Setswana Tokenisation and Computational Verb Morphology: Facing the Challenge of a Disjunctive Orthography

Rigardt Pretorius, Ansu Berg, Laurette Pretorius, Biffie Viljoen

Introduction

Setswana, a Bantu language in the Sotho group, is one of the eleven official languages of South Africa. The language is characterised by a disjunctive orthography, mainly affecting the important word category of verbs. In particular, verbal prefixal morphemes are usually written disjunctively, while suffixal morphemes follow a conjunctive writing style.

Verb morphology

The most basic form of the verb in Setswana consists of an infinitive prefix + a root + a verb-final suffix, for example, ***go bona*** (to see) consists of the infinitive prefix ***go***, the root ***bon-*** and the verb-final suffix ***-a*****.** While verbs in Setswana may also include various other prefixes and suffixes, the root always forms the lexical core of a word.

Prefixes of the Setswana verb

- The ***subject agreement morphemes*****,** written disjunctively, include non-consecutive subject agreement morphemes and consecutive subject agreement morphemes. For example, the non-consecutive subject agreement morpheme for class 5 is ***le*** as in ***lekau le a tshega*** (the young man is laughing), while the consecutive subject agreement morpheme for class 5 is ***la*** as in ***lekau la tshega*** (the young man **then** laughed).
- The ***object agreement morpheme*** is written disjunctively in most instances, for example ***ba di bona*** (**they** see it).
- The ***reflexive morpheme i-*** (-self) is always written conjunctively to the root, for example ***o ipona*** (he sees himself).
- The ***aspectual morphemes*** are written disjunctively and include the present tense morpheme ***a***, the progressive morpheme ***sa*** (still) and the potential morpheme ***ka*** (can). Examples are ***o a araba*** (he answers), ***ba sa ithuta*** (they are **still** learning) and ***ba ka ithuta*** (they **can** learn).
- The ***temporal morpheme tla*** (indicating the future tense) is written disjunctively, for example ***ba tla ithuta*** (they **shall** learn).
- The ***negative morphemes ga, sa*** and ***se*** are written disjunctively. Examples are ***ga ba ithute*** (they **do not** learn), ***re sa mo thuse*** (we **do not** help him), ***o se mo rome*** (**do not** send him).

Suffixes of the Setswana verb

Various morphemes may be suffixed to the verbal root and follow the conjunctive writing style.

Tokenisation

Tokenisation may be defined as the process of breaking up the sequence of characters in a text at the word boundaries. It may therefore be regarded as a core technology in natural language processing.

Example 1**:** In the English sentence **"I shall buy meat"** the **four tokens** (separated by "/") are **I / shall / buy / meat**. However, in the Setswana sentence ***Ke tla reka nama*** (I shall buy meat) the **two tokens** are ***Ke tla reka / nama***.

Example 2: Improper tokenisation may distort corpus linguistic conclusions and statistics. Compare the following:
A*/ *o itse*/ *rre*/ *yo*/*? (Do you know this gentleman?) **Interrogative particle**;
Re bone*/ *makau*/ *a*/ *maabane*/*. (We saw these young men yesterday.) **Demonstrative pronoun**;
Metsi*/ *a*/ *bollo*/*. (The water is hot.) **Descriptive copulative**;
***Madi*/ *a*/ *rona*/ *a*/ *mo*/ *bankeng*/.** (Our money (the money of us) is in the bank.) **Possessive particle and descriptive copulative**;
Mosadi*/ *a ba bitsa*/*. (The woman (then) called them.) **Subject agreement morpheme**;
Dintšwa*/ *ga di a re bona*/*. (The dogs did not see us.) **Negative morpheme**, which is concomitant with the negative morpheme ***ga*** when the negative of the perfect is indicated, thus an example of a separated dependency.

In the six occurrences of ***a*** above only four represent orthographic words that should form part of a word frequency count for ***a***.

Clearly, Setswana tokenisation cannot be based solely on whitespace, as is the case in many alphabetic, segmented languages, including the conjunctively written Nguni group of South African Bantu languages.

This paper shows how a combination of two tokeniser transducers and a finite-state (rule-based) morphological analyser may be combined to effectively solve the Setswana tokenisation problem. The approach has the important advantage of bringing the processing of Setswana beyond the morphological analysis level in line with what is appropriate for the Nguni languages. This means that the challenge of the disjunctive orthography is met at the tokenisation/morphological analysis level and does not in principle propagate to subsequent levels of analysis.

Research question

Can the development and application of a precise tokeniser and morphological analyser for Setswana resolve the issue of disjunctive orthography? If so, subsequent levels of processing could exploit the inherent structural similarities between the Bantu languages.

Our approach

Our underlying assumption is that the Bantu languages are structurally very closely related.

Our contention is that precise tokenisation will result in comparable morphological analyses, and that the similarities and structural agreement between Setswana and languages such as Zulu will prevail at subsequent levels of syntactic analysis, which could and should then also be computationally exploited.

Our approach is based on the novel combination of two tokeniser transducers and a morphological analyser for Setswana.

Morphological analyser

The finite-state morphological analyser prototype for Setswana, developed with the Xerox finite state toolkit, implements Setswana morpheme sequencing (morphotactics) by means of a **lexc** script containing cascades of so-called lexicons, each of which represents a specific type of prefix, suffix or root.

Sound changes at morpheme boundaries (morphophonological alternation rules) are implemented by means of **xfst** regular expressions.

These **lexc** and **xfst** scripts are then compiled and subsequently composed into a single finite state transducer, constituting the morphological analyser.

While the implementation of the morphotactics and alternation rules is, in principle complete, the word root lexicons still need to be extended to include all known and valid Setswana roots.

The verb morphology is based on the assumption that valid verb structures are disjunctively written.

For example, the verb token ***re tla dula*** (we will sit/stay) is analysed as follows:

```
Verb(INDmode),(FUTtense,Pos): AgrSubj-1p-Pl+TmpPre+[dul]+Term
```

or

```
Verb(PARmode),(FUTtense,Pos): AgrSubj-1p-Pl+TmpPre+[dul]+Term
```

Both modes, indicative and participial, constitute valid analyses. The occurrence of multiple valid morphological analyses is typical and would require (context dependent) disambiguation at subsequent levels of processing.

Tokeniser

A grammar for linguistically valid verb constructions is implemented with **xfst** regular expressions. The key principle on which the tokeniser is based is right-to-left longest match.

We note that:

(i) it may happen that a longest match does not constitute a valid verb construct;

(ii) the right-to-left strategy is appropriate since the verb root and suffixes are written conjunctively and therefore should not be individually identified at the tokenisation stage while disjunctively written prefixes need to be recognised.

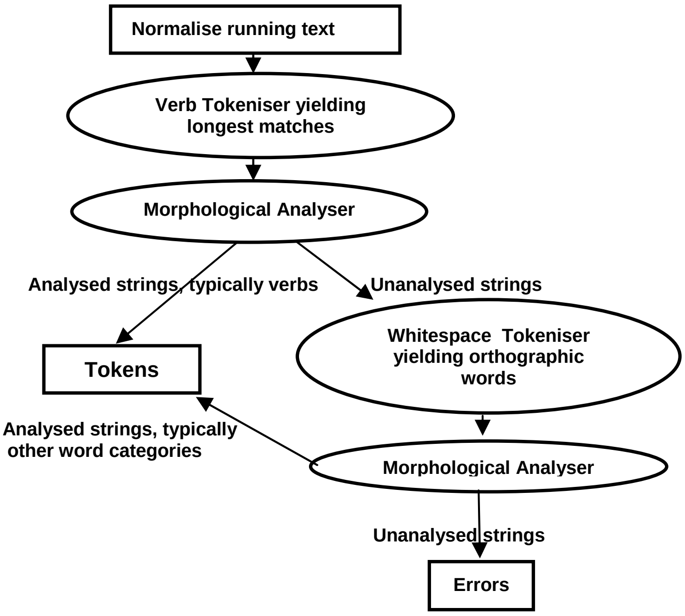
The two aspects that need further clarification are:

(i) How do we determine whether a morpheme sequence is valid?

(ii) How do we recognise disjunctively written prefixes?

Methodology

Central to our approach is the assumption that only analysed tokens are valid tokens and strings that could not be analysed are not valid linguistic words.


Normalise running text
Verb Tokeniser yielding longest matches
Morphological Analyser
Analysed strings, typically verbs
Unanalysed strings
Tokens
Whitespace Tokeniser yielding orthographic words
Analysed strings, typically other word categories
Morphological Analyser
Unanalysed strings
Errors

***Step 1*:** Normalise test data (running text) by removing capitalisation and punctuation;
***Step 2*:** Tokenise on longest match right-to-left;
Step 3: Perform a morphological analysis of the “tokens” from step 2;
Step 4: Separate the tokens that were successfully analysed in step 3 from those that could not be analysed;
Step 5: Tokenise all unanalysed “tokens” from step 4 on whitespace;
[Example: unanalysed ***wa me*** becomes ***wa*** and ***me***.]
Step 6: Perform a morphological analysis of the “tokens” in step 5;
Step 7: Again, as in step 4, separate the analysed and unanalysed strings resulting from step 6;
Step 8: Combine all the valid tokens from steps 4 and 7.

Finally a comparison of the correspondences and differences between the hand-tokenised tokens (hand-tokens) and the tokens obtained by computational means (auto-tokens) is necessary in order to assess the reliability of the described tokenisation approach.

Test data and results

Since the purpose was to establish the validity of the tokenisation approach, we made use of a short Setswana text of 547 orthographic words, containing a variety of verb constructions (see Table 1). The text was tokenised by hand and checked by a linguist in order to provide a means to measure the success of the tokenisation approach. Furthermore, the text was normalised not to contain capitalisation and punctuation. All word roots occurring in the text were added to the root lexicon of the morphological analyser to ensure that limitations in the analyser would not influence the tokenisation experiment.

Examples of output of step 2:
ke tla nna
o tla go kopa
le ditsebe
Examples of output of step 3:
Based on the morphological analysis, the first two of the above longest matches are tokens and the third is not. The relevant analyses are:
ke tla nna
`Verb(INDmode), (FUTtense,Pos): AgrSubj-1p-Sg+TmpPre+[nn]+Term`
o tla go kopa
`Verb(INDmode), (FUTtense,Pos): AgrSubj-Cl1+TmpPre+AgrObj-2p-Sg+[kop]+Term`
Examples of output of step 5:
le, ditsebe
Examples of output of step 6:
le
`CopVerb(Descr), (INDmode), (FUTtense,Neg): AgrSubj-Cl5`
ditsebe
`NPre10+[tsebe]`

The results of the tokenisation procedure applied to the test data, is summarised in Tables 1 and 2.

Token length (in orthographic words)	**Test data**	**Correctly tokenised**
2	84	68
3	25	25
4	2	2

Table 1. Verb constructions

Table 1 shows that 111 of the 409 tokens in the test data consist of more than one orthographic word (i.e. verb constructions) of which 95 are correctly tokenised. Moreover, it suggests that the tokenisation improves with the length of the tokens.

	Tokens	**Types**
Hand-tokens, *H*	409	208
Auto-tokens, *A*	412	202
H ∩ *A*	383 (93.6%)	193 (92.8%)
A \ *H*	29	9
H \ *A*	26	15
Precision, P	0.93	0.96
Recall, R	0.94	0.93
F-score, 2PR/(P+R)	0.93	0.94

Table 2. Tokenisation results

Issues

- Longest matches that allow morphological analysis, but do not constitute tokens. Examples are ***ba ba neng***, ***e e siameng*** and ***o o fetileng***. In these instances the tokeniser did not recognise the qualificative particle. The tokenisation should have been ***ba/ ba neng***, ***e/ e siameng*** and ***o/ o fetileng***.
- Longest matches that do not allow morphological analysis and are directly split up into single orthographic words instead of allowing verb constructions of intermediate length. An example is ***e le monna***, which was finally tokenised as ***e/ le/ monna*** instead of ***e le/ monna***.
- Finally, perfect tokenisation is context sensitive. The string ***ke tsala*** should have been tokenised as ***ke/ tsala*** (noun), and not as the verb construction ***ke tsala***. In another context it can however be a verb with ***tsal-*** as the verb root.

In conclusion, we have successfully demonstrated that the novel combination of a precise tokeniser and morphological analyser for Setswana could indeed form the basis for resolving the issue of disjunctive orthography.