Using the isiZulu GF Resource Grammar for morphological annotation

Laurette Marais
Natural Language Processing
CSIR
Pretoria, South Africa
lmatais@csir.co.za

Laurette Pretorius
Dept of Mathematical Sciences
Stellenbosch University
Stellenbosch, South Africa
lpretorius@sun.ac.za

Abstract

The isiZulu GF Resource Grammar (ZRG) enables syntactic parsing using the GF runtime system. In order to perform this task, the ZRG implicitly encodes rich morphosyntactic information about isiZulu. In this paper we show how such information can be made explicit by adapting the way the grammar linearises GF abstract syntax trees. The result is annotated text, which can be utilised in various ways for supporting natural language processing of an under-resourced, morphologically complex language like isiZulu.

1 Introduction

Large quantities of high quality data is essential for building language technologies (LTs) that are useful. This lack of data is often one of the most limiting factors in developing such technologies for languages such as isiZulu, one of the official Bantu languages of South Africa.

The most basic form of data may be thought of as language texts in digital, machine processable form, drawn from any source or genre and often collected by digitising existing written texts. In an age where LLMs are often perceived as the end goal of LT, the development of LLM models for the Bantu languages remains an open challenge due to the lack of sufficient data. Moreover, the situation is exacerbated by the morphosyntactic complexity of the languages, which increases the size requirements for datasets (Hussen et al., 2025). One way of extracting the most out of available datasets is via annotation.

Broadly speaking, linguistic annotation involves the association of descriptive or analytic notations with language data (Ide and Pustejovsky, 2017, p.2). Aside from providing richer data on which to train LTs, linguistically annotated data also provide a resource for corpus linguistics, a growing field of study within the Bantu languages (Prinsloo and de Schryver, 2001; Taljard and de Schryver, 2016).

There are many types of linguistic annotation. Well-known examples are morphosyntactic tagging (e.g. lemmatisation, part of speech tagging and morphological tagging), syntactic analyses, a range of semantic analyses (e.g. semantic roles, named entities, sentiment and opinion), time and event and spatial analyses, and discourse level analyses including discourse structure, co-reference, etc.

The two essential components of a linguistic annotation project are firstly, the *annotation scheme* that defines the labels or tags, and secondly, an *annotation tool* that supports accurate and fast annotation (Ide and Pustejovsky, 2017, p.3).

In this paper we show how morphological annotation of isiZulu, based on the *ZulMorph tagset*[1], is accurately and efficiently performed by adapting the existing *GF resource grammar for isiZulu* (ZRG) (Marais and Pretorius, 2023) as a tool.

For morphological annotation, two approaches could be considered, namely surface form (morph) annotation and canonical (morpheme) annotation[2]. This distinction has been noted especially with regards to morphological segmentation for isiZulu and the other languages in the Nguni language group (von der Wense et al., 2016; Moeng et al., 2021).

We start with a discussion of isiZulu in Section 2, followed by a brief overview of Grammatical Framework (GF) as an annotation tool in Section 3. In Section 4, we highlight the most relevant aspects of the isiZulu GF resource grammar (ZRG), before discussing our adaptation of it for morphological annotation in Section 5. Finally, in Section 6, we contrast the adapted ZRG with two state-of-the-art tools for isiZulu morphological annotation.

[1] https://portal.sadilar.org/FiniteState/demo/zulmorph

[2] A morpheme is the smallest meaningful unit in the grammar of a language and morph is the phonetic realization of a morpheme.

2 IsiZulu morphosyntactic features

IsiZulu morphosyntax is essentially based on two principles, viz. nominal classification (the system of noun classes) and concordial agreement (the system of concords). IsiZulu has a complex agglutinative morphology (Poulos and Msimang, 1998, p.6). Moreover, isiZulu exhibits a high degree of morphophonological alternation (Poulos and Msimang, 1998, pp.515-534). The language has a conjunctive orthography.

System of noun classes. The noun in isiZulu consists of two main parts, viz. a noun prefix (preprefix and basic prefix) and a noun stem[3]. Furthermore, every noun belongs to a so-called noun class by virtue of the form of its prefix, also referred to as its class gender. This notion of class gender is significant since it generates grammatical agreement by means of these class prefixes. The noun classes are numbered. IsiZulu has 18 noun classes. Generally speaking the nouns occur in singular/plural pairs[4]. Commonly found pairs are 1/2; 1a/2a; 3/4; 5/6; 7/8; 9/10; 11/10; 14/6. Nouns in classes 15, 16, 17 and 18 do not usually have a plural form. The noun classes 16, 17 and 18 are so-called locative classes (Poulos and Msimang, 1998, Chapter 1).

System of concords. A concord is a structural element (agreement marker/morpheme) which formally marks the relationship between a noun and other words in a sentence. This class gender agreement (see above) must be observed in all parts of the utterance which are linked to the noun. Therefore, we say that word categories such as verbs, pronouns, adjectives, relatives, possessives etc. are brought into concordial (i.e. grammatical) agreement by means of these concords (Kosch, 2006, p.90).

The verb. It is the morphologically most complex word category, consisting of multiple morphemes, viz. a root, the morpheme that carries the basic meaning of the verb, and affixes (prefixes and suffixes), morphemes added to the root to give the verb its required functional value, expressing a variety of moods, forms, tenses, aspects and polarity. The prefixes that may occur are, in order, a negative morpheme, subject concord, negative morpheme[5], temporal morpheme, aspectual morpheme, object concord and reflexive morpheme. Possible suffixes include verbal extensions, a verb terminative, a relative suffix and an imperative suffix. Filling these various slots for the affixes depends of the required functional value of the verb, with the root as only obligatory morpheme.

The need for computational morphological analysis as a first step in LT for isiZulu is illustrated by means of the sentence *izalukazi azizukuziphekela imifino* (the women will not cook vegetables for themselves) in Figure 1, analysed by means of ZulMorph (Pretorius and Bosch), a state-of-the art *finite-state* morphological analyser for isiZulu (Pretorius and Bosch, 2010).

The verb *azizukuziphekela* consists of seven morphemes that have to be identified and annotated in order to fully understand its meaning. It is also linked to the subject of the sentence (the class 8 noun) by means of the (class 8) subject concord, *zi*. From our example it is clear that once the sentence has been morphologically annotated as in (1), all its essential morphosyntactic information is known. In particular, it encompasses other kinds of annotation such as part-of-speech tags (e.g. `NOUN` and `VERB` resp.) and lemmatisation (as stems) (`alukazi`, `pheka` and `fino`).

3 Using GF for annotation

A *resource grammar* is a computational grammar that models the morphology and syntax of a language via a set of rules. It is specifically aimed at being precise and comprehensive in its coverage of the linguistic structure of a language, and is used for both parsing and generation.

Grammatical Framework (GF) (Ranta, 2011) is a grammar formalism for multilingual grammars. It provides a functional programming language for defining reversible mappings from interlinguas to concrete languages. GF has been used for building comprehensive resource grammars for over 40 languages. It is considered the state of the art in multilingual grammar engineering.

GF draws a distinction between abstract (language independent) syntax and concrete (language specific) syntax. An abstract syntax is defined by a set of categories and a set of functions by means of which *abstract syntax trees* are constructed. A concrete syntax, on the other hand, linearises these concepts and relations in a particular language in accordance with the linguistic requirements of the specific language and gives rise to *parse trees*. For

[3] Optional suffixes include the diminutive, augmentative, deverbative, feminative etc. They are not discussed further.

[4] Meinhof's numbering system

[5] At most one of the negative morphemes may occur. They originate from different verbal constructions.

(1) *izalukazi*
i[NPrePre][8]zi[BPre][8]alukazi[NStem][7-8]
azizukuziphekela
a[NegPre]zi[SC][8]zuku[FutNeg]zi[ReflPre]phek[VRoot]el[ApplExt]a[VT]
imifino
i[NPrePre][4]mi[BPre][4]fino[NStem][3-4]
'The old women will not cook vegetables for themselves.'

Figure 1: Illustrating the morphological complexity of an isiZulu sentence

each abstract category (`cat`) and function (`fun`) there is a corresponding linearisation type definition (`lincat`) and linearisation rule (`lin`), respectively, in the concrete syntax. A multilingual GF grammar consists of a single abstract syntax and a concrete syntax for every supported language.

As programming language, GF has, apart from its compiler, also a run-time system for performing parsing and linearisation. Translation can be achieved by parsing a sentence using the concrete syntax of one language and then linearising the resulting abstract syntax tree using the concrete syntax of a different language. *Annotation* using the GF runtime system can be achieved by parsing sentences using an RG and linearising the resulting abstract syntax trees using an adapted version of the RG that inserts annotation. The ZRG has been used in this way for morphological surface segmentation (Mkhwanazi and Marais, 2024).

Since an RG is designed to model deep linguistic structure, it encodes complete morphosyntactic information of a sentence. The deep structure is encoded explicitly in the abstract syntax tree, while surface structure is implicitly encoded in the concrete syntax and used to realise the correct surface form of the tree. Adapting an RG for morphological annotation involves adapting the strings of the grammar to make the linguistic structure explicit in the linearisation.

One advantage that this approach has over classical finite-state morphological analysis is the ability to perform sentence-internal morphological disambiguation. In classical finite-state morphological analysis, disambiguation between multiple possible analyses is seen as a next step in the NLP pipeline. The reason for this is that (finite-state) morphological analysis is performed on linguistic words and may therefore produce multiple plausible analyses that can only be disambiguated in the context in which the word occurs. When using the RG to annotate a sentence, the abstract syntax tree provides a syntactic context. Of course, when parsing a sentence, syntactic disambiguation is still required, since a sentence may have multiple possible parses. The GF parser can be utilised in a probabilistic way to assist with disambiguation at this level.

Once the morphosyntactic annotation has been done, shallow forms of annotation such as part-of-speech tagging and lemmatisation can be derived easily and accurately from it by discarding some aspects of the detailed annotation.

4 The isiZulu Resource Grammar

The ZRG is an implementation of isiZulu morphosyntax using GF (Marais and Pretorius, 2023). Apart from `cats`, `funs`, `lincats` and `lins`, two core constructs in GF that we briefly discuss by means of the example in Figure 4, are parameter (`param`) and table (`table`) types.

In linguistics, parameters are multi-valued features that represent the core grammatical properties of a language. In GF such parameters form part of the concrete syntax since they are specific to a language. Indeed "designing the parameter system ... is one of the main tasks in GF grammar writing" (Ranta et al., 2020, p.8). The parameter system of the ZRG consists of 25 parameter type definitions.

Tables operate on parameter types and are used, amongst others, to house morphological variation and inflection. For example, in the ZRG the parameter `RInit`, which differentiates between different possible initial letters (a vowel or any consonant) of a root, is defined by listing its possible values separated by 'l':

```
param RInit = RA | RE | RI | RO | RU | RC ;
```

Figure 4 shows a `table` operating on `RInit` by assigning to each value of `RInit`, the required variant (of type `Str`) of the instrumental prefix *nga-*, thereby ensuring the correct morphophonological alternation between the prefix and the root to which it is prefixed.

When developing a GF RG, it is possible either to include all the full forms of words as strings

in the grammar, and to combine these words into phrases and sentences at run time, or to include meaningful, useful subwords, i.e. morphs, as strings that are bound together at run time into words, which in turn are combined into phrases and sentences. Due to the significant reduction in compiled grammar size, the latter approach has become standard practise in GF RG development for morphologically complex languages.

5 Adapting the ZRG

In this section we discuss the chosen annotation approach for the work described in this paper, then we consider from a practical perspective how the implementation of a morphological adaptation of the ZRG (MZRG) would be done, and finally we discuss the adaptation itself in terms of the principles and design decisions involved.

5.1 Morphological annotation approach

In surface form annotation, the task is to identify where morpheme boundaries occur and to insert the correct morphological tag at these boundaries. For languages with a high degree of morphophonological alternation, this is not a trivial task. In fact, in the case of morpheme fusion in isiZulu, the question of where to mark the boundary often has no clear correct answer from a linguistic or computational point of view. However, a consistent, systematic strategy is essential. In canonical annotation, the task is to identify the morphemes that have given rise to a specific surface form and to reproduce the morphemes in their canonical form, along with the appropriate tags.

As has been noted for the case of morphological segmentation of isiZulu (Mkhwanazi and Marais, 2024), surface form morphological annotation and canonical morphological annotation typically serve different use cases. In particular, part-of-speech tags can be derived trivially from surface form annotation, while lemmatisation can be derived from canonical annotation.

In this paper, we focus on surface form annotation. Consider the syntax tree in Figure 2 that linearises to the isiZulu sentence *umfazi uyapheka* (‘the woman cooks’) using the standard ZRG. Note that the tree shows how the grammar utilises subword strings. These strings, representing morphs, bind together in specified ways at runtime to produce the correct surface forms.

Our goal in adapting the ZRG for surface form morphological annotation is to generate a syntax tree like that shown in Figure 3.

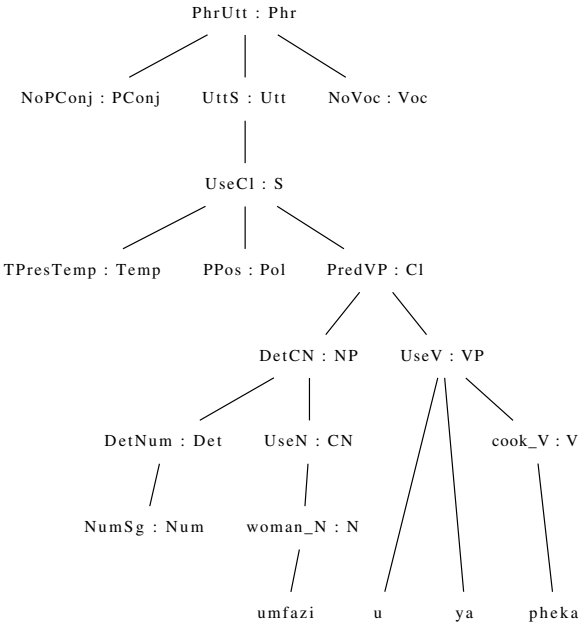

Figure 2: Syntax tree of an isiZulu sentence.

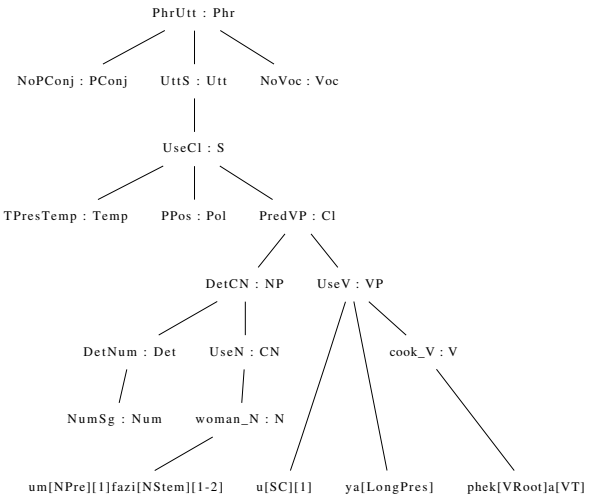

Figure 3: Syntax tree of an annotated isiZulu sentence.

5.2 Implementation of the MZRG

Since all strings contained in the ZRG are defined in the `ResZul` resource module, from a practical perspective, adapting the grammar involves making changes to a single module. With the *ZulMorph* tagset as our guide, this module was systematically changed by inserting the appropriate tags into the string elements. Figure 4 shows such a change, which involved inserting the morpheme tag at the end of each string element. In effect, the morphs

that constitute the strings of the grammar are simply decorated with the appropriate tags. Figure 5 shows what the adaptation of the same operation would have looked like, if canonical annotation were in view. Here, the dimensionality of the table is reduced to remove the mechanism for dealing with morphophonological alternation. Note that although this represents a more substantial change, it is straight forward to implement, since it consists of simplifying the grammar in systematic and predictable ways.

Other changes required for surface annotation were more complex, such as the one shown by the comparison in Figure 6. In the original ZRG, the operation `prefix_nasal` performs pattern matching on the root string in a case statement and generates the required substring, namely a root prefixed with the appropriate form of the nasal sound associated with morphemes for classes 9 and 10. For example, the first branch of the case statement models the case of roots that start with "ph", to which the nasal sound is prefixed as "m", while the "h" is dropped from the root. In the MZRG, the `[NPre]` tag is inserted, along with a tag that indicates the class associated with the prefix. This is generated by a helper operation called `noun_prefix_tag`, to which the class gender value and number are supplied as arguments. These in turn must be supplied to `prefix_nasal`. Given that this information always forms part of the syntactic context within which `prefix_nasal` is called, this change is easily integrated into the grammar.

In a small number of cases, finer distinctions had to be implemented in order to differentiate cases where different morphemes result in the same surface string. This involved adding conditional branches to existing operations.

5.3 Designing a tagset and annotation strategy

In order to retain consistency with previous work, the ZulMorph tagset and analyser served as our starting point, taking into account that it produces canonical morphological annotations, while our adaptation is aimed at surface form morphological annotation.

The main difference in surface form morphological annotation is the identification of morpheme boundaries. Due to the high degree of morphophonological alternation, and particularly morpheme fusion, that isiZulu exhibits, a strategy for systematically identifying such boundaries was required.

Since they provide the main semantic content, roots and stems are given priority. If the root stays in tact during morpheme fusion, the adjacent morpheme is considered to have absorbed the change. For example, when the class 9 relative concord (with canonical form *e-* is prefixed to the verb root *eq*, while the ZulMorph annotation is `e[RC][9]eq[VRoot]a[VT]`, the ZRG annotation is `[RC][9]eq[VRoot]a[VT]`. As in this case, this sometimes leads to "empty" morphemes. The tags are nevertheless inserted, since the annotation must still indicate which morphemes contribute to the final surface form. Apart from cases where there are differences in the tagsets of ZulMorph and the MZRG (discussed later in this section), this ensures that morpheme tag sequences are identical between the two systems for any given word. Table 1 lists the examples discussed in this section, with the ZulMorph annotation alongside the MZRG annotation in blue.

Sometimes, both the root/stem and an adjacent morpheme undergo changes, as in the case of locative palatalisation that occurs in nouns when adding the locative suffix. For example, for the noun stem *-chopho* ('pole (of the earth/magnetic)'), its singular locative form is *echosheni*. We annotate this as shown in example 2 of Table 1, where *-eni* is regarded as the locative suffix, despite its canonical form being *-ini*. The chain of phonetic processes that cause this kind of change has been documented by Poulos and Msimang (1998), showing that the initial *i* of *-ini* suffix undergoes vowel lowering to become an *e*, which then triggers further processes within the noun stem. In cases of stems ending in *e*, the suffix is considered to be *-ni*. Therefore, the annotation for *ezweni* ('in/to the country') would be `e[LocPre]zwe[NStem][5-6]ni[LocSuf]` (see example 3 in Table 1).

For other morphemes, any parts of a morpheme that are retained after a sound change are considered part of the morpheme. For example, the locative for the class 1a noun *kudadewenu* ('to your sister') is annotated as shown in example 4 of Table 1.

Generally, if no other principle can be applied, syllable boundaries guide morpheme boundary identification in cases of morpheme fusion.

One major difference between ZulMorph and the MZRG is the way noun prefixes are handled. ZulMorph consistently distinguishes the pre-prefix

```
-- with
instrPref : RInit => Str = table {
  RU => "ngo" ;
  RI => "nge" ;
  RO => "ngo" ;
  _  => "nga"
} ;
```

```
-- with
instrPref : RInit => Str = table {
  RU => "ngo[AdvPre]" ;
  RI => "nge[AdvPre]" ;
  RO => "ngo[AdvPre]" ;
  _  => "nga[AdvPre]"
} ;
```

Figure 4: Original and adapted code snippets for generating variants of the instrumental prefix.

```
-- with
instrPref : Str = "nga[AdvPre]" ;
```

Figure 5: Adapted code for canonical annotation.

(with tag `[NPrePre]`) and the base prefix (with tag `[BPre]`) which together constitute the noun prefix. Due to the frequency with which at least of these morphemes becomes "empty" due to morphophonological alternation, the MZRG tags the noun prefix as a single morpheme with tag `[NPre]`.

Another significant difference is the lack of annotation for verb root extensions. This was a design decision of the ZRG, in which such extensions were assumed to form part of the extended verb roots in a lexicon, instead of being handled productively within the morphosyntactic implementation of the ZRG. Consequently, *ushisa* is annotated by ZulMorph as `u[SC][3]sh[VRoot]is[CausExt]a[VT]`, while the MZRG annotation is `u[SC][3]shis[VRoot]a[VT]` (see example 5 of Table 1).

6 Applying the MZRG

The goal of this section is to show the MZRG in action as a morphological analyser, and to contextualise it as a tool among other state-of-the-art morphological analysers for isiZulu. We therefore contrast the MZRG to both ZulMorph (as another rule-based tool) and the CTexT Core Technologies (a data-driven tool).

The corpus used as our basis is a treebank of 100 sentences taken from an isiZulu textbook (Taljaard and Bosch, 1988). The sentences were chosen to represent a linguistically diverse set of sentences, originally used to demonstrate and teach a variety of linguistic constructions in isiZulu. It therefore represents a suitable set of sentences for comparing the ZulMorph analysis with that of the MZRG. The 100 sentences comprise 243 tokens, of which 184 are unique, although they appear in different syntactic contexts (and may therefore have different analyses).

A GF treebank can be obtained via direct engineering or by parsing a corpus of sentences using the probabilistic parsing functionality of the GF runtime. This corpus of 100 sentences was obtained using a combination of both (Marais and Pretorius, 2023). The sentences were therefore known to fall within the definition of the original ZRG.

6.1 MZRG compared to a rule-based tool

When comparing two rule-based tools, similar assumptions about the performance of the tools exist, namely that rule-based tools are engineered for correctness and may fail entirely on ungrammatical input or input that falls outside of its definition. In comparing the MZRG to ZulMorph, the aim is not to evaluate accuracy, but rather to discuss the effects of the different designs of the two systems on a set of well-formed sentences.

While our corpus was chosen to be linguistically diverse, it cannot serve as a suitable basis for drawing statistical conclusions about the differences that ZulMorph and the MZRG would produce on a typical isiZulu text. Nevertheless, some insight may still be gained by quantifying the differences within this set.

For our comparison, we use both the original ZRG and the MZRG to linearise all 100 sentences. We then use ZulMorph to obtain analyses for each token in the ZRG linearisations, which are then manually disambiguated. We also perform a simplification of the ZulMorph analyses in order to discount the major known differences between the two sets, namely the consolidation of the noun preprefix and base prefix, as well as the extensions into the verb roots.

We therefore have a parallel list of 243 analysed tokens from a corpus of 100 sentences, resulting in an average sentence length of 2.4 tokens. We

```
prefix_nasal : Str -> Str = \root -> case root of {
  "ph"+x => "mp" + x ;
  "Ph"+x => "mP" + x ;
  "bh"+x => "mb" + x ;
  "Bh"+x => "mB" + x ;
  -- ...

prefix_nasal : Str -> ClassGender -> Number -> Str = \root,classgender,number -> let
  class_tag = noun_prefix_tag classgender number ;
in
  case root of {
  "ph"+x => "m[NPre]"+class_tag+"p" + x ;
  "Ph"+x => "m[NPre]"+class_tag+"P" + x ;
  "bh"+x => "m[NPre]"+class_tag+"b" + x ;
  "Bh"+x => "m[NPre]"+class_tag+"B" + x ;
  -- ...
```

Figure 6: Original and adapted code snippets for roots prefixed with the nasal sound of with classes 9 and 10.

	isiZulu word	Analyses
1	eqa	e[RC][9]eq[VRoot]a[VT]
		[RC][9]eq[VRoot]a[VT]
2	echosheni	e[LocPre]i[NPrePre][5]li[BPre][5]chopho[NStem][5-6]ini[LocSuf]
		e[LocPre]chosh[NStem][5-6]eni[LocSuf]
3	ezweni	e[LocPre]u[NPrePre][14]bu[BPre][14]zwe[NStem][14]ini[LocSuf]
		e[LocPre]zwe[NStem][5-6]ni[LocSuf]
4	kudadewenu	ku[LocPre]u[NPrePre][1a]dadewenu[NStem][1a-2a]
		k[LocPre]u[NPre][1a]dadewenu[NStem][1a-2a]
5	ushisa	u[SC][3]sh[VRoot]is[CausExt]a[VT]
		u[SC][3]shis[VRoot]a[VT]

Table 1: A comparison of annotations by ZulMorph and MZRG (blue)

first determine the unique analyses in each list: the ZulMorph list contains 187 unique analyses, while the MZRG list contains 188. The discrepancy is due to the word *ukudla* being used both as a noun meaning 'food' and as an infinitive meaning 'to eat' in different trees. In simplified form, ZulMorph always provides the deverbative analysis, namely `uku[NPre][15]dl[VRoot]a[VT]`, while in different syntactic contexts, depending on the composition of the abstract syntax tree, the MZRG provides `uku[NPre][15]dl[VRoot]a[VT]` (in the sentence meaning 'To eat is good') and `uku[NPre][15]dla[NStem][15]` (in the sentence meaning 'The woman cooks the food.').

Among the 243 tokens in each list, 127 analyses (52.2%) are identical, of which 101 are unique. These words represent those that have undergone no morphophonological alternation, and hence not only the tag sequence, but also the surface forms are the same. On the other hand, 208 analyses (85.6%) share identical tag sequences, of which 159 are unique. In such cases, ZulMorph and the MZRG produces the same tag sequence (apart from the simplification mentioned above), although the surface form of the token exhibits morphophonological alternation.

The differences in tag sequences for the remaining tokens are the result of different design decisions taken in the development of the two systems, such as differences in how the subject concord is tagged in different contexts, cases where nouns belong only to one class and not to a pair of classes (this is not encoded in the ZRG), a distinction in the ZRG between locative nouns and other adverbs, and slight differences in how class information is associated with certain tags, such as the copulative prefix. However, both sets of analyses are completely accurate with respect to these design decisions.

6.2 MZRG compared to a data-driven tool

Having shown how the MZRG compares to ZulMorph, we now contrast it to a data-driven tool. Arguably the state-of-the-art data-driven morphological analyser that is freely available for isiZulu is the CTexT Core Technologies (CTT) tool bundled with the *ctextcore* Python package (Centre for Text Technology). It produces an analysis for any input, in contrast to the ZRG and ZulMorph. Gener-

Measure	Count	Percentage
Correct analysis	103	42.4%
Correct stem/root	130	53.5%
Plausible sequence	166	68.3%

Table 2: Preliminary comparison of accuracy

ally speaking, rule-based tools prioritise accuracy over coverage, while data-driven tools prioritise coverage, often resulting in lower accuracy. This is especially the case in resource-scarce environments.

Our aim in this comparison (Table 2) is to give some indication of the difference in accuracy of the CTT analyser compared to the MZRG. We therefore run the CTT analyser on the same set of linguistically diverse sentences. The results are manually evaluated and successes and errors categorised: for each analysis of a token, we indicate if it is correct, whether the stem or root of the token is correctly identified, and whether the morpheme sequence is at all plausible. Note that the final category does not consider whether the analysis of any given token is plausible for that word, but only whether the tag sequence is one that *could* exist in isiZulu. Typical errors are cases where two roots or stems are present in the analysis, where verb prefixes and a noun stem or (incorrect) noun prefixes and a verb root are present, or where prefixes appear at the end of a sequence.

One concern with the CTT analyser is its inconsistency with regards to surface form or canonical annotation. The data on which it was trained comprises canonical annotations (Gaustad and McKellar, 2024), but in many cases, the CTT failed to provide canonical forms of the morphemes. Our analysis in Table 2 did not penalise the CTT for this, but it may prove problematic for using the CTT within a larger NLP pipeline.

The three measures revealed subsets within the analyses: the set of tokens correctly analysed is a subset of those for which the stem or root is correctly identified, which in turn is a subset of those analyses that represent a plausible morpheme sequence. Viewed from a user perspective, on a well-behaved set of sentences with average length of 2.4 tokens per sentence, a user can expect around a third of analyses to be implausible, with the root or stem incorrectly identified. Moreover, a user can expect just less than half of analyses to incorrectly identify the root or stem and just below 60% of analyses to be erroneous in some way.

A full comparison of the accuracy and coverage, with reference to precision and recall, of the MZRG and the CTT on a larger, less curated and more natural corpus is beyond the scope of this paper and forms part of future work. The comparison given here should be regarded as a preliminary result indicating that a more in-depth comparative evaluation may be worthwhile.

6.3 Obtaining accurate annotations

We end this section by briefly sketching the workflows required for using the MZRG, ZulMorph and the CTT analyser to obtain accurate morphological annotations for an isiZulu corpus.

ZulMorph is highly accurate (Bosch, 2020) and with a lexicon of several thousand items, provides excellent coverage of the isiZulu language. However, additional manual effort is required to disambiguate all the possible analyses for each token. This kind of approach was followed in developing the annotated corpora on which the CTT analyser was trained (Gaustad and McKellar, 2024).

CTT analyser will produce output on any input, but requires additional manual effort in correcting erroneous analyses. Our analysis shows that around 60% of tokens may need to be corrected in this way, or around 50% if only correct identification of stems or roots is required.

MZRG depends on the availability of a GF treebank. The GF runtime in conjunction with the original ZRG provides a probabilistic parser, which mitigates the effort involved, namely that of manually disambiguating between possible trees. Once the treebank is obtained, highly reliable, syntactically contextualised annotations may be generated via the MZRG.

7 Conclusion

We presented an adapted version of the ZRG for morphological annotation (MZRG). The string elements of the ZRG were decorated with morphological tags, resulting in contextually appropriate morphological analyses. The adaptation makes explicit the morphosyntactic model of the ZRG during the process of linearisation. We contrasted the MZRG with two other state-of-the-art tools to show how it could be used on a linguistically diverse set of isiZulu sentences.

References

Sonja Bosch. 2020. Computational morphology systems for Zulu–a comparison. *Nordic Journal of African Studies*, 29(3):28–28.

Centre for Text Technology. CTexT Core Technologies for South African languages.

Tanja Gaustad and Cindy A McKellar. 2024. Updated morphologically annotated corpora for 9 south african languages. *Journal of Open Humanities Data*, 10(1).

Kedir Yassin Hussen, Walelign Tewabe Sewunetie, Abinew Ali Ayele, Sukairaj Hafiz Imam, Shamsuddeen Hassan Muhammad, and Seid Muhie Yimam. 2025. The state of Large Language Models for African languages: Progress and challenges.

Nancy Ide and James Pustejovsky. 2017. *Handbook of Linguistic Annotation*, 1st edition. Springer Publishing Company, Incorporated.

Inge M. Kosch. 2006. *Topics in morphology in the African language context*. Unisa Press.

Laurette Marais and Laurette Pretorius. 2023. Parsing IsiZulu Text Using Grammatical Framework. In *Distributed Computing and Artificial Intelligence, Special Sessions I, 20th International Conference*, pages 167–177, Cham. Springer Nature Switzerland.

Sthembiso Mkhwanazi and Laurette Marais. 2024. Generation of segmented isiZulu text. *Journal of the Digital Humanities Association of Southern Africa*, 5(1).

Tumi Moeng, Sheldon Reay, Aaron Daniels, and Jan Buys. 2021. Canonical and surface morphological segmentation for Nguni languages. In *Southern African Conference for Artificial Intelligence Research*, pages 125–139. Springer.

George Poulos and Christian T. Msimang. 1998. *A linguistic analysis of Zulu*. Via Afrika Ltd.

Laurette Pretorius and Sonja Bosch. ZulMorph: Finite state morphological analyser for Zulu (Version 20190103).

Laurette Pretorius and Sonja Bosch. 2010. Finite State Morphology of the Nguni Language Cluster: Modelling and Implementation Issues. In *Finite-State Methods and Natural Language Processing*, pages 123–130, Berlin, Heidelberg. Springer Berlin Heidelberg.

DJ Prinsloo and Gilles-Maurice de Schryver. 2001. Corpus applications for the African languages, with special reference to research, teaching, learning and software. *Southern African Linguistics and Applied Language Studies*, 19(1-2):111–131.

Aarne Ranta. 2011. *Grammatical Framework: Programming with multilingual grammars*. CSLI Publications, Stanford, California.

Aarne Ranta, Krasimir Angelov, Normunds Gruzitis, and Prasanth Kolachina. 2020. Abstract syntax as interlingua: Scaling up the Grammatical Framework from controlled languages to robust pipelines. *Computational Linguistics*, 46(2):425–486.

P.C. Taljaard and S.E. Bosch. 1988. *Handbook of IsiZulu*. J.L. Van Schaik.

Elsabé Taljard and Gilles-Maurice de Schryver. 2016. A corpus-driven account of the noun classes and genders in Northern Sotho. *Southern African Linguistics and Applied Language Studies*, 34(2):169–185.

Katharina von der Wense, Ryan Cotterell, and Hinrich Schütze. 2016. Neural morphological analysis: Encoding-decoding canonical segments. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 961–967.