a + u \rightarrow o$), in (7b) we have an example of consonantalization ($a + i \rightarrow ayi$) as

well as palatalization ($mb + w \rightarrow njw$), while (7c) illustrates vowel elision ($i + o \rightarrow o$).

4. Implementation

4.1. GENERAL APPROACH

Computationally modeling and implementing Zulu morphology requires the representation of the morphophonological phenomena of Zulu, as they occur in the natural language, as closely as possible. Any regular linguistic behavior should be exploited, without imposing rules or structure where none seem to be justified. For example, when a Zulu word allows multiple correct analyses, this vagueness or apparent ambiguity in the natural language should be allowed to persist since such (required) disambiguation can often only be performed when the context in which a word occurs is taken into account. This falls outside the scope of morphological analysis and should be dealt separately. In cases where different schools of linguistic thought or linguistic vagueness regarding precise analysis exist, linguistically justified choices are made for the sake of implementation and computation. Such choices are clearly documented for future reference and possible modification.

A finite state computational approach is employed. The increased application of finite state methods in all aspects of NLP is evident from the proliferation of finite state tools available for building and manipulating large-scale finite state natural language systems, see for example (Silberztein, 1999; Karttunen and Oflazer, 2000; Van Noord, 2002; Sproat, 2003). The Zulu analyzer prototype under discussion is being developed with the Xerox finite state toolkit.

4.2. DESIGNING FOR ACCURACY AND CORRECTNESS

This section primarily concerns the quality of the final finite state transducer as an independent software artefact, in other words, the accuracy with which it models the morphological structure of *all and only* the words in Zulu. A distinction is made between the stable part of the analyzer, namely the morphological structure (morphotactics and alternation rules) and the closed word categories on the one hand, and the dynamic part, as represented by the open word categories that may evolve with time and will require continued enrichment and maintenance as the natural language develops and changes, on the other. Four design features are highlighted.

4.2.1. *Accurate Modeling of the Morphological Processes*

The Xerox software tool for modeling the morphotactics is `lexc`. An accurate specification of the Zulu word structure, i.e. all and only word roots in the language, all and only the affixes for all the word categories, as well as a complete description of the valid combinations of these morphemes for forming all and only the words of Zulu, is created as a `lexc` script file, which is then compiled into a finite state network. The words generated by this network are morphotactically well-formed, but still rather abstract lexical or morphophonemic words, e.g. (8).

(8) *ba-**bamb**-w-a*
SC2-VRoot-PassExt-VerbTerm
'they are being caught'

The morphophonological (phonological and orthographical) alternations are modeled with `xfst`. The changes (orthographic/spelling) that take place between the lexical and surface words when morphemes are combined to form new words/word forms, are described by means of the Xerox regular expression language. The alternation rules are formulated in terms of the `xfst` regular expression language and appropriately composed into a single regular expression. The regular expression is then compiled into a finite state network by means of `xfst`, as in $mb + w \rightarrow njw$ to handle *babanjwa* (8). In `xfst` this becomes the rule (9),

(9) `m b -> n j || _ W`

read as "*mb* maps to *nj* in the right context of *w*".

By combining (by means of composition) the two mentioned finite state networks, a single network – a "lexical transducer" – is formed. Since such finite state transducers are inherently bidirectional, this lexical transducer constitutes the morphological analyzer *and* generator, and it embodies all and only the morphological information about the language being analyzed or generated, including derivation, inflection, alternation and compounding.

4.2.2. *Exhaustive Listing*

By "exhaustive listing" is meant the explicit inclusion of surface words in the embedded lexicon of the analyzer (via the `lexc` script). This approach is practical and convenient in the case of closed word categories and for words that exhibit irregular and exceptional morphological behavior since such items are usually limited in number. Examples of such closed word categories are the absolute and quantitative pronouns, as well as the demonstrative.

Examples of irregular or exceptional forms include the variant form *abe-* of the class 2 prefix. This form is usually linked to noun stems with a specific semantic content, namely the names of tribal or ethnic groups, and is not determined morphophonologically, e.g. *abeSuthu* 'Sotho people', *abeTswana* 'Tswana people'. Therefore a separate entry, `CL1-2abe` is provided for in `lexc`, in addition to the standard entry `CL1-2` for the regular forms. This is in contrast to *ab-*, the other variant form of the class 2 prefix, which occurs when prefixed to vowel-commencing stems, e.g. *ab-**akhi*** 'builders'. Since the variant form *ab-* is morphophonologically conditioned, it is modeled by means of a rule (10).

(10) `^BA -> b | | _ Vowel`

4.2.3. *Intermediate Stem Analysis to Prevent Spurious Over-Generation*

Conceptually, the root is the constant core element in the formation of Zulu words, and the rest is inflection and derivation. This implementation approach consists of the inclusion of only noun roots and verb roots as baseforms in the `lexc` script, with the rest following from the morphotactics and the alternation rules. There is, however, also another approach, suggested by the Zulu linguistic process of forming nouns from verb roots. For purposes of implementation, this process may be viewed as having two stages. In the first stage, the verb root is transformed into a so-called noun stem by adding certain suffixes to the verb root. Not all such possible suffixes combine with all verb roots. Once these noun stems have been formed, the second stage consists of the same noun formation processes that apply to noun roots. An example is the verb root *-fund-* 'learn/read' and its associated noun stems as shown in Table II.

Therefore, instead of listing only the verb root as a baseform as in the first approach, the noun stems associated with each verb root are also explicitly included[1] in the embedded lexicon (in the `lexc` script). This approach limits the possible formation of spurious verb root–deverbative suffix(es) combinations and thereby simplifies implementation. In terms of implementation, this requires two stages of analysis, illustrated by means of the following example. The first stage of the analysis of the surface form *umfundisi* 'teacher' proceeds as in (11).

(11) *u* *mu* ***fundisi***
preprefix basic-prefix **noun-stem**

The second stage of the analysis then maps the noun stem portion resulting from the first stage of analysis to the verb root and its associated suffix(es) (12).

Table II. Noun stems derived from the verb root *fund* 'learn/read'

Stem	Meaning	Class
-funda (uku-)	'learn', 'read'	15
-funda (im-/izim-)	'one beginning to learn'	9–10
-fundelo (im-/izim-)	'special training'	9–10
-fundi (um-/aba-)	'pupil', 'scholar', 'disciple'	1–2
-fundi (ubu-)	'discipleship'	14
-fundisano (im-/izim-)	'influence on one another'	9–10
-fundisi (ubu-)	'teaching profession'	14
-fundisi (um-/aba-)	'teacher'	1-2
-fundiso (im-/izim-)	'teaching', 'doctrine'	9–10
-fundiswa (isi-/izi-)	'learned person'	7–8
-fundo (im-)	'education', 'learning'	9
-fundo (isi-/izi-)	'lesson', 'lecture'	7–8.

(12) ***-fund*** *is* *i*
verb-root verbal-extension deverbative-suffix

The implementation of the second stage of analysis is done by means of a number of `xfst` rules that essentially strip off any legal suffix(es) and do not require access to the actual Zulu verb root list.

In summary, by explicitly listing the noun stems of the verb root *-fund-*, no suffixes other than *-a*, *-el-o*, *-i*, *-is-an-o*, *-is-i*, *-is-o*, *-is-wa* and *-o* will occur with *-fund-*. This process therefore prevents possible overgeneration of verbal extension combinations with the verb root.

4.2.4. *Modeling Separated Dependencies*

Zulu morphology is, among others, characterized by the occurrence of separated morphological dependencies within (linguistic) words. Any computational model and implementation of Zulu morphology should therefore be able to take such dependencies into account in an accurate and efficient way.

A particularly useful device, offered by the Xerox finite state toolkit as an extension of its standard finite state functionality, is the "flag diacritics". Flag diacritics provide means of feature setting and feature unification that keep transducers small, enforce desirable results, and simplify grammars. In particular, they are used to block illegal paths at run time by the

analysis and generation routines. In `lexc` and `xfst`, they are treated as multi-character symbols, that is, special members of the alphabet and are spelt according to the two templates `@operator.feature@` and `@operator.feature.value@`. Typical operators are Unification, Positive setting, and Require test. For a detailed discussion, see Beesley and Karttunen (2003, p. 339).

Another versatile way of keeping track of these dependencies is by making use of the freedom (available to the implementer) to design and customize the alphabet of the so-called "intermediate language". Special members of the intermediate language alphabet may be introduced as multicharacter symbols. In the Xerox approach, this language conceptually forms the lower language in the transducer that originates from the `lexc` script and the upper language in the transducer that is built by means of `xfst`. This language disappears when the two transducers are composed and is completely invisible to the user.

The implementation of certain kinds of separated dependencies occurring in Zulu is illustrated by means of examples.

4.2.4.1. *Separated Dependencies Arising in the Formation of Nouns*

1. All noun prefix–noun stem combinations in Zulu are governed by the nominal classification system, as discussed in Section 2. This is implemented by attaching to every noun stem entry in the `lexc` script a marker in the form of a flag diacritic to reflect the requirements of the nominal classification system. For instance, the noun root *-lomo* 'mouth' is flagged to indicate that it takes noun prefixes from classes 3, 4 and 14, yielding entries such as those shown in (13).

   (13)
   ```
   lomo@U.CL.3-4@
   lomo@U.CL.14@
   ```

   Noun preprefix–basic prefix pairs independently appear in the `lexc` script as in (14).

   (14)
   ```
   u...mu...@P.CL.1-2@...
   u...mu...@P.CL.3-4@...
   i...li...@P.CL.5-6@...
        ⋮
   u...bu...@P.CL.14@...
   u...ku...@P.CL.15@...
   ```

   The positive setting and associated unification capability offered by the flag diacritics above will at run time guarantee that only the correct combinations are allowed. For example, an incorrect analysis containing

`u[NPrepre1]mu[BPre1]lomo[WRoot]` will be prevented by the setting of the `CL` feature (for class) to `1-2` when processing *umu-* and the subsequent failure to unify this setting with a `CL` feature value of `3-4` when processing *-lomo*.

2. Two types of roots are distinguished in the formation of noun stems; one is a noun root (e.g. *-lomo* 'mouth') and the other a verb root (e.g. *fund* 'learn'). All noun stems (whether they have as root a noun root or a verb root) require noun prefixes. However, differentiated treatment is necessary since a noun root does not require any suffixes for completeness, while a verb root requires certain suffixes to form a complete noun stem. Both types of roots take suffixes such as the diminutive. We may also handle this separated dependency by means of flag diacritics. If a noun prefix is prefixed to a noun stem derived from a verb root, then a nominal suffix is eventually required to complete the noun. Only verb roots that have been preceded by noun prefixes (flagged by `@P.NomSuf.ON@`) may take nominal suffixes such as *-i*, *-a* and *-o*. This dependency is enforced by means of `@R.NomSuf.ON@`.
3. The division of the noun prefix into a preprefix and a basic prefix is also important for another reason. In some instances of classes 5, 11 and 14, the basic prefix is often deleted, with the result that only the preprefix appears in the surface form, while both prefixes should be present in the analysis. This phenomenon is implemented by means of special symbols added to the intermediate language. Both the presence of the preprefix and the basic prefix is reflected by their occurrence as special symbols in the intermediate language.
   For example, in the `lexc` script, `u-bu` (class 14) is mapped to `^U^BU` in the intermediate language, and then an `xfst` rule such as (15),

   (15) `^BU -> 0 || L _ R`

   where `L` and `R` are appropriate *surface language* contexts, removes the basic prefix when necessary.
   This approach is also useful for the formation of vocatives, in which case the preprefix falls away in most classes, e.g. (16).

   (16) *abafana* > *bafana!* 'boys'
   ugogo > *gogo!* 'grandmother'

4.2.4.2. *Separated Dependencies Arising in the Formation of Verbs*

1. A positive verb in the present tense commences with a subject concord which may be followed by the long present tense morpheme under certain syntactic conditions. Morphologically, however, the long present tense morpheme does not occur in the negative, e.g. (17).

(17)
positive:	*ba-**ya**-hamb-a*	'they are going'
negative:	***a**-ba-hamb-i*	'they are not going'

2. A negative verb in the present tense commences with a negative prefix, followed by a subject concord, etc. A separated dependency needs to be marked between the negative prefix and negative verb terminative, e.g. (18).

(18)
positive:	*si-khulum-a*	'she talks'
negative:	***a**-si-khulum-**i***	'she does not talk'

3. In the case of a negative verb in the future tense, a separated dependency exists between the negative prefix and the (negative) future tense morpheme. In addition, the verb terminative remains an *-a* in this case, e.g. (19).

(19)
positive:	*ba-zoku-hamb-a*	'they will go'
negative:	***a**-ba-**zuku**-hamb-**a***	'they will not go'

4. Intransitive verb roots need to be marked as incompatible with object concords in an example such as *-khukhuka* 'get swept away'.

Regarding the implementation of these separated dependencies, only example (19) will be explained since the other examples are handled similarly.

Modeling the condition in example (19) requires that the occurrence of the negative prefix *a-* be flagged by setting the feature `NEG` to the value `ON`. On subsequently encountering the future tense morpheme the latter should be checked for negativity by insisting that the feature `NEG` has previously been set to the value `ON` *and* it should be flagged as a future tense morpheme. Finally, the correct verb terminative *-a* is ensured by checking that the `NEG` feature had not been set to `OFF` and the feature `FT` (future tense) is `ON`.

This is realized in the `lexc` script as the succession of (partial) entries (20).

(20)
```
a@P.NEG.ON@...
      ⋮
zuku@R.NEG.ON@@P.FT.ON@...
      ⋮
a@U.NEG.ON@@R.FT.ON@...
```

4.3. INTERFACE AND USAGE

In addition to the inherent computational functionality of the lexical transducer, a key design and implementation issue is its interface with its environment, such as other software artefacts or human users. Since the transducer is bidirectional and its functionality concerns the morphological analysis of given Zulu surface forms in the one direction and the generation of Zulu surface forms, given the appropriate analyses, lemmas or lexical forms in the other, this interface may, at a rudimentary and conceptual level, be thought of as the representation of the Zulu surface forms and their associated lexical forms.

4.3.1. *Specification of the Lexical Forms*

Regarding lexical forms, the content and format of the morphological feature information that constitute these forms requires specification, in particular, the morphological granularity and the feature information to be provided, the choice of an appropriate tag set and the positioning of the tags in relation to the morphemes they are associated with.

While standardization and rigid rules regarding these aspects are impractical due to the large differences between natural languages, guidelines and some degree of uniformity among similar or related languages (McEnery et al., 1998; Allwood et al., 2003; Beesley and Karttunen, 2003; Erjavec, 2004) is desirable in terms of, for instance maintenance and re-usability.

4.3.1.1. *The Granularity and Feature Information.* In Sections 2 and 3, a tacit exposition of the morphological granularity and the appropriate linguistic feature information customary in Zulu morphology was presented. It was shown that the morphological granularity goes beyond the levels of parts of speech and syntagmatic morphological categories by including paradigmatic distinctions within such categories, where necessary. Typical feature information for Zulu is given in Table III.

4.3.1.2. *The Morphological Tag-Set.* In choosing tags, existing guidelines and *de facto* standards and linguistic conventions are used for features that are common to most languages, where possible. In cases where clear, widely accepted guidelines do not yet exist, as is the case for the indigenous languages of southern Africa, tags were devised that consist of intuitive mnemonic character strings that abbreviate the features they are associated with. A fragment of the tag set used in the Zulu analyzer prototype under discussion is shown in Table III.

Table III. Features and tags

Feature	Tag
Noun stem	`[NStem]`
Verb root	`[VRoot]`
Pronoun stem first person singular	`[PronStem1ps]`
Adverbial formative	`[AdvForm]`
Noun preprefix class 1	`[NPrePre1]`
Basic prefix class 1	`[BPre1]`
Copulative prefix	`[CopPre]`
Locative prefix	`[LocPre]`
Negative prefix	`[NegPre]`
Future negative prefix	`[FutNeg]`
Subject concord class 15	`[SC15]`
Negative subject concord class 1a	`[NegSC1a]`
Object concord class 15	`[OC15]`
Possessive concord class 8	`[PossConc8]`
Diminutive suffix	`[DimSuf]`
Locative suffix	`[LocSuf]`
Causative extension	`[CausExt]`
Passive extension	`[PassExt]`
Verb terminative negative	`[VerbTermNeg]`

4.3.1.3. *The Format.* The choice of representation of the morphological analyses in the analyzer under discussion includes the root, as well as the lexical forms of the affixes, and each morpheme is followed by its tag(s) (see also Allwood et al., 2003, Beesley and Karttunen, 2003). For example, the word *umfundisi* 'teacher' is lemmatized as (21) (cf. (11–12) above).

(21) `u[NPrePre1]mu[BPre1]fund[VRoot]is[CausExt]i[NomSuf]`

Another often used form of representation includes only the baseform with tags punctuated by the symbol '+' (Oflazer and Inkelas, 2003; Talmon and Wintner, 2003; Úi Dhonnchadha, 2003). According to this representation *umfundisi* may be analyzed as (22).

(22) `NPrePre1+BPre1+fund+CausExt+NomSuf`

Yet another form of representation may be found in Karttunen (2003).

However, in the realm of the Xerox finite state calculus, the differences between these representations are cosmetic and conversion from the one to the other via replace rules is straightforward.

4.3.2. *Selected Computational Results*

In the computational results in (23–31), the first line denotes the surface word, and the subsequent lines the analyses (lemmas or lexical forms). The glosses are added for clarity.

(23) *ngumlomo* 'it is the mouth'
```
ngu[CopPre]u[NPrePre3]mu[BPre3]lomo[NStem]
```
(24) *incwajana* 'small letter'
```
i[NPrePre9]n[BPre9]ncwadi[NStem]ana[DimSuf]
```
(25) *emotweni* 'in/at the motor car'
```
e[LocPre]i[NPrePre9]n[BPre9]moto[NStem]ini[LocSuf]
```
(26) *ngotshani* 'with grass'
```
nga[AdvForm]u[NPrePre14]bu[BPre14]tshani[NStem]
```
(27) *akahambi* 'he/she/it does not walk'
```
a[NegPre]ka[NegSC1a]hamb[VRoot]i[VerbTermNeg]
a[NegPre]ka[NegSC1]hamb[VRoot]i[VerbTermNeg]
a[NegPre]ka[NegSC6]hamb[VRoot]i[VerbTermNeg]
```
(28) *yimi* 'it is I'
```
yi[CopPre]mi[PronStem1ps]
```
(29) *zikababa* 'father's'
```
zika[PossConc8]u[NPrePre1a]baba[NStem]
zika[PossConc10]u[NPrePre1a]baba[NStem]
```
(30) *kukhulunywa* 'it is being spoken'
```
ku[SC15]khulum[VRoot]w[PassExt]a[VerbTerm]
```
(31) *abazukuthengisa* 'they (will not) sell it'
```
a[NegPre]ba[SC2]zuku[FutNeg]theng[VRoot]is[CausExt]
     a[VerbTerm]
a[NegPre]ba[SC2]zu[FutNeg]ku[OC15]theng[VRoot]
     is[CausExt]a[VerbTerm]
```

The multiple analyses in some of the above examples may be ascribed to the fact that certain morphemes are identical in a number of classes. For instance the negative subject concord *-ka-* has the same form in classes 1a, 1 and 6, while the possessive concord *-zika-* is identical in classes 8 and 10. In the last example *-zuku-* may feature either as a negative future tense morpheme, or it may feature as two morphemes with *-zu-* as a variant negative future tense morpheme, and *-ku-* as a class 15 object concord.

Morphologically all the analyses are correct, but disambiguation needs to be done at a syntactic level, taking context into consideration.

5. Scope and Coverage

A prototype may be defined as a working model that is functionally equivalent to a subset of the ultimate software product.[2] So, when presenting a prototype of any kind, one of the first questions that comes to mind is: In what sense is this prototype not yet functionally complete, or, in what sense is it still a subset of the final product? This question regarding the Zulu analyzer prototype is addressed by considering two aspects, previously alluded to in Section 4.2, namely its *scope* in terms of word categories and their morphological structure included, and its *lexical coverage* as reflected by the number of different noun stems and verb roots present in its embedded lexicon.

5.1. WORD CATEGORIES

Table IV indicates which of the morphemes (roots and affixes) have already been included in the prototype. A number of complex linguistic phenomena have already been modeled, for example the implementation of certain separated dependencies as discussed in Section 4.2.4, while the compound tenses of Zulu are at present being included.

5.2. THE EMBEDDED LEXICON

Table IV indicates which of the morphemes (roots and affixes) have already been included in the prototype. For the open categories, a selected collection of entries was used as the initial embedded lexicon of the analyzer. For the purposes of systematic testing, this collection consisted of a linguistically representative set of noun stems and verb roots. This set was compiled to contain all the salient features necessary for development.

In Table V, it is shown to what extent the number of noun stems in the embedded lexicon was expanded. The subsequent systematic expansion of the embedded lexicon to include all and only the valid stems/roots of Zulu is discussed in the next section.

6. Testing, Validation and Maintenance

6.1. TESTING AND VALIDATION

One of the attractive characteristics of the Xerox finite state toolkit is the automated testing and debugging functionality that it offers to developers

Table IV. Word categories and morphological structures in the prototype

Word category	Morphemes	Entries
Noun	Noun stems	See Table V
	Noun prefixes	All classes
	Noun suffixes	4
Pronoun		
Absolute	Pronoun stems	All classes
	Pronoun suffixes	All
Quantitative	Quantitative concords	All classes
	Quantitative stems	All
Demonstrative		All
Qualificative	Adjective stems	65
	Relative stems	400
	Possessive concords	All classes
	Possessive stems	All
	Enumerative concords	All classes
	Enumerative stems	All
Verb	Roots	7400
	Negative prefixes	6
	Subject concords (present tense)	All classes
	Subject concords (past tense)	All classes
	Object concords	All classes
	Future tense prefixes	All
	Verb extensions	4
	Verb terminatives	7
Copulative	Copulative prefixes	All
Adverb	Adverbial prefixes	All
	Locative prefixes	All
	Underived adverbs	10
Ideophone		950
Conjunction		12
Interrogative		10

(Beesley and Karttunen, 2003, p. 311). These were employed throughout the design and development of the prototype to ensure that both the analytic as well as the generative capabilities of the analyzer were systematically and incrementally tested and validated.

Table V. Noun stem entries

Class	In development lexicon	From word list
1–2abe	3	20
1a–2a	5	1500
1–2	6	500
3–4	11	2200
5–6	19	2400
6	1	300
7–8	14	2800
9–6	1	160
9	1	400
9–10	18	2400
11–10	13	800
11	1	500
14	9	600
Total		13000

The design and implementation of the analyzer prototype, as discussed in Sections 4 and 5, were based on a selected representative lexicon. The next phase of testing and validation consisted of extending its lexical coverage and testing it on an electronically available Zulu word list of 26,000 (correct) words. This resulted in the addition of approximately 13,000 noun stems, as summarized in Table V, as well as 7400 verb roots, 65 adjective stems, 400 relative stems and 950 ideophones to the prototype. Due to the agglutinative nature of Zulu morphology, the number of words that can be handled by this prototype is, of course, orders of magnitude greater. The current prototype has been applied to a raw sample corpus consisting of 13,000 tokens and resulted in the analysis of 69% of the tokens.

Ultimately all and any suitable data available, including electronically accessible corpora will be used for testing and validation purposes.

6.2. MAINTENANCE

Since a limitation in the NLP of Zulu is the fact that a machine-readable Zulu lexicon is not readily available in any form, a so-called "guesser" (Beesley and Karttunen, 2003) variant of the morphological analyzer/generator is being used concurrently to extend/enhance our morphological analyzer and build an XML Zulu lexicon.

This guesser is based on the current analyzer prototype and has been designed to recognize all phonologically possible word roots/stems in Zulu. This option enables the analyzer to identify "new" roots/stems which are not yet embedded in the lexicon, either because they are newly coined or because they have simply been overlooked. If for instance, the morphological analyzer does not include the noun stem *-zi* in its embedded lexicon, and it comes across the word *imizi* 'village' in a corpus, applying the guesser to such a corpus, yields (32).

(32) `i[NPrePre5]li[BPre5]mizi+Guess[NStem]`
`i[NPrePre4]mi[Bpre4]zi+Guess[NStem]`

On scrutinizing the various possibilities, the Zulu linguist identifies *-zi* as a new stem. While *-mizi* may be a phonologically possible stem, it is not a real stem in the Zulu language. The lexicographer builds the lemma (singular and plural) (32), which then leads to an entry in the XML lexicon.

(33) `-zi (umuzi ... imizi) n 1. village, homestead ...`

The guesser is a very useful feature for identifying and extracting new words which are being created daily in a developing language such as Zulu. By means of the guesser, a significant number of new roots/stems for inclusion in the analyzer may be extracted. These new roots/stems will lead to new entries in the evolving XML lexicon as well. The (semi-automated) enhancement cycle is completed by the inclusion of these new roots/stems in the morphological analyzer and by the subsequent recompilation thereof.

7. Future Work and Conclusion

When all the word categories have been included, the analyzer will be made available online for experimentation and use by any interested party in order to obtain constructive feedback and suggestions regarding errors, omissions, extensions and modifications. In parallel to this initiative, new language corpora will be systematically and regularly explored in order to ensure an up-to-date morphological analyzer and XML lexicon.

Indeed, the online availability of the Zulu analyzer and its use by interested parties, as well as new Zulu corpora, are regarded as key components in ensuring that the Zulu analyzer prototype will be increasingly able to analyze and generate all and only the valid words of Zulu.

Acknowledgements

The authors would like to acknowledge the continued support and expert advice of Dr Ken Beesley of Xerox Research Centre Europe (XRCE). This material is based upon work supported by the National Research Foundation of South Africa under grant number 2053403. Any opinion, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Research Foundation.

Notes

[1] This is also standard practice in traditional paper dictionaries.

[2] A software engineering perspective: See for example Schach (2002), p. 70.

References

Allwood, Jens, Leif Grönqvist and A. P. Hendrikse: 2003, 'Developing a Tag Set and Tagger for the African Languages of Southern Africa with Special Reference to Xhosa', *Journal of Southern African Linguistics and Applied Language Studies* **21**, 223–237.

Beesley, Ken and Lauri Karttunen: 2003, *Finite-state Morphology*, Stanford, CA: CSLI Publications.

Erjavec, Tomaž (ed.): 2004, The MULTEXT-East Morphosyntactic Specifications Version 3.0, available at `http://nl.ijs.si/ME/V3/msd/msd.pdf`, accessed 14/10/2004.

Hurskainen, A.: 1992, 'A Two-Level Formalism for the Analysis of Bantu Morphology: An Application to Swahili', *Nordic Journal of African Studies* **1**, 87–122.

Karttunen, Lauri: 2003, 'Computing with Realizational Morphology', in Alexander Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing, 4th International Conference, CICLing 2003, Mexico City, Mexico*, Lecture Notes in Computer Science 2588, Berlin; Springer, pp. 203–214.

Karttunen, L. and K. Oflazer: 2000, 'Introduction to the Special Issue on Finite-State Methods in NLP', *Computational Linguistics* **26**, 1–2.

McEnery, Tony, Lou Burnard, Andrew Wilson and Baker [sic]: 1998, Validation of Morphosyntactic Analyses, available at `http://www.elra.info/services/valid/wp3/MORPH.htm`, accessed on 10/3/2003.

Meinhof, C.: 1932, *Introduction to the Phonology of the Bantu Languages*, Berlin: Dietrich Reimer/Ernst Vohsen.

Oflazer, Kemal and Sharon Inkelas: 2003, 'A Finite-State Pronunciation Lexicon for Turkish', in *Proceedings of the Workshop on Finite-state Methods in Natural Language Processing, 10th Conference of the European chapter of the Association for Computational Linguistics*, Budapest, Hungary, pp. 11–18.

Poulos, G. and C. T. Msimang: 1998, *A Linguistic Analysis of Zulu*, Pretoria: Via Afrika.

Schach, Steven R.: 2002, *Object-oriented and Classical Software Engineering*, 5th ed., Boston: McGraw Hill.

Silberztein, M.: 1999, 'INTEX 4.1 for Windows:a Walkthrough', in J. M. Champarnaud, D. Maurel and D. Ziadi (eds), *Automata Implementation, Third International Workshop on Implementing Automata, WIA'98, Rouen, France*, Lecture Notes in Computer Science 1660, Berlin: Springer, pp. 230–243.

Sproat, Richard: 2003, Lextools: a toolkit for finite-state linguistic analysis, AT&T Labs-Research, available online at `http://www.research.att.com/sw/tools/lextools/`, accessed 3/10/2003.

Talmon, Rafi and Shuly Wintner: 2003, 'Morphological Tagging of the Qur'an', in *Proceedings of the Workshop on Finite-state Methods in Natural Language Processing, 10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary, pp. 67–74.

Úi Dhonnchadha, Elaine: 2003, 'Finite-State Morphology and Irish', in *Proceedings of the Workshop on Finite-state Methods in Natural Language Processing, 10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary, pp. 43–50.

Van Noord, G.: 2002, FSA6.2xx: Finite State Automata Utilities, available online at `http://odur.let.rug.nl/~vannoord/Fsa/fsa.html`, accessed 3/10/2003.