Improving the Computational Morphological Analysis of a Swahili Corpus for Lexicographic Purposes*

Guy De Pauw, *CNTS — Language Technology Group, University of Antwerp, Antwerp, Belgium; School of Computing and Informatics, University of Nairobi, Nairobi, Kenya; and Xhosa Department, University of the Western Cape, Bellville, Republic of South Africa (guy.depauw@ua.ac.be)*,
and
Gilles-Maurice de Schryver, *Department of African Languages and Cultures, Ghent University, Ghent, Belgium; Xhosa Department, University of the Western Cape, Bellville, Republic of South Africa; and TshwaneDJe HLT, Pretoria, Republic of South Africa (gillesmaurice.deschryver@UGent.be)*

Abstract: Computational morphological analysis is an important first step in the automatic treatment of natural language and a useful lexicographic tool. This article describes a corpus-based approach to the morphological analysis of Swahili. We particularly focus our discussion on its ability to retrieve lemmas for word forms and evaluate it as a tool for corpus-based dictionary compilation.

Keywords: LEXICOGRAPHY, MORPHOLOGY, CORPUS ANNOTATION, LEMMATIZATION, MACHINE LEARNING, SWAHILI (KISWAHILI)

Samenvatting: Accuratere computationele morfologische analyse van een Swahili corpus voor lexicografische doeleinden. Computationele morfologische analyse is een belangrijke eerste stap in de automatische verwerking van natuurlijke taal en een nuttig lexicografisch hulpmiddel. Dit artikel beschrijft een corpusgebaseerde aanpak voor de morfologische analyse van het Swahili. We concentreren ons hierbij vooral op de lemmatiseringseigenschappen van het ontwikkelde systeem en evalueren het als een hulpmiddel bij de corpusgebaseerde ontwikkeling van woordenboeken.

Sleutelwoorden: LEXICOGRAFIE, MORFOLOGIE, CORPUSANNOTATIE, LEMMATISERING, AUTOMATISCHE LEERTECHNIEKEN, SWAHILI (KISWAHILI)

* An earlier version of this article was presented at the Thirteenth International Conference of the African Association for Lexicography, organized by the Bureau of the *Woordeboek van die Afrikaanse Taal*, Stellenbosch, Republic of South Africa, 1–3 July 2008.

1. Bantu computational lexicography

The last couple of years have seen a definite empirical shift in Bantu lexicography. The integration of corpus data in the arduous process of dictionary compilation allows the lexicographer to semi-automatically unearth examples for the dictionary entries in actual language use. It has become unimaginable to compile a wide-coverage dictionary for a Bantu language without the use of a large language corpus and a functional corpus query package (CQP). In De Schryver and De Pauw (2007) it was shown how the fields of natural language processing (NLP) and lexicography can collaborate towards enhancing the functionality of a CQP, by integrating a fast and accurate data-driven part-of-speech (POS) tagger.

In this article, we investigate how another typical NLP component — namely morphological analysis — can be developed with a minimal amount of manual effort, and demonstrate how it can be used as a CQP component. As a case study, we choose Swahili, a widely spoken Bantu language with no (publicly accessible) morphological analyzer or morphologically annotated lexicon. We will show how both of these resources can easily be developed using a machine-learning approach.

In Section 2, we describe some of the current approaches to morphological analysis and provide a comprehensive overview of previous work on Bantu languages. We then discuss, in Section 3, the construction of a Swahili morphological database, which will be used as an information source for the machine-learning approach described in Section 4. After a quantitative evaluation of the system, in Section 5, we conclude, in Section 6, with a discussion of the current state of affairs and some pointers to future work.

2. Computational morphological analysis

Computational morphological analysis is an important first step in the automatic treatment of natural language. Finding the *minimal meaning bearing units* that constitute a word, can provide a wealth of linguistic information that becomes useful when processing the text on other levels of linguistic description, such as phonology, syntax and even semantics.

In most practical language technology applications, morphological analysis is used to perform *lemmatization*. A typical application of a lemmatizer is integrated in *Google*'s search facility, which automatically lemmatizes a search term like 'discussions' to also produce hits for the word form 'discussion'. Lemmatization is also often used to enhance statistical models of language in other language technology applications, like machine translation (Oflazer 2008) and speech recognition (De Pauw et al. 2004). There are however few publications that explicitly discuss the obvious lexicographic application of a lemmatizer, i.e. as a CQP component (but see Christ 1994, Kilgarriff et al. 2008). In this article, we will therefore focus our discussion on the lemmatization abilities of

the developed system and evaluate it as a tool for corpus-based dictionary compilation.

2.1 Current approaches

The most widely used approach to computational morphology uses the *two-level formalism* (Koskenniemi 1983). This rule-based method typically operates on the character level and associates each character with a given morphological property. The approach distinguishes between the surface and lexical realizations of a given morpheme (hence two-level) and attempts to establish a mapping between the two. The two-level formalism uses a (large) collection of finite-state transducers which each implement a particular morphological rule. While the framework itself is language independent, these rules typically need to be manually constructed for each language and/or sub-domain of a language, which makes the development of such a morphological analyzer very costly and time-consuming.

In the late nineties a few interesting corpus-based alternatives have surfaced. Rather than requiring expert linguistic knowledge to construct a morphological analyzer, these approaches automatically induce the required information from a morphologically annotated data set, such as CELEX (Baayen et al. 1995). Using statistical processing (Masaaki 1999) and/or machine-learning techniques (Van den Bosch and Daelemans 1999), these data-driven methods establish an effective and truly language-independent technique for morphological analysis, that can easily be ported to new domains. Furthermore, manually constructed rule-based analyzers typically do not significantly outperform data-driven approaches in a direct comparison (De Pauw et al. 2004).

In more recent years, research on computational morphology has mainly concentrated on unsupervised approaches. These methods attempt to automatically induce the morphological properties of a language on the basis of raw, unannotated text, using minimum-distance edit metrics and pattern-matching techniques.

2.2 Bantu computational morphological analysis

While great advances have been made for many Indo-European and Asian languages, most computational morphological models for Bantu languages are still in the developmental stage. This is not only due to the relatively limited commercial interest in these languages, but also because of the often intricate morphology, which renders both the construction of rule-based and data-driven methods troublesome.

Most of the research on computational morphology of Bantu languages is being conducted in South Africa and is rooted in the rule-based two-level formalism. Morphological analyzers are being developed for Northern Sotho

(Kotzé and Anderson 2005, Bosch et al. 2006, Anderson et al. 2007), Zulu (Taljard and Bosch 2005, Bosch et al. 2006, Pretorius and Bosch 2007), Xhosa, Swazi and Tswana (Bosch et al. 2006). Smaller projects have also looked into aspects of the morphology of Shona (Ridings and Mavhu 2002), Zimbabwean Ndebele (Maphosa 2002), Kwanyama (Hurskainen and Halme 2001) and Rwanda (Muhirwe 2007).

The rule-based two-level morphology formalism has also been applied to the verbal morphology of Gusii (Elwell 2006) and to Swahili (Hurskainen 1992, 1996, 2004). The latter rule-based morphological analyzer for Swahili is known as SALAMA, and was used to lemmatize the *Helsinki Corpus of Swahili* (HCS, Hurskainen 2004a). This system is not publicly available, however.

The user interface for two Swahili dictionaries on the Internet, viz. the *Kamusi Project* and the *Online Swahili–English Dictionary* (Hillewaert, Joffe and De Schryver 2008), integrate rule-based morphological analyzers. These are useful for dictionary queries, but are limited to analyzing (verb) forms for which the underlying lemma is also present in the dictionary.

Data-driven approaches are indeed few and far between, with some notable exceptions. A data-driven morpho-syntactic tagger was developed for Swahili[1] (De Pauw et al. 2006) and Northern Sotho[2] (De Schryver and De Pauw 2007). An unsupervised approach to morphological analysis has been applied to Luo, a Nilotic language (De Pauw et al. 2007) and Gikuyu (De Pauw and Wagacha 2007). The latter actually constitutes a viable alternative to unsupervised methods such as *AutoMorphology* (Goldsmith 2001) or *Morfessor* (Creutz et al. 2005), which are not well equipped to handle Bantu morphology. Compare also with Elwell (2008).

Lindén (2008) describes a semi-supervised method for the lemmatization of Swahili words. The method uses the annotation of HCS to induce a probabilistic model that is able to guess base forms of previously unseen words.

Finally, Elwell (2008) describes a novel technique for verbal morphological analysis of Swahili. It uses the insight that Swahili morphemes are open syllables and monosyllabic to create a maximum entropy-based classifier that categorizes syllables for different aspects of the verbal morphology. While limited to verbal morphology only, it is to our knowledge the only machine-learning approach to morphological analysis that specifically caters to a Bantu language in terms of knowledge representation.

3. Towards a Swahili morphological database

The research described in this article wants to fill the void by creating a data-driven morphological analyzer for Swahili that handles all morphologically productive word classes. To this end, one needs a morphologically annotated word list. While this is not available as such, one can go a long way by extracting the necessary information from HCS, lemmatized using the SALAMA morphological analyzer.

In HCS, every word is associated with its lemma, POS-tag, some morphological features and an English translation, like in examples (1) and (2).

(1) **ulikanusha** kanusha V [1/2-SG2-SP VFIN PAST SV EXT: SVO-C CAUS:sh :EXT] deny, disprove, refute, negate

(2) **ulikoanzia** anza V [1/2-SG2-SP VFIN PAST 15-SG-REL SV SVO EXT: APPL :EXT] begin, establish

We can use this information to perform pattern-matching and match the lemma to the word form. Through this operation we can automatically induce a morphologically segmented surface and lexical representation of the word form, in which we distinguish a prefix group (**[P]**), the root morpheme (**[R]**) and a suffix group (**[S]**). In some cases, this is straightforward, as for the entry in example (1) which can easily be transformed into example (3).

(3) **ulikanusha** kanusha → Surface: uli**[P]** + kanusha**[R]**
→ Lexical: uli**[P]** + kanusha**[R]**

For the entry in example (2), this leads to the creation of a bound root morpheme *anz-* in the surface representation, associated with the full lemma *anza* in the lexical representation.

(4) **ulikoanzia** anza → Surface: uliko**[P]** + anz**[R]** + ia**[S]**
→ Lexical: uliko**[P]** + anza**[R]** + ia**[S]**

Using this method, we automatically extracted a morphological database of 97 000 entries from the 9.7-million-word HCS. We retained word forms from morphologically productive word classes only, and filtered out noise as much as possible by discarding low-frequency tokens and English words. However, some misspelt words (e.g. *uuondoe*) and non-English loan words (e.g. *Deutsche*) still make up for some noise in the data that cannot be automatically discarded.

Since HCS has been lemmatized using an automated method, quite a few erroneous and inconsistent lemmatizations can be observed in the data. We therefore randomly extracted 10% of the data from the morphological database and had it manually annotated according to the prefix-root-suffix ([P]-[R]-[S]) protocol illustrated in examples (3) and (4). The availability of this manually annotated *gold-standard evaluation set* does not only allow us to cross-check the accuracy of our system on clean data, but also enables a *post-hoc* evaluation of the rule-based approach used to annotate HCS.

Similarly to the annotation approach described in De Schryver and De Pauw (2007), we used Microsoft Excel as the annotation environment. The annotation sheet seen in Figure 1 lists each word on a separate row. The word form itself is listed in Column A. Column B contains a sentence extracted from HCS, illustrating that word form in context. The minimized sentence can be displayed in full by double-clicking on the cell. Columns C and onwards list the individual characters of the word form from Column A, separated by blank cells.

Each blank cell has a drop-down box available with three options: **P** (end of prefix group), **R** (end of root group) and **S** (end of suffix group). The annotator can quickly move through the annotation process using only the keyboard or mouse clicks. In practice, the **S** annotation does not need to be indicated, as any character to the right of the root group automatically constitutes the suffix group (see e.g. line 7304). Furthermore, if the word does not have a suffix group, only the **P** annotation needs to be identified by the annotator, since what remains is automatically considered to be the root group (see e.g. line 7314).

In this way, the surface representation of the morpheme boundaries is annotated. In a second annotation step, the lexical representations of the roots, thus the actual lemmas, are double-checked and corrected where necessary.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
7304	wakaleta		w		a		k		a	P	l		e		t	R	a													
7305	wakalima		w		a		k		a	P	l		i		m	R	a													
7306	wakamatwa		w		a	P	k		a		m		a		t	R	w		a											
7307	wakamilifu		w		a	P	k		a		m		i		l		i	R	f		u									
7308	wakamkuta		w		a		k		a		m	P	k		u		t	R	a											
7309	wakampigia		w		a		k		a		m	P	p		i		g	R	i		a									
7310	wakamtegemea		w		a		k		a		m	P	t		e		g		e		m		e	R	a					
7311	wakamuuliza		w		a		k		a		m		u	P	u		l		i		z	R	a							
7312	wakamwomba		w		a		k		a		m		w	P	o		m		b	R	a									
7313	wakandamizwaji		w		a	P	u		a		n		d		a		m		i		z	R	w		a		j		i	
7314	wakanishauri		w		a	P R S			a		n		i	P	s		h		a		u		r		i					
7315	wakanywa		w		a				a		n		y	R	w		a													
7316	wakaongea		w		a		k		a	P	o		n		g		e	R	a											

Figure 1: Excel sheet containing the material annotated by the annotator

4. The 'Memory-Based Swahili Morphological Analyzer' (MBSMA)

In this section, we describe our data-driven method for morphological analysis of Swahili, which is based on supervised machine learning. It reuses and refines the basic methodology coined in Van den Bosch and Daelemans (1999) which has been successfully applied to morphologically rich(er) languages such as Dutch (De Pauw et al. 2004) and Arabic (Van den Bosch et al. 2007). We use the data set described in Section 3 as our primary information source, and describe two systems.

4.1 Character-based morphological analysis

The first system directly ports the character-based approach of the original method (Van den Bosch and Daelemans 1999) to Swahili. The technique is

based on the machine-learning method of memory-based learning, which takes a database of *instances* as training material. These instances have to be represented as a fixed-length string of features, which describe the linguistic context of the token to be classified. Each instance is associated with a class, in this case a morphological category. The memory-based learning algorithm then stores this data in memory and classifies new, unseen instances, by comparing them to the ones in memory and extrapolating the class of the closest matching instance in memory.

For morphological processing, we extracted instances from the morphological database described in Section 3 as follows: for each character in each word form, we created a single instance that describes that character in its context. In the case of example (4) (uliko**[P]** + anz**[R]** + ia**[S]**), we can extract the ten instances displayed in Table 1. Each character is used once as a focus character (**F**) and associated with the five characters to its left (**L1→L5**) and the five characters to its right (**R1→R5**).

The *o* character in Instance 5, for example, is preceded by a dash (-) marking the word boundary and the characters *u*, *l*, *i* and *k*. It is followed by the characters *a*, *n*, *z*, *i*, *a*. This instance is then associated with a morphological classification, in this case **P**(refix), marking the fact that *o* is the last character of the prefix group. Similarly, the character *z* (Instance 8) is associated with the **R**(oot) class, and the *a* in Instance 10 with the **S**(uffix) class. Characters that do not mark the end of a morpheme are classed with the default category **0**.

The window size for the surrounding context is a parameter that needs to be optimized. Contexts that are too small or too large will hamper the performance of the classifier. The optimal window size of five characters before and after the focus characters was automatically established on the basis of comparative experiments on a development set.

Table 1: Character-based instances extracted from the morphological database

	L5	*L4*	*L3*	*L2*	*L1*	*F*	*R1*	*R2*	*R3*	*R4*	*R5*	*CLASS*
1	-	-	-	-	-	**u**	l	i	k	o	a	**0**
2	-	-	-	-	u	**l**	i	k	o	a	n	**0**
3	-	-	-	u	l	**i**	k	o	a	n	z	**0**
4	-	-	u	l	i	**k**	o	a	n	z	i	**0**
5	-	u	l	i	k	**o**	a	n	z	i	a	**P**
6	u	l	i	k	o	**a**	n	z	i	a	-	**0**
7	l	i	k	o	a	**n**	z	i	a	-	-	**0**
8	i	k	o	a	n	**z**	i	a	-	-	-	**R+a**
9	k	o	a	n	z	**i**	a	-	-	-	-	**0**
10	o	a	n	z	i	**a**	-	-	-	-	-	**S**

Furthermore, characters marked with an **R** classification can have an extra instruction, like in Instance 8 in Table 1, where the full class is **R+a**. The added **+a** instruction functions as an indication that the full lexical representation for this root morpheme needs to be *repaired* from the surface representation to the lexical representation by adding an *-a* to the end.

During actual morphological analysis, i.e. the morphological segmentation of previously unseen word forms, the words are similarly deconstructed and represented as instances using the same information. If, for example, we are to morphologically segment the previously unseen word *kulikoamuriwa* (kuliko**[P]** + amuriwa**[R]**), we classify the instances for each character. During the processing of this word, we will encounter the instance in Table 2, for which the morphological class is unknown.

Table 2: Instance to be classified

L5	***L4***	***L3***	***L2***	***L1***	***F***	***R1***	***R2***	***R3***	***R4***	***R5***	***CLASS***
k	u	l	i	k	o	a	m	u	r	i	??

This instance is compared to each and every instance in the training set, recorded by the memory-based learner. In doing so, the classifier will try to find that training instance in memory that most closely resembles it. For the instance in Table 2, this might be Instance 5 in Table 1, as they share six features (**L4**, **L3**, **L2**, **L1**, **F** and **R1**). The memory-based learner then extrapolates the **P** class of this training instance and *predicts* it to be the class of the new instance. Finally, in a post-processing phase, the words are recompiled and the predicted classes, i.e. morpheme boundaries, are inserted.

4.2 Syllable-based morphological analysis

The second version of the memory-based morphological analyzer moves away from the default level of the character and instead describes the problem on the level of the syllable. For Swahili, this has already been shown to constitute a relevant level of description (Elwell 2008).

Using syllables rather than characters as features does involve an extra pre-processing step, namely syllabification. We adopted the syllabification approach described in Ngugi, Okelo-Odongo and Wagacha (2005) and marked syllable boundaries for the words in our morphological database. This process is very precise for Swahili word forms, although mistakes are made on inflected loan words. Example (5) illustrates the syllabification process for the word *ulikoanzia*. Note that the syllable *zi* is never considered as a syllable within the representation, as it is split by a morpheme boundary that yields the desired bound root morpheme.

(5) uliko[P] + anz[R] + ia[S] → u | li | ko[P] + a | n | z[R] + i | a[S]

The rest of the processing remains the same. Instances are extracted in much the same way, except that the features now refer to syllables instead of single characters (see Table 3). Working on the syllable level also means fewer instances are being extracted, which helps speed up training times for the memory-based learner.

Table 3: Syllable-based instances extracted from the syllabified morphological database

	L5	L4	L3	L2	L1	F	R1	R2	R3	R4	R5	CLASS
1	-	-	-	-	-	**u**	li	ko	a	nz	i	0
2	-	-	-	-	u	**li**	ko	a	nz	i	a	0
3	-	-	-	u	li	**ko**	a	nz	i	a	-	P
4	-	-	u	li	ko	**a**	nz	i	a	-	-	0
5	-	u	li	ko	a	**nz**	i	a	-	-	-	R+a
6	u	li	ko	a	nz	**i**	a	-	-	-	-	0
7	li	ko	a	nz	i	**a**	-	-	-	-	-	S

5. Experiments and evaluation

In this section, we evaluate MBSMA. First, we look at its performance as an NLP tool *per se*, observing its accuracy as a morphological segmenter and lemmatizer. Next, we take a more qualitative look at the approach as a lexicographic tool.

5.1 Evaluation as an NLP tool

We are most interested in the accuracy of the morphological analyzer on previously unseen words: how well is the system able to morphologically segment and lemmatize unknown word forms? To investigate this, we perform *blind testing*, which involves partitioning the data in two parts: a 90% partition to train the system, and a 10% partition to evaluate it. For the latter, we use the manually annotated gold standard evaluation set, described in Section 3.

There are many experimental parameters to consider while building the system. The optimal combination of information source and algorithmic parameters can be established through thorough experimentation on the training set. At no point during this optimization process however, do we gauge the performance of the system on the evaluation set. This would not only produce artificially inflated accuracy scores in the final evaluation, but would also serve to *overfit* the system on one particular set of words.

We compare the accuracy of four different approaches to morphological segmentation and lemmatization of Swahili:

— **Morfessor** (Creutz et al. 2005): an unsupervised approach that takes a list of words (without annotation) and automatically induces a morphological model. This model is then used to segment the words in the evaluation set.

— **SALAMA^{x}** (Hurskainen 2004): the morphological analyzer used to lemmatize HCS. Since the SALAMA morphological analyzer itself is not publicly available, we reverse engineered its accuracy score by comparing the original annotation of the HCS annotations to the manually corrected annotation of the gold-standard evaluation set.

— **MBSMA-c**: the memory-based morphological analyzer working on the character level.
— **MBSMA-s**: the memory-based morphological analyzer working on the syllable level.

We will follow the standard approach of using word-error rate (WER) as our primary evaluation metric. It expresses the accuracy on the word-level, i.e. how many words have *not* been completely correctly segmented and lemmatized. In other words: the lower the WER, the better the system.

Table 4 displays the experimental results. As expected, **Morfessor** does not yield a great accuracy score. It is only able to completely correctly segment the surface representation of the words in the evaluation set 29.3% of the time. Morfessor is further hindered in its lemmatization accuracy, as it is unable to map surface representations onto lexical representations. It should be pointed out, however, that this is by far the most cost-effective system to develop, since it does not require any prior knowledge of the morphology of the language in question and thereby completely factors out the human element.

Table 4: Accuracy scores for Morfessor, SALAMA^x, MBSMA-c and MBSMA-s on the manually annotated evaluation set

	Segmentation of the surface representation	Further lemmatization
	WER	WER
Morfessor	70.7 %	73.6 %
SALAMA^x	11.7 %	12.0 %
MBSMA-c	13.3 %	13.6 %
MBSMA-s	11.6 %	11.7 %

We evaluated **SALAMA^x** by comparing the original HCS annotation to the manually annotated evaluation set. SALAMA^x obviously performs much better with a WER of 11.7%, an enormous error reduction over Morfessor. Most errors are made on the selection of the wrong lemma for a given word form, which further percolates into the segmentation.

The result for **MBSMA-c** shows that simply porting the original methodology described in Van den Bosch and Daelemans (1999) provides a functional data-driven morphological analyzer. However, this character-based approach is still significantly being outperformed by SALAMA^x.

Finally, when we move the level of description up to the syllable, **MBSMA-s**, the memory-based approach can be observed to slightly outperform SALAMA^x, establishing a small, but statistically significant reduction in WER on surface-level segmentation and a more substantial reduction for lemmatization.

This result may be surprising: how can a data-driven approach outperform the system that was used to create its information source? The answer to

this question lies in the generalization capabilities of the machine-learning technique. As previously mentioned and as further illustrated by the SALA-MA^{x} results in Table 4, quite a few erroneous analyses can be found in the annotation of HCS. Rather than completely mimicking the properties of the data the machine-learning approach uses to train its model, it implicitly generalizes over the data and filters out the noise.

A closer look at the output of MBSMA-s reveals some general tendencies. About half of the mistakes are made by MBSMA-s either by introducing a prefix group where there should not be one (36%) or by misjudging the length of the prefix group (18%). MBSMA-s fails to identify a prefix group 22% of the time and simply attaches it to the root. This also happens 18% of the time for the suffix group. Only very rarely is MBSMA-s unable to retrieve the right lemma from a correctly segmented surface representation, underlining its ability to restore a surface root form into its underlying lexical representation.

5.2 Evaluation as a lexicographic tool

Now that we have established the accuracy of the Swahili morphological analyzer on a purely quantitative basis, we turn to evaluating it as a lexicographic tool. In this discussion, we will focus on the lemmatization capabilities of the morphological analyzer and discuss processing times, retrievability and lemma discovery in the context of lexicography. We refer to two aspects of the lemmatizer as a lexicographic tool:

— The lemmatizer as a component in a CQP: we want the tool to be able to quickly and accurately lemmatize all the words in a corpus, so that example contexts for a given dictionary entry/lemma can easily be looked up and included in the description. We consider this to be an off-line task.

— The lemmatizer as a tool for digital dictionary consultation: a user should be able to input an inflected word form in the lookup interface. This word form is then lemmatized on the fly and its associated lemma is looked up in the dictionary. This is considered to be an online task.

Particularly the latter purpose requires the lemmatizer to be fast. **Processing times** are luckily quite favourable for MBSMA-s. On a standard Duo Core 2Ghz machine, the system is able to lemmatize over 70 words per second, using about 64Mb of internal memory. Speed and memory usage can be further optimized by using more efficient algorithmic parameters with only a minimal negative impact on its accuracy.

Apart from processing speed, we also identify another parameter to evaluate the lemmatizer as a lexicographic tool, namely **retrievability**, literally defined as its *ability* to *retrieve* the word forms for a given lemma. During dictionary compilation, we want to be able to provide the lexicographer with as many proper example sentences as possible for a given dictionary entry. The lemmatizer can be of great assistance in this task.

We can quantify the retrievability performance of a lemmatizer by running a controlled experiment: we group the word forms in the manually annotated evaluation set according to lemma. We consider only those lemmas that are associated with at least two distinct word forms in the evaluation set, like in example (6).

(6) **umba:** alituumba aliumba aliwaumba aliyemuumba

We then look at the output of the MBSMA-s lemmatizer and similarly group the word forms according to their predicted lemma, like in example (7).

(7) **umba:** aliumba aliwaumba aliyemuumba itayumba uliyumba

We notice three problems in (7): the word form *alituumba* which should have been in this list, is not there, while the word forms *itayumba* and *uliyumba* are erroneously associated with lemma *umba*. We can quantify the retrievability performance of the lemmatizer by calculating precision and recall:

— **Precision** counts the number of correct word forms retrieved by the lemmatizer and divides it by the total number of found lemmas. In example (7), precision would be 3/5 or 60%.

— **Recall** again counts the number of word forms correctly associated by the lemmatizer to the lemma in question and divides it by the number of lemmas that should have been retrieved, i.e. the number of word forms in example (6). For the predictions in example (7) the lemmatizer obtains a recall score of 3/4 or 75%.

If we perform this calculation for each lemma of the evaluation set and average the scores, we get some insight into the performance of the lemmatizer in terms of retrievability (see Table 5), and find that **MBSMA-s** compares favourably to **SALAMA[x]**. The precision score expresses that nine out of ten word forms provided by the lemmatizer are proper inflections of the lemma, while the recall score shows that the lemmatizer on average fails to retrieve only two out of ten word forms for a given lemma.

Table 5: Quantification of retrievability of the lemmatizer

	Precision	Recall
SALAMA[x]	89.4%	83.6%
MBSMA-s	92.4%	83.4%

A final aspect of the lemmatizer as a CQP tool relates to the **discovery** of new lemmas. Not only do we want the lemmatizer to relate word forms to existing lemmas, we also want it to be able to discover new lemmas not yet described in the dictionary.

To evaluate the lemmatizer from this perspective, we run another controlled experiment, again using the manually annotated evaluation set. We list all the lemmas in the evaluation set and subsequently remove all word forms associated with these lemmas in the training set. This means that the evaluation set solely consists of word forms for lemmas for which no linguistic evidence exists in the training set. After retraining the system on the new, disadvantaged training set, we can estimate the performance of MBSMA-s as a lemma discovery technique, by calculating how many of the lemmas in the evaluation set are still correctly found.

Table 6 displays the results for this experiment. We also provide extrapolated results from a similar experiment, using a probabilistic semi-supervised method (Lindén 2008). Note that this is not a direct comparison — since different evaluation techniques and data sets were used — and therefore only serves as a guideline to interpret the scores for MBSMA-s. The score expresses the percentage of *unknown* lemmas that have been correctly identified. The results for MBSMA-s are encouraging.

Table 6: Quantification of discovery capability of the lemmatizer

	Accuracy
Lindén 2008	68.2%
MBSMA-s	81.2%

6. Discussion and future work

To the best of our knowledge, the research results presented above describe the first attempt at building a comprehensive data-driven morphological analyzer for a Bantu language. It improves on previous rule-based approaches in terms of development time and accuracy, as well as in its ability to handle word forms for previously unseen lemmas.

We have demonstrated how this can be achieved with relatively little manual effort, and experimental results show that the method compares favourably to a meticulously designed rule-based technique, even when it is trained on the basis of its output. Defining the problem of data-driven morphological analysis on the syllable level, rather than on the character level, we furthermore showed how techniques typically designed with Indo-European language processing in mind, can be adjusted to work for Bantu languages as well.

The system shows promising results as a lexicographic tool: the lemmatizer yields encouraging results when considered as a corpus annotation tool and can therefore be considered as a useful addition to a CQP. Furthermore, the lemmatizer enables the discovery of previously unrecorded lemmas and can also function as a component in an interface for dictionary consultation.

The performance of the system can undoubtedly still be improved. The current system has been trained on an automatically annotated corpus of Swa-

hili. While the experimental results show that the machine-learning algorithm is to some extent able to filter out the noise in the data, we believe that cleaner training data can significantly improve the accuracy of the induced morphological analyzer.

During the development of the morphological analyzer, we have made some important and necessary abstractions. In the context of building a lemmatizer, it is not problematic to limit the system to recognize entire prefix and suffix groups. A true morphological analyzer should however also be able to segment and label the individual affixes. To this end, the construction of a large Swahili morphological database would be welcome, similar in scope to CELEX. We are confident that the morphological analyzer described in this article can significantly aid the construction of this data by providing a fast and accurate automatic pre-processor to the manual annotation.

Demonstration system and acknowledgements

A demonstration system for the MBSMA-s system can be found on the AfLaT website (http://aflat.org/?q=node/241).

Guy De Pauw is funded as a Postdoctoral Fellow of the Research Foundation – Flanders (FWO). Gilles-Maurice de Schryver would like to thank Ghent University for its continued support of his field trips to South Africa. Both authors would also like to thank Naomi Maajabu (http://aflat.org) for her annotation efforts.

Endnotes

1. For an online demo of the Swahili tagger, consult http://aflat.org/?q=node/10
2. For an online demo of the Northern Sotho tagger, consult http://aflat.org/?q=node/177

References

Anderson, W.A., P.M. Kotzé and A.E. Kotzé. 2007. *Application and Testing of Performance Enhancing Morphological Analysis Techniques*. Paper presented at the LSSA/SAALA/SAALT Joint Annual Conference, held at the North-West University, Potchefstroom, Republic of South Africa, 4–6 July 2007.

Baayen, R.H., R. Piepenbrock and L. Gulikers. 1995. *The Celex Lexical Database (Release2) [CD-ROM]*. Philadelphia: University of Pennsylvania, Linguistic Data Consortium.

Bosch, S.E., J. Jones, L. Pretorius and W.A. Anderson. 2006. Resource Development for South African Bantu Languages: Computational Morphological Analysers and Machine-Readable Lexicons. Roux, J. (Ed.). 2006. *Proceedings of the Workshop on Networking the Development of Language Resources for African Languages*, *LREC 2006*: 38-43. Genoa: ELRA.

Christ, O. 1994. A Modular and Flexible Architecture for an Integrated Corpus Query System. *Proceedings of COMPLEX'94: 3rd Conference on Computational Lexicography and Text Research*: 23-32. Budapest: Hungarian Academy of Sciences, Linguistics Institute.

Creutz, M., K. Lagus, K. Lindén and S. Virpioja. 2005. Morfessor and Hutmegs: Unsupervised Morpheme Segmentation for Highly-Inflecting and Compounding Languages. Langemets, M. and P. Penjam (Eds.). 2005. *Proceedings of the Second Baltic Conference on Human Language Technologies*: 107-112. Tallinn: Tallinn University of Technology.

De Pauw, G., G.-M. de Schryver and P.W. Wagacha. 2006. Data-driven Part-of-speech Tagging of Kiswahili. Sojka, P. et al. (Eds.). 2006. *Proceedings of Text, Speech and Dialogue, 9th International Conference*: 197-204. Berlin: Springer.

De Pauw, G., T. Laureys, W. Daelemans and H. van Hamme. 2004. A Comparison of Two Different Approaches to Morphological Analysis of Dutch. *Proceedings of the Seventh Meeting of the ACL Special Interest Group in Computational Phonology*: 62-69. Barcelona: ACL.

De Pauw, G. and P.W. Wagacha. 2007. Bootstrapping Morphological Analysis of Gĩkũyũ Using Unsupervised Maximum Entropy Learning. *Proceedings of the Eighth INTERSPEECH Conference, Antwerp, Belgium.*

De Pauw, G., P.W. Wagacha and D.A. Abade. 2007. Unsupervised Induction of Dholuo Word Classes using Maximum Entropy Learning. Getao, K. and E. Omwenga (Eds.). 2007. *Proceedings of the 1st International Conference in Computer Science and ICT*: 139-143. Nairobi: University of Nairobi.

De Schryver, G.-M. and G. De Pauw. 2007. Dictionary Writing System (DWS) + Corpus Query Package (CQP): The Case of *TshwaneLex*. *Lexikos* 17: 226-246.

Elwell, R. 2006. Finite State Methods for Bantu Verb Morphology. *Proceedings of the Texas Linguistics Society X, Austin.*

Elwell, R. 2008. Using Syllables as Features in Morpheme Tagging in Swahili. *Proceedings of the Fifth Midwest Computational Linguistics Colloquium, East Lansing.*

Gelbukh, A. (Ed.). 2008. *Computational Linguistics and Intelligent Text Processing: 9th International Conference, CICLing 2008 Proceedings*. Berlin: Springer.

Goldsmith, J. 2001. Unsupervised Learning of the Morphology of a Natural Language. *Compututational Linguistics* 27: 153-198.

Google. 2008. Google Search Engine [online]. http://google.com/.

Hillewaert, S., P. Joffe and G.-M. de Schryver. 2008. *Kamusi ya Kiswahili — Kiingereza Katika Mtandao/Online Swahili–English Dictionary* [online]. http://africanlanguages.com/swahili/.

Hurskainen, A. 1992. A Two-Level Computer Formalism for the Analysis of Bantu Morphology: An Application to Swahili. *Nordic Journal of African Studies* 1: 87-119.

Hurskainen, A. 1996. Disambiguation of Morphological Analysis in Bantu Languages. *Proceedings of the 16th Conference on Computational Linguistics*: 568-573. Copenhagen: ACL.

Hurskainen, A. 2004. Swahili Language Manager: A Storehouse for Developing Multiple Computational Applications. *Nordic Journal of African Studies* 13: 363-397.

Hurskainen, A. 2004a. HCS 2004 — Helsinki Corpus of Swahili. Compilers: Institute for Asian and African Studies (University of Helsinki) and CSC.

Hurskainen, A. and R. Halme. 2001. Mapping Between Disjoining and Conjoining Writing Systems in Bantu Languages: Implementation on Kwanyama. *Nordic Journal of African Studies* 10: 399-414.

Kamusi Project. 2008. The Internet Living Swahili Dictionary [online]. http://kamusiproject.org/.

Kilgarriff, A. et al. 2008. *Sketch Engine* [online]. http://sketchengine.co.uk/.

Koskenniemi, K. 1983. *Two-level Morphology: A General Computational Model for Word-form Recognition and Production*. Helsinki: University of Helsinki, Department of General Linguistics.

Kotzé, P.M. and W.A. Anderson. 2005. A Computational Morphological Analyser for Northern Sotho Deverbative Nouns: Applying Xerox Finite-state Software to Traditional Grammar. *South African Journal of African Languages* 25: 59-70.

Lindén, K. 2008. A Probabilistic Model for Guessing Base Forms of New Words by Analogy. Gelbukh, A. (Ed.). 2008: 106-116.

Maphosa, M. 2002. *Word Division and Orthography as Some of the Factors Posing Challenges in the Development of the Ndebele Grammatical Parser*. Paper presented at the Seventh International Conference of the African Association for Lexicography, organized by the Dictionary Unit of South African English, Rhodes University, Grahamstown, Republic of South Africa, 8–10 July 2002. For an abstact, see the AFRILEX 2002 conference brochure, 23-24.

Masaaki, N. 1999. A Japanese Morphological Analysis Method Using a Statistical Language Model and an *N*-best Search Algorithm. *Transactions of Information Processing Society of Japan* 40: 3420-3431.

Muhirwe, J. 2007. Computational Analysis of Kinyarwanda Morphology: The Morphological Alternations. Kizza, J.M. et al. (Eds.). 2007. *Special Topics in Computing and ICT Research: Strengthening the Role of ICT in Development*: 78-87. Kampala: Fountain Publishers.

Ngugi, K., W. Okelo-Odongo and P.W. Wagacha. 2005. Swahili Text-to-Speech System. *African Journal of Science and Technology* 6(1): 80-89.

Oflazer, K. 2008. Statistical Machine Translation into a Morphologically Complex Language. Gelbukh, A. (Ed.). 2008: 376-388.

Pretorius, L. and S.E. Bosch. 2007. Containing Overgeneration in Zulu Computational Morphology. Vetulani, Z. (Ed.). 2007. *Human Language Technologies as a Challenge for Computer Science and Linguistics*, *Proceedings of 3rd Language and Technology Conference*: 54-58. Poznan: Wydawnictwo Poznańskie Sp. z o.o.

Ridings, D. and W. Mavhu. 2002. *Problems and Challenges Encountered when Developing a Morphological Parser for the Shona Language*. Paper presented at the Seventh International Conference of the African Association for Lexicography, organized by the Dictionary Unit of South African English, Rhodes University, Grahamstown, Republic of South Africa, 8–10 July 2002. For an abstact, see the AFRILEX 2002 conference brochure, 24-26.

Taljard, E. and S.E. Bosch. 2005. A Comparison of Approaches Towards Word Class Tagging: Disjunctively vs Conjunctively Written Bantu Languages. *Proceedings of the Conference on Lesser Used Languages and Computer Linguistics*, *Bolzano*.

Van den Bosch, A. and W. Daelemans. 1999. Memory-based Morphological Analysis. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*: 285-292. Maryland: ACL.

Van den Bosch, A., E. Marsi and A. Soudi. 2007. Memory-based Morphological Analysis and Part-of-speech Tagging of Arabic. Soudi, A. et al. (Eds.). 2007. *Arabic Computational Morphology: Knowledge-based and Empirical Methods*: 203-219. Berlin: Springer.