EACL 2009

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages

AfLaT 2009

31 March 2009
Megaron Athens International Conference Centre
Athens, Greece

Production and Manufacturing by
TEHNOGRAFIA DIGITAL PRESS
7 Ektoros Street
152 35 Vrilissia
Athens, Greece

This workshop is sponsored by

ISBN 1-932432-25-6

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

Preface

In multilingual situations, language technologies are crucial for providing access to information and opportunities for economic development. With somewhere between 1,000 and 2,000 different languages, Africa is a multilingual continent *par excellence* and presents acute challenges for those seeking to promote and use African languages in the areas of business development, education, research, and relief aid.

In recent times a number of researchers and institutions, both from Africa and elsewhere, have come forward to share the common goal of developing capabilities in language technologies for the African languages. The goal of the workshop, then, is to provide a forum to meet and share the latest developments in this field. It also seeks to attract linguists who specialize in African languages and would like to leverage the tools and approaches of computational linguistics, as well as computational linguists who are interested in learning about the particular linguistic challenges posed by the African languages.

The workshop consists of an invited talk on African language families and their structural properties by Prof. Sonja Bosch (UNISA, South Africa), followed by refereed research papers in computational linguistics. The call for papers specified that the focus would be on the less-commonly studied and lesser-resourced languages, such as those of sub-Saharan Africa. These could include languages from all four families: Niger-Congo, Nilo-Saharan, Khoisan and Afro-Asiatic, with the exception of Arabic (which is covered by the 'Computational Approaches to Semitic Languages' workshop). Variants of European languages such as African French, African English or Afrikaans were also excluded from the call. The call was well answered, with 24 proposals being submitted. Following a rigorous review process, nine were withheld as full papers, and a further seven as poster presentations. These proceedings contain the double-blind, peer reviewed texts of the contributions that were withheld.

We would like to thank the US Army RDECOM International Technology Center - Atlantic for generously sponsoring this workshop.

We wish you a very enjoyable time at AfLaT 2009!

— The Organizers.

AfLaT 2009 Organizers

Proceedings Editors:

Guy De Pauw
Gilles-Maurice de Schryver
Lori Levin

Organizers:

Lori Levin, Language Technologies Institute, Carnegie Mellon University (USA)
Guy De Pauw, CNTS - Language Technology Group, University of Antwerp (Belgium) - School of Computing and Informatics, University of Nairobi (Kenya) - AfLaT.org
John Kiango, Institute of Kiswahili Research, University of Dar Es Salaam (Tanzania)
Judith Klavans, Institute for Advanced Computer Studies, University of Maryland (USA)
Manuela Noske, Microsoft Corporation (USA)
Gilles-Maurice de Schryver, African Languages and Cultures, Ghent University (Belgium) - Xhosa Department, University of the Western Cape (South Africa) - AfLaT.org
Peter Waiganjo Wagacha, School of Computing and Informatics, University of Nairobi (Kenya) - AfLaT.org

Program Committee:

Alan Black, Carnegie Mellon University (USA)
Sonja Bosch, University of South Africa (South Africa)
Christopher Cieri, University of Pennsylvania, Linguistic Data Consortium (USA)
Guy De Pauw, University of Antwerp (Belgium)
Gilles-Maurice de Schryver, Ghent University (Belgium)
Robert Frederking, Carnegie Mellon University (USA)
Dafydd Gibbon, University of Bielefeld (Germany)
Jeff Good, SUNY Buffalo (USA)
Mike Gasser, Indiana University (USA)
Gregory Iverson, University of Maryland, Center for Advanced Study of Language (USA)
Stephen Larocca, US Army Research Lab (USA)
Michael Maxwell, University of Maryland, Center for Advanced Study of Language (USA)
Manuela Noske, Microsoft Corporation (USA)
Tristan Purvis, University of Maryland, Center for Advanced Study of Language (USA)
Clare Voss, US Army Research Lab (USA)
Peter Waiganjo Wagacha, University of Nairobi (Kenya)
Briony Williams, University of Wales, Bangor (Great Britain)

Invited Speaker:

Sonja Bosch, Department of African Languages, University of South Africa (South Africa)

Table of Contents

Conference Program

Tuesday, March 31, 2009

09:00–10:30 Invited Talk: ”African Language Families and their Structural Properties” by Sonja Bosch

10:30–11:00 Coffee Break

Session 1: Corpora (11:00–12:30)

11:00–11:30 *Collecting and Evaluating Speech Recognition Corpora for Nine Southern Bantu Languages*
Jaco Badenhorst, Charl Van Heerden, Marelie Davel and Etienne Barnard

11:30–12:00 *The SAWA Corpus: A Parallel Corpus English - Swahili*
Guy De Pauw, Peter Waiganjo Wagacha and Gilles-Maurice de Schryver

12:00–12:30 *Information Structure in African Languages: Corpora and Tools*
Christian Chiarcos, Ines Fiedler, Mira Grubic, Andreas Haida, Katharina Hartmann, Julia Ritz, Anne Schwarz, Amir Zeldes and Malte Zimmermann

12:30–14:00 Lunch Break

Session 2: Morphology, Speech and Part-of-Speech Tagging (14:00–16:00)

14:00–14:30 *A Computational Approach to Yorùbá Morphology*
Raphael Finkel and Odetunji Ajadi Odejobi

14:30–15:00 *Using Technology Transfer to Advance Automatic Lemmatisation for Setswana*
Hendrik Johannes Groenewald

15:00–15:30 *Part-of-Speech Tagging of Northern Sotho: Disambiguating Polysemous Function Words*
Gertrud Faaß, Ulrich Heid, Elsabé Taljard and Danie Prinsloo

15:30–16:00 *Development of an Amharic Text-to-Speech System Using Cepstral Method*
Tadesse Anberbir and Tomio Takara

16:00–16:30 Coffee Break

Tuesday, March 31, 2009 (continued)

Session 3: General Papers and Discussion (16:30–18:00)

16:30–17:00 *Building Capacities in Human Language Technology for African Languages*
Tunde Adegbola

17:00–17:30 *Initial Fieldwork for LWAZI: A Telephone-Based Spoken Dialog System for Rural South Africa*
Tebogo Gumede and Madelaine Plauché

17:30–18:00 Discussion

Poster Session (During Coffee and Lunch Breaks)

Setswana Tokenisation and Computational Verb Morphology: Facing the Challenge of a Disjunctive Orthography
Rigardt Pretorius, Ansu Berg, Laurette Pretorius and Biffie Viljoen

Interlinear Glossing and its Role in Theoretical and Descriptive Studies of African and other Lesser–Documented Languages
Dorothee Beermann and Pavel Mihaylov

Towards an Electronic Dictionary of Tamajaq Language in Niger
Chantal Enguehard and Issouf Modi

A Repository of Free Lexical Resources for African Languages: The Project and the Method
Piotr Bański and Beata Wójtowicz

Exploiting Cross-Linguistic Similarities in Zulu and Xhosa Computational Morphology
Laurette Pretorius and Sonja Bosch

Methods for Amharic Part-of-Speech Tagging
Björn Gambäck, Fredrik Olsson, Atelach Alemu Argaw and Lars Asker

An Ontology for Accessing Transcription Systems (OATS)
Steven Moran

Collecting and evaluating speech recognition corpora for nine Southern Bantu languages

Jaco Badenhorst, Charl van Heerden, Marelie Davel and Etienne Barnard
HLT Research Group, Meraka Institute, CSIR, South Africa
jbadenhorst@csir.co.za, mdavel@csir.co.za
cvheerden@csir.co.za, ebarnard@csir.co.za

Abstract

We describe the Lwazi corpus for automatic speech recognition (ASR), a new telephone speech corpus which includes data from nine Southern Bantu languages. Because of practical constraints, the amount of speech per language is relatively small compared to major corpora in world languages, and we report on our investigation of the stability of the ASR models derived from the corpus. We also report on phoneme distance measures across languages, and describe initial phone recognisers that were developed using this data.

1 Introduction

There is a widespread belief that spoken dialog systems (SDSs) will have a significant impact in the developing countries of Africa (Tucker and Shalonova, 2004), where the availability of alternative information sources is often low. Traditional computer infrastructure is scarce in Africa, but telephone networks (especially cellular networks) are spreading rapidly. In addition, speech-based access to information may empower illiterate or semi-literate people, 98% of whom live in the developing world.

Spoken dialog systems can play a useful role in a wide range of applications. Of particular importance in Africa are applications such as education, using speech-enabled learning software or kiosks and information dissemination through media such as telephone-based information systems. Significant benefits can be envisioned if information is provided in domains such as agriculture (Nasfors, 2007), health care (Sherwani et al., ; Sharma et al., 2009) and government services (Barnard et al., 2003). In order to make SDSs a reality in Africa, technology components such as text-to-speech (TTS) systems and automatic speech recognition (ASR) systems are required. The latter category of technologies is the focus of the current contribution.

Speech recognition systems exist for only a handful of African languages (Roux et al., ; Seid and Gambck, 2005; Abdillahi et al., 2006), and to our knowledge no service available to the general public currently uses ASR in an indigenous African language. A significant reason for this state of affairs is the lack of sufficient linguistic resources in the African languages. Most importantly, modern speech recognition systems use statistical models which are trained on corpora of relevant speech (i.e. appropriate for the recognition task in terms of the language used, the profile of the speakers, speaking style, etc.) This speech generally needs to be curated and transcribed prior to the development of ASR systems, and for most applications speech from a large number of speakers is required in order to achieve acceptable system performance. On the African continent, where infrastructure such as computer networks is less developed than in countries such as America, Japan and the European countries, the development of such speech corpora is a significant hurdle to the development of ASR systems.

The complexity of speech corpus development is strongly correlated with the amount of data that is required, since the number of speakers that need to be canvassed and the amount of speech that must be curated and transcribed are major factors in determining the feasibility of such development. In order to minimise this complexity, it is important to have tools and guidelines that can be used to assist in designing the smallest corpora that will be sufficient for typical applications of ASR systems. As minimal corpora can be extended by sharing data across languages, tools are also required to indicate when data sharing will be beneficial and when detrimental.

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 1–8,
Athens, Greece, 31 March 2009.

In this paper we describe and evaluate a new speech corpus of South African languages currently under development (the Lwazi corpus) and evaluate the extent in which computational analysis tools can provide further guidelines for ASR corpus design in resource-scarce languages.

2 Project Lwazi

The goal of Project Lwazi is to provide South African citizens with information and information services in their home language, over the telephone, in an efficient and affordable manner. Commissioned by the South African Department of Arts and Culture, the activities of this three year project (2006-2009) include the development of core language technology resources and components for all the official languages of South Africa, where, for the majority of these, no prior language technology components were available.

The core linguistic resources being developed include phoneme sets, electronic pronunciation dictionaries and the speech and text corpora required to develop automated speech recognition (ASR) and text-to-speech (TTS) systems for all eleven official languages of South Africa. The usability of these resources will be demonstrated during a national pilot planned for the third quarter of 2009. All outputs from the project are being released as open source software and open content(Meraka-Institute, 2009).

Resources are being developed for all nine Southern Bantu languages that are recognised as official languages in South Africa (SA). These languages are: (1) isiZulu (zul[1]) and isiXhosa (xho), the two Nguni languages most widely spoken in SA. Together these form the home language of 41% of the SA population. (2) The three Sotho languages: Sepedi (nso), Setswana (tsn), Sesotho (sot), together the home language of 26% of the SA population. (3) The two Nguni languages less widely spoken in SA: siSwati (ssw) and isiNdebele (nbl), together the home language of 4% of the SA population. (4) Xitsonga (tso) and Tshivenda (ven), the home languages of 4% and 2% of the SA population, respectively (Lehohla, 2003). (The other two official languages of South Africa are Germanic languages, namely English (eng) and Afrikaans (afr).)

For all these languages, new pronunciation dictionaries, text and speech corpora are being developed. ASR speech corpora consist of approximately 200 speakers per language, producing read and elicited speech, recorded over a telephone channel. Each speaker produced approximately 30 utterances, 16 of these were randomly selected from a phonetically balanced corpus and the remainder consist of short words and phrases: answers to open questions, answers to yes/no questions, spelt words, dates and numbers. The speaker population was selected to provide a balanced profile with regard to age, gender and type of telephone (cellphone or landline).

3 Related work

Below, we review earlier work relevant to the development of speech recognisers for languages with limited resources. This includes both ASR system design (Sec. 3.1) and ASR corpus design (Sec. 3.2). In Sec. 3.3, we also review the analytical tools that we utilise in order to investigate corpus design systematically.

3.1 ASR for resource-scarce languages

The main linguistic resources required when developing ASR systems for telephone based systems are electronic pronunciation dictionaries, annotated audio corpora (used to construct acoustic models) and recognition grammars. An ASR audio corpus consists of recordings from multiple speakers, with each utterance carefully transcribed orthographically and markers used to indicate non-speech and other events important from an ASR perspective. Both the collection of appropriate speech from multiple speakers and the accurate annotation of this speech are resource-intensive processes, and therefore corpora for resource-scarce languages tend to be very small (1 to 10 hours of audio) when compared to the speech corpora used to build commercial systems for world languages (hundreds to thousands of hours per language).

Different approaches have been used to best utilise limited audio resources when developing ASR systems. Bootstrapping has been shown to be a very efficient technique for the rapid development of pronunciation dictionaries, even when utilising linguistic assistants with limited phonetic training (Davel and Barnard, 2004).

Small audio corpora can be used efficiently by utilising techniques that share data across lan-

[1] After each language name, the ISO 639-3:2007 language code is provided in brackets.

guages, either by developing multilingual ASR systems (a single system that simultaneously recognises different languages), or by using additional source data to supplement the training data that exists in the target language. Various data sharing techniques for language-dependant acoustic modelling have been studied, including cross-language transfer, data pooling, language adaptation and bootstrapping (Wheatley et al., 1994; Schultz and Waibel, 2001; Byrne et al., 2000). Both (Wheatley et al., 1994) and (Schultz and Waibel, 2001) found that useful gains could be obtained by sharing data across languages with the size of the benefit dependent on the similarity of the sound systems of the languages combined. In the only cross-lingual adaptation study using African languages (Niesler, 2007), similar gains have not yet been observed.

3.2 ASR corpus design

Corpus design techniques for ASR are generally aimed at specifying or selecting the most appropriate subset of data from a larger domain in order to optimise recognition accuracy, often while explicitly minimising the size of the selected corpus. This is achieved through various techniques that aim to include as much variability in the data as possible, while simultaneously ensuring that the corpus matches the intended operating environment as accurately as possible.

Three directions are primarily employed: (1) explicit specification of phonotactic, speaker and channel variability during corpus development, (2) automated selection of informative subsets of data from larger corpora, with the smaller subset yielding comparable results, and (3) the use of active learning to optimise existing speech recognition systems. All three techniques provide a perspective on the sources of variation inherent in a speech corpus, and the effect of this variation on speech recognition accuracy.

In (Nagroski et al., 2003), Principle Component Analysis (PCA) is used to cluster data acoustically. These clusters then serve as a starting point for selecting the optimal utterances from a training database. As a consequence of the clustering technique, it is possible to characterise some of the acoustic properties of the data being analysed, and to obtain an understanding of the major sources of variation, such as different speakers and genders (Riccardi and Hakkani-Tur, 2003).

Active and unsupervised learning methods can be combined to circumvent the need for transcribing massive amounts of data (Riccardi and Hakkani-Tur, 2003). The most informative untranscribed data is selected for a human to label, based on acoustic evidence of a partially and iteratively trained ASR system. From such work, it soon becomes evident that the optimisation of the amount of variation inherent to training data is needed, since randomly selected additional data does not necessarily improve recognition accuracy. By focusing on the selection (based on existing transcriptions) of a uniform distribution across different speech units such as words and phonemes, improvements are obtained (Wu et al., 2007).

In our focus on resource-scarce languages, the main aim is to understand the amount of data that needs to be collected in order to achieve acceptable accuracy. This is achieved through the use of analytic measures of data variability, which we describe next.

3.3 Evaluating phoneme stability

In (Badenhorst and Davel, 2008) a technique is developed that estimates how stable a specific phoneme is, given a specific set of training data. This statistical measure provides an indication of the effect that additional training data will have on recognition accuracy: the higher the stability, the less the benefit of additional speech data.

The model stability measure utilises the Bhattacharyya bound (Fukunaga, 1990), a widely-used upper bound of the Bayes error. If P_i and $p_i(X)$ denote the prior probability and class-conditional density function for class i, respectively, the Bhattacharyya bound ϵ is calculated as:

$$\epsilon = \sqrt{P_1 P_2} \int \sqrt{p_1(X) p_2(X)} dX \quad (1)$$

When both density functions are Gaussian with mean μ_i and covariance matrix Σ_i, integration of ϵ leads to a closed-form expression for ϵ:

$$\epsilon = \sqrt{P_1 P_2} e^{-\mu(1/2)} \quad (2)$$

where

$$\mu(1/2) = \frac{1}{8}(\mu_2 - \mu_1)^T \Big[\frac{\Sigma_1 + \Sigma_2}{2}\Big]^{-1} (\mu_2 - \mu_1) + \frac{1}{2} ln \frac{|\frac{\Sigma_1 + \Sigma_2}{2}|}{\sqrt{|\Sigma_1||\Sigma_2|}} \quad (3)$$

is referred to as the Bhattacharyya distance.

In order to estimate the stability of an acoustic model, the training data for that model is separated into a number of disjoint subsets. All subsets are selected to be mutually exclusive with respect to the speakers they contain. For each subset, a separate acoustic model is trained, and the Bhattacharyya bound between each pair of models calculated. By calculating both the mean of this bound and the standard deviation of this measure across the various model pairs, a statistically sound measure of model estimation stability is obtained.

4 Computational analysis of the Lwazi corpus

We now report on our analysis of the Lwazi speech corpus, using the stability measure described above. Here, we focus on four languages (isiNdebele, siSwati, isiZulu and Tshivenda) for reasons of space; later, we shall see that the other languages behave quite similarly.

4.1 Experimental design

For each phoneme in each of our target languages, we extract all the phoneme occurrences from the 150 speakers with the most utterances per phoneme. We utilise the technique described in Sec. 3.3 to estimate the Bhattacharyya bound both when evaluating phoneme variability and model distance. In both cases we separate the data for each phoneme into 5 disjoint subsets. We calculate the mean of the 10 distances obtained between the various intra-phoneme model pairs when measuring phoneme stability, and the mean of the 25 distances obtained between the various inter-phoneme model pairs when measuring phoneme distance.

In order to be able to control the number of phoneme observations used to train our acoustic models, we first train a speech recognition system and then use forced alignment to label all of the utterances using the systems described in Sec. 5. Mel-frequency cepstral coefficients (MFCCs) with cepstral mean and variance normalisation are used as features, as described in Sec. 5.

4.2 Analysis of phoneme variability

In an earlier analysis of phoneme variability of an English corpus (Badenhorst and Davel, 2008), it was observed that similar trends are observed when utilising different numbers of mixtures in a Gaussian mixture model. For both context dependent and context independent models similar trends are also observed. (Asymptotes occur later, but trends remain similar.) Because of the limited size of the Lwazi corpus, we therefore only report on single-mixture context-independent models in the current section.

As we also observe similar trends for phonemes within the same broad categories, we report on one or two examples from several broad categories which occur in most of our target languages. Using SAMPA notation, the following phonemes are selected: /a/ (vowels), /m/ (nasals), /b/ and /g/ (voiced plosives) and /s/ (unvoiced fricatives), after verifying that these phonemes are indeed representative of the larger groups.

Figures 1 and 2 demonstrate the effects of variable numbers of phonemes and speakers, respectively, on the value of the mean Bhattacharyya bound. This value should approach 0.5 for a model fully trained on a sufficiently representative set of data. In Fig. 1 we see that the various broad categories of sounds approach the asymptotic bound in different ways. The vowels and nasals require the largest number of phoneme occurrences to reach a given level, whereas the fricatives and plosives converge quite rapidly (With 10 observations per speaker, both the fricatives and plosives achieve values of 0.48 or better for all languages, in contrast to the vowels and nasals which require 30 observations to reach similar stability). Note that we employed 30 speakers per phoneme group, since that is the largest number achievable with our protocol.

For the results in Fig. 2, we keep the number of phoneme occurrences per speaker fixed at 20 (this ensures that we have sufficient data for all phonemes, and corresponds with reasonable convergence in Fig. 1). It is clear that additional speakers would still improve the modelling accuracy for especially the vowels and nasals. We observe that the voiced plosives and fricatives quickly achieve high values for the bound (close to the ideal 0.5).

Figures 1 and 2 – as well as similar figures for the other phoneme classes and languages we have studied – suggest that all phoneme categories require at least 20 training speakers to achieve reasonable levels of convergence (bound levels of 0.48 or better). The number of phoneme observations required per speaker is more variable, rang-

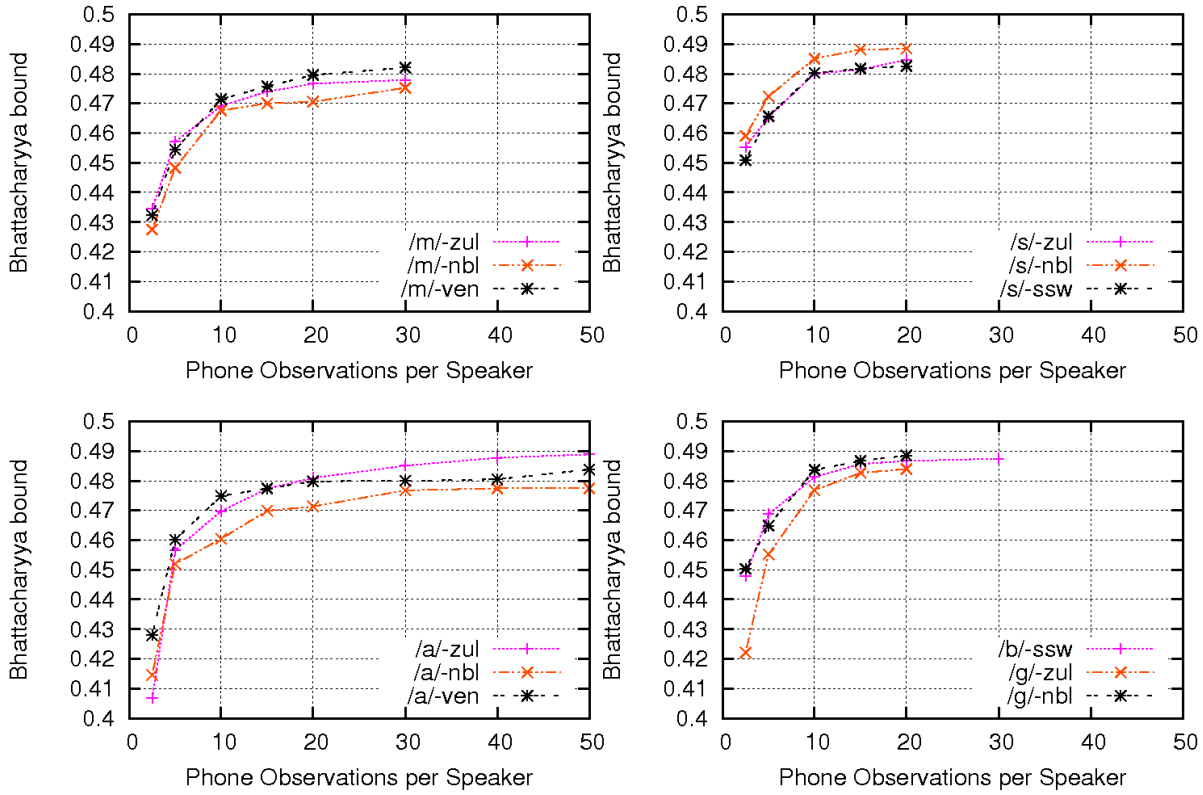


Figure 1: *Effect of number of phoneme utterances per speaker on mean of Bhattacharyya bound for different phoneme groups using data from 30 speakers*

ing from less than 10 for the voiceless fricatives to 30 or more for vowels, liquids and nasals. We return to these observations below.

4.3 Distances between languages

In Sec. 3.1 it was pointed out that the similarities between the same phonemes in different languages are important predictors of the benefit achievable from pooling the data from those languages. Armed with the knowledge that stable models can be estimated with 30 speakers per phoneme and between 10 and 30 phonemes occurrences per speaker, we now turn to the task of measuring distances between phonemes in various languages.

We again use the mean Bhattacharyya bound to compare phonemes, and obtain values between all possible combinations of phonemes. Results are shown for the isiNdebele phonemes /n/ and /a/ in Fig. 3. As expected, similar phonemes from the different languages are closer to one another than different phonemes of the same language. However, the details of the distances are quite revealing: for /a/, siSwati is closest to the isiNdebele model, as would be expected given their close linguistic relationship, but for /n/, the Tshivenda model is found to be closer than either of the other Nguni languages. For comparative purposes, we have included one non-Bantu language (Afrikaans), and we see that its models are indeed significantly more dissimilar from the isiNdebele model than any of the Bantu languages. In fact, the Afrikaans /n/ is about as distant from isiNdebele /n/ as isiNdebele and isiZulu /l/ are!

5 Initial ASR results

In order to verify the usability of the Lwazi corpus for speech recognition, we develop initial ASR systems for all 11 official South African languages. A summary of the data statistics for the Bantu languages investigated is shown in Tab. 1, and recognition accuracies achieved are summarised in Tab. 2. For these tests, data from 30 speakers per language were used as test data, with the remaining data being used for training.

Although the Southern Bantu languages are tone languages, our systems do not encode tonal

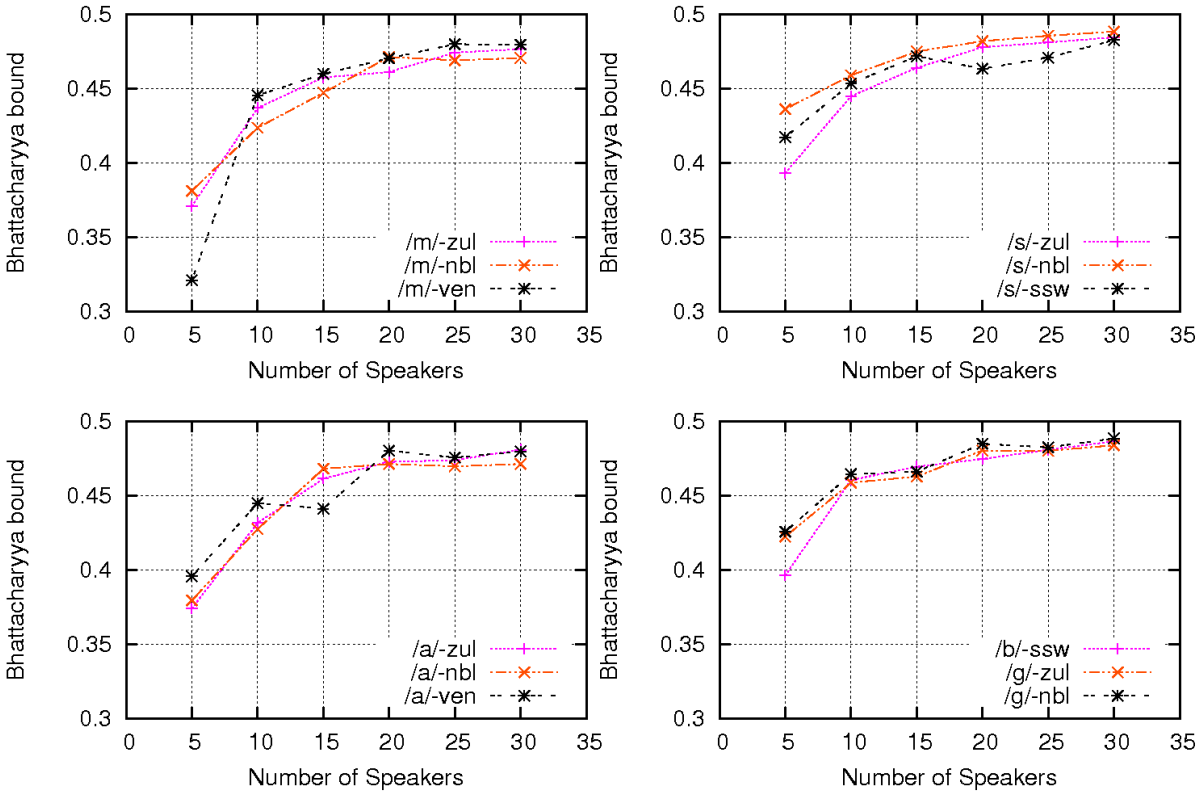


Figure 2: *Effect of number of speakers on mean of Bhattacharyya bound for different phoneme groups using 20 utterances per speaker*

Language	total # minutes	# speech minutes	# distinct phonemes
isiNdebele	564	465	46
isiXhosa	470	370	52
isiZulu	525	407	46
Tshivenda	354	286	38
Sepedi	394	301	45
Sesotho	387	313	44
Setswana	379	295	34
siSwati	603	479	39
Xitsonga	378	316	54
N-TIMIT	315	-	39

Table 1: *A summary of the Lwazi ASR corpus: Bantu languages.*

information, since tone is unlikely to be important for small-to-medium vocabulary applications (Zerbian and Barnard, 2008).

As the initial pronunciation dictionaries were developed to provide good coverage of the language in general, these dictionaries did not cover the entire ASR corpus. Grapheme-to-phoneme rules are therefore extracted from the general dictionaries using the Default&Refine algorithm (Davel and Barnard, 2008) and used to generate missing pronunciations.

We use HTK 3.4 to build a context-dependent cross-word HMM-based phoneme recogniser with triphone models. Each model had 3 emitting states with 7 mixtures per state. 39 features are used: 13 MFCCs together with their first and second order derivatives. Cepstral Mean Normalisation (CMN) as well as Cepstral Variance Normalisation (CMV) are used to perform speaker-independent normalisation. A diagonal covariance matrix is used; to partially compensate for this incorrect assumption of feature independence semi-tied transforms are applied. A flat phone-based language model is employed throughout.

As a rough benchmark of acceptable phoneme-recognition accuracy, recently reported results obtained by (Morales et al., 2008) on a similar-sized telephone corpus in American English (N-TIMIT) are also shown in Tab. 2. We see that the Lwazi results compare very well with this benchmark.

An important issue in ASR corpus design is

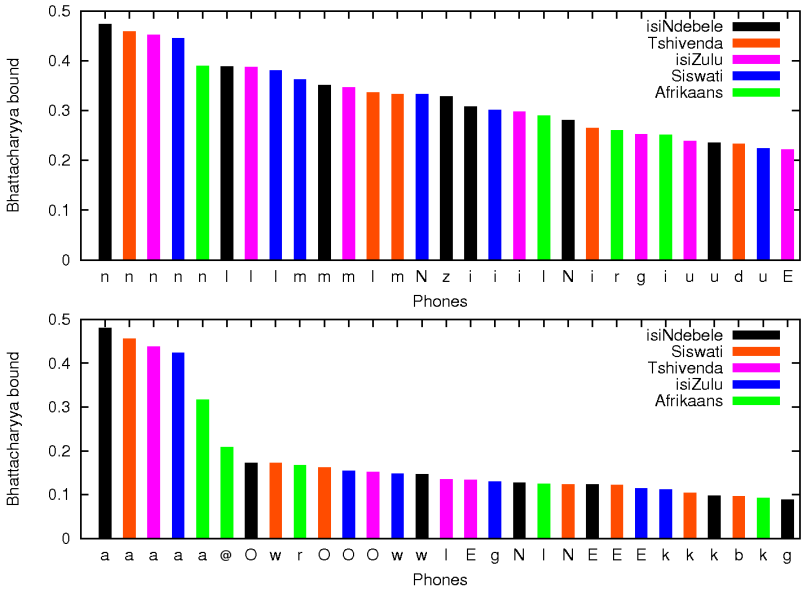


Figure 3: *Effective distances in terms of the mean of the Bhattacharyya bound between a single phoneme (/n/-nbl top and /a/-nbl bottom) and each of its closest matches within the set of phonemes investigated.*

Language	% corr	% acc	avg # phons	total # speakers
isiNdebele	74.21	65.41	28.66	200
isiXhosa	69.25	57.24	17.79	210
isiZulu	71.18	60.95	23.42	201
Tshivenda	76.37	66.78	19.53	201
Sepedi	66.44	55.19	16.45	199
Sesotho	68.17	54.79	18.57	200
Setswana	69.00	56.19	20.85	207
siSwati	74.19	64.46	30.66	208
Xitsonga	70.32	59.41	14.35	199
N-TIMIT	64.07	55.73	-	-

Table 2: *Initial results for South African ASR systems. The column labelled "avg # phonemes" lists the average number of phoneme occurrences for each phoneme for each speaker.*

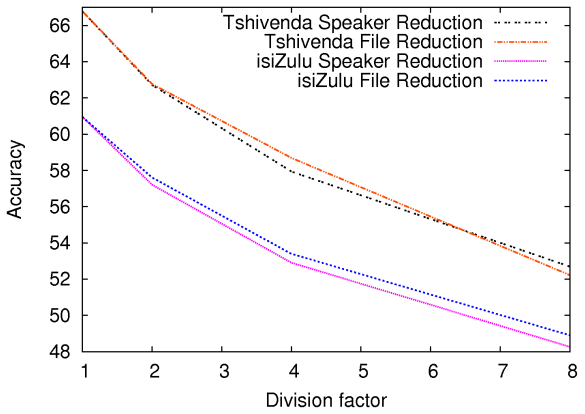


Figure 4: *The influence of a reduction in training corpus size on phone recognition accuracy.*

the trade-off between the number of speakers and the amount of data per speaker (Wheatley et al., 1994). The figures in Sec. 4.2 are not conclusive on this trade-off, so we have also investigated the effect of reducing either the number of speakers or the amount of data per speaker when training the isiZulu and Tshivenda recognisers. As shown in Fig. 4, the impact of both forms of reduction is comparable across languages and different degrees of reduction, in agreement with the results of Sec. 4.2.

These results indicate that we now have a firm baseline to investigate data-efficient training methods such as those described in Sec. 3.1.

6 Conclusion

In this paper we have introduced a new telephone speech corpus which contains data from nine Southern Bantu languages. Our stability analysis shows that the speaker variety as well as the amount of speech per speaker is sufficient to achieve acceptable model stability, and this conclusion is confirmed by the successful training of phone recognisers in all the languages. We confirm the observation in (Badenhorst and Davel, 2008) that different phone classes have different

data requirements, but even for the more demanding classes (vowels, nasals, liquids) our amount of data seems sufficient. Our results suggest that similar accuracies may be achievable by using more speech from fewer speakers – a finding that may be useful for the further development of speech corpora in resource-scarce languages.

Based on the proven stability of our models, we have performed some preliminary measurements of the distances between the phones in the different languages; such distance measurements are likely to be important for the sharing of data across languages in order to further improve ASR accuracy. The development of real-world applications using this data is currently an active topic of research; for that purpose, we are continuing to investigate additional methods to improve recognition accuracy with such relatively small corpora, including cross-language data sharing and efficient adaptation methods.

References

Nimaan Abdillahi, Pascal Nocera, and Jean-Franois Bonastre. 2006. Automatic transcription of Somali language. In *Interspeech*, pages 289–292, Pittsburgh, PA.

J.A.C. Badenhorst and M.H. Davel. 2008. Data requirements for speaker independent acoustic models. In *PRASA*, pages 147–152.

E. Barnard, L. Cloete, and H. Patel. 2003. Language and technology literacy barriers to accessing government services. *Lecture Notes in Computer Science*, 2739:37–42.

W. Byrne, P. Beyerlein, J. M. Huerta, S. Khudanpur, B. Marthi, J. Morgan, N. Peterek, J. Picone, D. Vergyri1, and W. Wang. 2000. Towards language independent acoustic modeling. In *ICASSP*, volume 2, pages 1029–1032, Istanbul, Turkey.

M. Davel and E. Barnard. 2004. The efficient creation of pronunication dictionaries: human factors in bootstrapping. In *Interspeech*, pages 2797–2800, Jeju, Korea, Oct.

M. Davel and E. Barnard. 2008. Pronunciation predication with Default&Refine. *Computer Speech and Language*, 22:374–393, Oct.

K. Fukunaga. 1990. *Introduction to Statistical Pattern Recognition*. Academic Press, Inc., 2nd edition.

Pali Lehohla. 2003. *Census 2001: Census in brief*. Statistics South Africa.

Meraka-Institute. 2009. Lwazi ASR corpus. Online: http://www.meraka.org.za/lwazi.

N. Morales, J. Tejedor, J. Garrido, J. Colas, and D.T. Toledano. 2008. STC-TIMIT: Generation of a single-channel telephone corpus. In *LREC*, pages 391–395, Marrakech, Morocco.

A. Nagroski, L. Boves, and H. Steeneken. 2003. In search of optimal data selection for training of automatic speech recognition systems. *ASRU workshop*, pages 67–72, Nov.

P. Nasfors. 2007. Efficient voice information services for developing countries. Master's thesis, Department of Information Technology, Uppsala University.

T. Niesler. 2007. Language-dependent state clustering for multilingual acoustic modeling. *Speech Communication*, 49:453–463.

G. Riccardi and D. Hakkani-Tur. 2003. Active and unsupervised learning for automatic speech recognition. In *Eurospeech*, pages 1825–1828, Geneva, Switzerland.

J.C. Roux, E.C. Botha, and J.A. du Preez. Developing a multilingual telephone based information system in african languages. In *LREC*, pages 975–980, Athens, Greece.

T. Schultz and A. Waibel. 2001. Language-independent and language-adaptive acoustic modeling for speech recognition. *Speech Communication*, 35:31–51, Aug.

Hussien Seid and Bjrn Gambck. 2005. A speaker independent continuous speech recognizer for Amharic. In *Interspeech*, pages 3349–3352, Lisboa, Portugal, Oct.

A. Sharma, M. Plauche, C. Kuun, and E. Barnard. 2009. HIV health information access using spoken dialogue systems: Touchtone vs. speech. Accepted at IEEE Int. Conf. on ICTD.

J. Sherwani, N. Ali, S. Mirza, A. Fatma, Y. Memon, M. Karim, R. Tongia, and R. Rosenfeld. Healthline: Speech-based access to health information by low-literate users. In *IEEE Int. Conf. on ICTD*, pages 131–139.

R. Tucker and K. Shalonova. 2004. The Local Language Speech Technology Initiative. In *SCALLA Conf.*, Nepal.

B. Wheatley, K. Kondo, W. Anderson, and Y. Muthusumy. 1994. An evaluation of cross-language adaptation for rapid HMM development in a new language. In *ICASSP*, pages 237–240, Adelaide.

Y. Wu, R. Zhang, and A. Rudnicky. 2007. Data selection for speech recognition. *ASRU workshop*, pages 562–565, Dec.

S. Zerbian and E. Barnard. 2008. Phonetics of intonation in South African Bantu languages. *Southern African Linguistics and Applied Language Studies*, 26(2):235–254.

The SAWA Corpus: a Parallel Corpus English - Swahili

Guy De Pauw
CNTS - Language Technology Group, University of Antwerp, Belgium
School of Computing and Informatics, University of Nairobi, Kenya
guy.depauw@ua.ac.be

Peter Waiganjo Wagacha
School of Computing and Informatics, University of Nairobi, Kenya
waiganjo@uonbi.ac.ke

Gilles-Maurice de Schryver
African Languages and Cultures, Ghent University, Belgium
Xhosa Department, University of the Western Cape, South Africa
gillesmaurice.deschryver@ugent.be

Abstract

Research in data-driven methods for Machine Translation has greatly benefited from the increasing availability of parallel corpora. Processing the same text in two different languages yields useful information on how words and phrases are translated from a source language into a target language. To investigate this, a parallel corpus is typically aligned by linking linguistic tokens in the source language to the corresponding units in the target language. An aligned parallel corpus therefore facilitates the automatic development of a machine translation system and can also bootstrap annotation through projection. In this paper, we describe data collection and annotation efforts and preliminary experimental results with a parallel corpus English - Swahili.

1 Introduction

Language technology applications such as machine translation can provide an invaluable, but all too often ignored, impetus in bridging the digital divide between the Western world and Africa. Quite a few localization efforts are currently underway that improve ICT access in local African languages. Vernacular content is now increasingly being published on the Internet, and the need for robust language technology applications that can process this data is high.

For a language like Swahili, digital resources have become increasingly important in everyday life, both in urban and rural areas, particularly thanks to the increasing number of web-enabled mobile phone users in the language area. But most research efforts in the field of natural language processing (NLP) for African languages are still firmly rooted in the rule-based paradigm. Language technology components in this sense are usually straight implementations of insights derived from grammarians. While the rule-based approach definitely has its merits, particularly in terms of design transparency, it has the distinct disadvantage of being highly language-dependent and costly to develop, as it typically involves a lot of expert manual effort.

Furthermore, many of these systems are decidedly *competence*-based. The systems are often tweaked and tuned towards a small set of ideal sample words or sentences, ignoring the fact that real-world language technology applications have to be principally able to handle the *performance* aspect of language. Many researchers in the field are quite rightly growing weary of publications that ignore quantitative evaluation on real-world data or that report incredulously high accuracy scores, excused by the erroneously perceived *regularity* of African languages.

In a linguistically diverse and increasingly computerized continent such as Africa, the need for a more empirical approach to language technology is high. The data-driven, corpus-based approach described in this paper establishes such an alternative, so far not yet extensively investigated for African languages. The main advantage of this

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 9–16,
Athens, Greece, 31 March 2009.

approach is its language independence: all that is needed is (linguistically annotated) language data, which is fairly cheap to compile. Given this data, existing state-of-the-art algorithms and resources can consequently be re-used to quickly develop robust language applications and tools.

Most African languages are however resource-scarce, meaning that digital text resources are few. An increasing number of publications however are showing that carefully selected procedures can indeed bootstrap language technology for Swahili (De Pauw et al., 2006; De Pauw and de Schryver, 2008), Northern Sotho (de Schryver and De Pauw, 2007) and smaller African languages (Wagacha et al., 2006a; Wagacha et al., 2006b; De Pauw and Wagacha, 2007; De Pauw et al., 2007a; De Pauw et al., 2007b).

In this paper we outline on-going research on the development of a parallel corpus English - Swahili. The parallel corpus is designed to bootstrap a data-driven machine translation system for the language pair in question, as well as open up possibilities for projection of annotation.

We start off with a short survey of the different approaches to machine translation (Section 2) and showcase the possibility of projection of annotation (Section 3). We then concentrate on the required data collection and annotation efforts (Section 4) and describe preliminary experiments on sentence, word and morpheme alignment (Sections 5 and 6). We conclude with a discussion of the current limitations to the approach and provide pointers for future research (Section 7).

2 Machine Translation

The main task of Machine Translation (MT) can be defined as having a computer take a text input in one language, the Source language (SL), decode its meaning and re-encode it producing as output a similar-meaning text in another language, the Target language (TL). The idea of building an application to automatically convert text from one language to an equivalent text-meaning in a second language traces its roots back to Cold War intelligence efforts in the 1950's and 60's for Russian-English text translations. Since then a large number of MT systems have been developed with varying degrees of success. For an excellent overview of the history of MT, we refer the reader to Hutchins (1986).

The original dream of creating a fully automatic MT system has long since been abandoned and most research in the field currently concentrates on minimizing human pre- and post-processing effort. A human translator is thus considered to work alongside the MT system to produce faster and more consistent translations.

The Internet brought in an interesting new dimension to the purpose of MT. In the mid 1990s, free on-line translation services began to surface with an increasing number of MT vendors. The most famous example is Yahoo!'s Babelfish , offering on-line versions of Systran to translate English, French, German, Spanish and other Indo-European languages. Currently Google.inc is also offering translation services. While these systems provide far from perfect output, they can often give readers a sense of what is being talked about on a web page in a language (and often even character set) foreign to them.

There are roughly three types of approaches to machine translation:

1. **Rule-based** methods perform translation using extensive lexicons with morphological, syntactic and semantic information, and large sets of manually compiled rules, making them very labor intensive to develop.

2. **Statistical** methods entail the collection and statistical analysis of bilingual text corpora, i.e. parallel corpora. The technique tries to find the highest probability translation of a sentence or phrase among the exponential number of choices.

3. **Example-based** methods are similar to statistical methods in that they are parallel corpus driven. An Example-Based Machine Translator (EBMT) scans for patterns in both languages and relates them in a translation memory.

Most MT systems currently under development are based on methods (2) and/or (3). Research in these fields has greatly benefited from the increasing availability of parallel corpora, which are needed to bootstrap these approaches. Such a parallel corpus is typically aligned by linking, either automatically or manually, linguistic tokens in the source language to the corresponding units in the target language. Processing this data enables the development of fast and effective MT systems in both directions with a minimum of human involvement.

	English Sentences	Swahili Sentences	English Words	Swahili Words
New Testament	7.9k		189.2k	151.1k
Quran	6.2k		165.5k	124.3k
Declaration of HR	0.2k		1.8k	1.8k
Kamusi.org	5.6k		35.5k	26.7k
Movie Subtitles	9.0k		72.2k	58.4k
Investment Reports	3.2k	3.1k	52.9k	54.9k
Local Translator	1.5k	1.6k	25.0k	25.7k
Full Corpus Total	33.6k	33.6k	542.1k	442.9k

Table 1: Overview of the data in the SAWA Corpus

3 Projection of Annotation

While machine translation constitutes the most straightforward application of a parallel corpus, projection of annotation has recently become an interesting alternative use of this type of resource. As previously mentioned, most African languages are resource-scarce: annotated data is not only unavailable, but commercial interest to develop these resources is limited. Unsupervised approaches can be used to bootstrap annotation of a resource-scarce language (De Pauw and Wagacha, 2007; De Pauw et al., 2007a) by automatically finding linguistic patterns in large amounts of raw text.

Projection of annotation attempts to achieve the same goal, but through the use of a parallel corpus. These techniques try to transport the annotation of a well resourced source language, such as English, to texts in a target language. As a natural extension of the domain of machine translation, these methods employ parallel corpora which are aligned at the sentence and word level. The direct correspondence assumption coined in Hwa et al. (2002) hypothesizes that words that are aligned between source and target language, must share linguistic features as well. It therefore allows for the annotation of the words in the source language to be projected unto the text in the target language. The following general principle holds: the closer source and target language are related, the more accurate this projection can be performed. Even though lexical and structural differences between languages prevent a simple one-to-one mapping, knowledge transfer is often able to generate a well directed annotation of the target language.

This holds particular promise for the annotation of dependency analyses for Swahili, as exemplified in Figure 1, since dependency grammar focuses on semantic relationships, rather than core syntactic properties, that are much more troublesome to project across languages. The idea is that a relationship that holds between two words in the source language (for instance the *subj* relationship between *cat* and *fell*), also holds for the corresponding linguistic tokens in the target language, i.e. *paka* and *alianguka*.

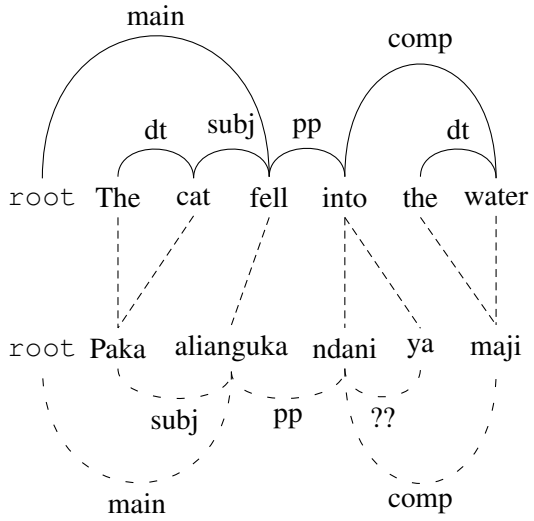


Figure 1: Projection of Dependency Analysis Annotation

In the next section we describe data collection and preprocessing efforts on the SAWA Corpus, a parallel corpus English - Swahili (cf. Table 1), which will enable this type of projection of annotation, as well as the development of a data-driven machine translation system.

4 Data Collection and Annotation

While digital data is increasingly becoming available for Swahili on the Internet, sourcing useful

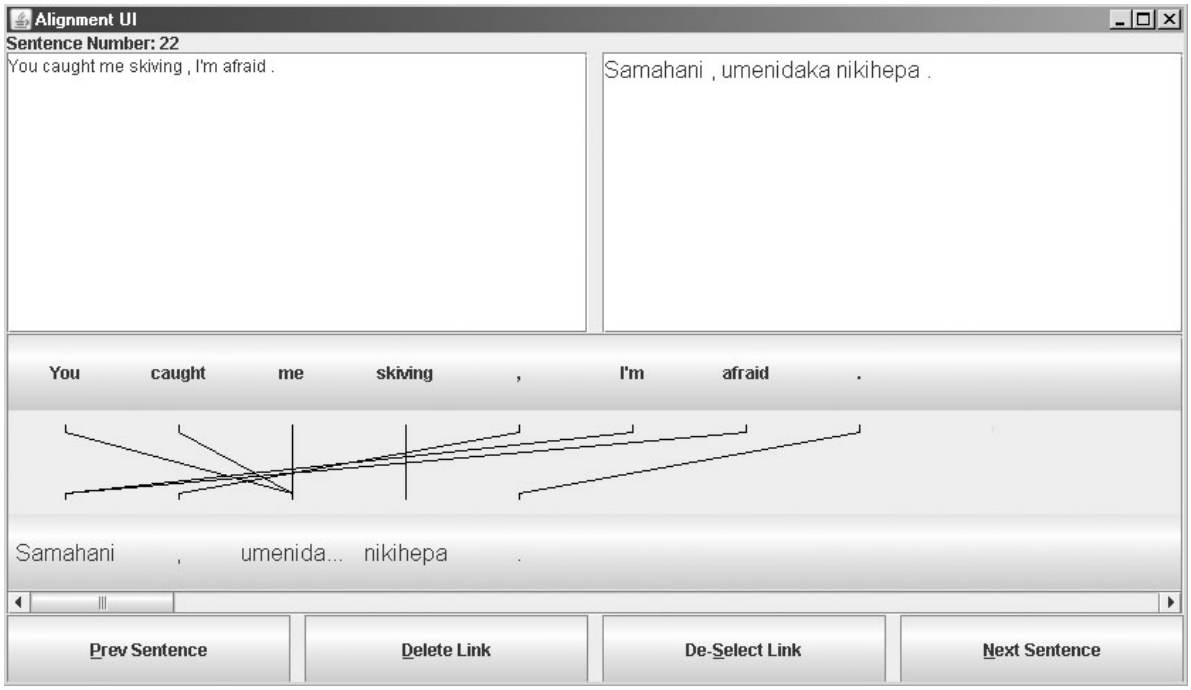


Figure 2: Manual word alignment using the UMIACS interface

bilingual data is far from trivial. At this stage in the development of the MT system, it is paramount to use faithfully translated material, as this benefits further automated processing. The corpus-based MT approaches we wish to employ, require word alignment to be performed on the texts, during which the words in the source language are linked to the corresponding words in the target language (also see Figures 1 and 2).

But before we can do this, we need to perform sentence-alignment, during which we establish an unambiguous mapping between the sentences in the source text and the sentences in the target text. While some data is inherently sentence-aligned, other texts require significant preprocessing before word alignment can be performed.

The SAWA Corpus currently consists of a reasonable amount of data (roughly half a million words in each language), although this is not comparable to the resources available to Indo-European language pairs, such as the Hansard corpus (Roukos et al., 1997) (2.87 million sentence pairs). Table 1 gives an overview of the data available in the SAWA Corpus. For each segment it lists the number of sentences and words in the respective languages.

4.1 Sentence-aligned Resources

We found digitally available Swahili versions of the New Testament and the Quran for which we sourced the English counterparts. This is not a trivial task when, as in the case of the Swahili documents, the exact source of the translation is not provided. By carefully examining subtle differences in the English versions, we were however able to track down the most likely candidate. While religious material has a specific register and may not constitute ideal training material for an open-ended MT system, it does have the advantage of being inherently aligned on the verse level, facilitating further sentence alignment. Another typical bilingual text is the UN Declaration of Human Rights, which is available in many of the world's languages, including Swahili. The manual sentence alignment of this text is greatly facilitated by the fixed structure of the document.

The downloadable version of the on-line dictionary English-Swahili (Benjamin, 2009) contains individual example sentences associated with the dictionary entries. These can be extracted and used as parallel data in the SAWA corpus. Since at a later point, we also wish to study the specific linguistic aspects of spoken language, we opted to have some movie subtitles manually translated. These can be extracted from DVDs and while the

language is compressed to fit on screen and constitutes scripted language, they nevertheless provide a reasonable approximation of spoken language. Another advantage of this data is that it is inherently sentence-aligned, thanks to the technical time-coding information. It also opens up possibilities for MT systems with other language pairs, since a commercial DVD typically contains subtitles for a large number of other languages as well.

4.2 Paragraph-aligned Resources

The rest of the material consists of paragraph-aligned data, which was manually sentence-aligned. We obtained a substantial amount of data from a local Kenyan translator. Finally, we also included Kenyan investment reports. These are yearly reports from local companies and are presented in both English and Swahili. A major difficulty was extracting the data from these documents. The company reports are presented in colorful brochures in PDF format, meaning automatic text exports require manual post-processing and paragraph alignment (Figure 3). They nevertheless provide a valuable resource, since they come from a fairly specific domain and are a good sample of the type of text the projected MT system may need to process in a practical setting.

The reader may note that there is a very diverse variety of texts within the SAWA corpus, ranging from movie subtitles to religious texts. While it certainly benefits the evaluation to use data from texts in one specific language register, we have chosen to maintain variety in the language data at this point. Upon evaluating the decoder at a later stage, we will however investigate the bias introduced by the specific language registers in the corpus.

4.3 Word Alignment

All of the data in the corpus was subsequently tokenized, which involves automatically cleaning up the texts, conversion to UTF-8, and splitting punctuation from word forms. The next step involved scanning for sentence boundaries in the paragraph-aligned text, to facilitate the automatic sentence alignment method described in Section 5.

While not necessary for further processing, we also performed manual word-alignment annotation. This task can be done automatically, but it is useful to have a gold-standard reference against which we can evaluate the automated method.


EAST AFRICAN BREWERIES LIMITED

Jukumu la kampuni kwa jamii -
Kuleta mabadiliko yenye manufaa

Kukuza unywaji wa Hekima

Mafunzo kwa wahudumu wa Baa

Mapambo ya biashara na kampeini za vyombo vya habari

Kanuri za uuzaji

Sera ya pombe kwa wafanyikazi

Makala ya Kila Siku juu ya unywaji wa hekima

EABL ANNUAL REPORT & FINANCIAL STATEMENTS 2007

33

Figure 3: Text Extraction from Bilingual Investment Report

Monitoring the accuracy of the automatic word-alignment method against the human reference, will allow us to tweak parameters to arrive at the optimal settings for this language pair.

We used the UMIACS word alignment interface (Hwa and Madnani, 2004) for this purpose and asked the annotators to link the words between the two sentences (Figure 2). Given the linguistic differences between English and Swahili, this is by no means a trivial task. Particularly the morphological richness of Swahili means that there is a lot of convergence from words in English to words in Swahili (also see Section 6). This alignment was done on some of the manual translations of movie subtitles, giving us a gold-standard word-alignment reference of about 5,000 words. Each annotator's work was cross-checked by another annotator to improve correctness and consistency.

5 Alignment Experiments

There are a number of packages available to process parallel corpora. To preprocess the paragraph-aligned texts, we used Microsoft's bilingual sentence aligner (Moore, 2002). The

Precision	Recall	F($\beta = 1$)
39.4%	44.5%	41.79%

Table 2: Precision, Recall and F-score for the word-alignment task using GIZA++

Precision	Recall	F($\beta = 1$)
50.2%	64.5%	55.8%

Table 3: Precision, Recall and F-score for the morpheme/word-alignment task using GIZA++

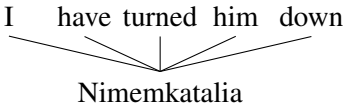
output of the sentence alignment was consequently manually corrected. We found that 95% of the sentences were correctly aligned with most errors being made on sentences that were not present in English, i.e. instances where the translator decided to add an extra clarifying sentence to the direct translation from English. This also explains why there are more Swahili words in the paragraph aligned texts than in English, while the situation is reversed for the sentence aligned data.

For word-alignment, the state-of-the-art method is GIZA++ (Och and Ney, 2003), which implements the word alignment methods IBM1 to IBM5 and HMM. While this method has a strong Indo-European bias, it is nevertheless interesting to see how far we can get with the default approach used in statistical MT.

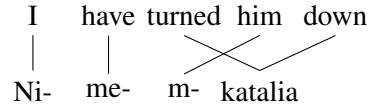
We evaluate by looking at the word alignments proposed by GIZA++ and compare them to the manually word-aligned section of the SAWA Corpus. We can quantify the evaluation by calculating precision and recall and their harmonic mean, the F-score (Table 2). The former expresses how many links are correct, divided by the total number of links suggested by GIZA++. The latter is calculated by dividing the number of correct links, by the total number of links in the manual annotation. The underwhelming results presented in Table 2 can be attributed to the strong Indo-European bias of the current approaches. It is clear that extra linguistic data sources and a more elaborate exploration of the experimental parameters of GIZA++ will be needed, as well as a different approach to word-alignment. In the next section, we describe a potential solution to the problem by defining the problem on the level of the morpheme.

6 Alignment into an Agglutinating Language

The main problem in training a GIZA++ model for the language pair English - Swahili is the strong agglutinating nature of the latter. Alignment patterns such as the one in Figures 1 and 2 are not impossible to retrieve. But no corpus is exhaustive enough to provide enough linguistic evidence to unearth strongly converging alignment patterns, such as the one in Example 1.

(1)


Morphologically deconstructing the word however can greatly relieve the sparse data problem for this task:

(2)


The isolated Swahili morphemes can more easily be linked to their English counterparts, since there will be more linguistic evidence in the parallel corpus, linking for example *ni* to *I* and *m* to *him*. To perform this kind of morphological analysis, we developed a machine learning system trained and evaluated on the Helsinki corpus of Swahili (Hurskainen, 2004). Experimental results show that the data-driven approach achieves state-of-the-art performance in a direct comparison with a rule-based method, with the added advantage of being robust to word forms for previously unseen lemmas (De Pauw and de Schryver, 2008). We can consequently use morphological deconstruction as a preprocessing step for the alignment task, similar to the method described by Goldwater and McClosky (2005), Oflazer (2008) and Stymne et al. (2008).

We have no morphologically aligned parallel data available, so evaluation of the morphology-based approach needs to be done in a roundabout way. We first morphologically decompose the Swahili data and run GIZA++ again. Then we recompile the Swahili words from the morphemes and group the word alignment links accordingly. Incompatible linkages are removed. The updated scores are presented in Table 3. While this certainly improves on the scores in Table 2, we need to be aware of the difficulty that the morphological preprocessing step will introduce in the decoding phase, necessitating the introduction of a language model that not only works on the word level, but

also on the level of the morpheme.

For the purpose of projection of annotation, this is however not an issue. We performed a preliminary experiment with a dependency-parsed English corpus, projected unto the morphologically decompounded tokens in Swahili. We are currently lacking the annotated gold-standard data to perform quantitative evaluation, but have observed interesting annotation results, that open up possibilities for the morphological analysis of more resource-scarce languages.

7 Discussion

In this paper we presented parallel corpus collection work that will enable the construction of a machine translation system for the language pair English - Swahili, as well as open up the possibility of corpus annotation through projection. We are confident that we are approaching a critical amount of data that will enable good word alignment that can subsequently be used as a model for an MT decoding system, such as the Moses package (Koehn et al., 2007). While the currently reported scores are not yet state-of-the-art, we are confident that further experimentation and the addition of more bilingual data as well as the introduction of extra linguistic features will raise the accuracy level of the proposed MT system.

Apart from the morphological deconstruction described in Section 6, the most straightforward addition is the introduction of part-of-speech tags as an extra layer of linguistic description, which can be used in word alignment model IBM5. The current word alignment method tries to link word forms, but knowing that for instance a word in the source language is a noun, will facilitate linking it to a corresponding noun in the target language, rather than considering a verb as a possible match. Both for English (Ratnaparkhi, 1996) and Swahili (De Pauw et al., 2006), we have highly accurate part-of-speech taggers available.

Another extra information source that we have so far ignored is a digital dictionary as a seed for the word alignment. The kamusiproject.org electronic dictionary will be included in further word-alignment experiments and will undoubtedly improve the quality of the output.

Once we have a stable word alignment module, we will further conduct learning curve experiments, in which we train the system with gradually increasing amounts of data. This will provide us with information on how much more data we need to achieve state-of-the-art performance. This additional data can be automatically found by parallel web mining, for which a few systems have recently become available (Resnik and Smith, 2003).

Furthermore, we will also look into the use of comparable corpora, i.e. bilingual texts that are not straight translations, but deal with the same subject matter. These have been found to work as additional material within a parallel corpus (McEnery and Xiao, 2007) and may further help improve the development of a robust, open-ended and bidirectional machine translation system for the language pair English - Swahili. The most innovative prospect of the parallel corpus is the annotation of dependency analysis in Swahili, not only on the syntactic level, but also on the level of the morphology. The preliminary experiments indicate that this approach might provide a valuable technique to bootstrap annotation in truly resource-scarce languages.

Acknowledgments

The research presented in this paper was made possible through the support of the VLIR-IUC-UON program and was partly funded by the SAWA BOF UA-2007 project. The first author is funded as a Postdoctoral Fellow of the Research Foundation - Flanders (FWO). We are greatly indebted to Dr. James Omboga Zaja for contributing some of his translated data, to Mahmoud Shokrollahi-Far for his advice on the Quran and to Anne Kimani, Chris Wangai Njoka and Naomi Maajabu for their annotation efforts.

References

M. Benjamin. 2009. *The Kamusi Project*. Available at: http://www.kamusiproject.org (Accessed: 14 January 2009).

G. De Pauw and G.-M. de Schryver. 2008. Improving the computational morphological analysis of a Swahili corpus for lexicographic purposes. *Lexikos*, 18:303–318.

G. De Pauw and P.W. Wagacha. 2007. Bootstrapping morphological analysis of Gĩkũyũ using unsupervised maximum entropy learning. In *Proceedings of the eighth INTERSPEECH conference*, Antwerp, Belgium.

G. De Pauw, G.-M. de Schryver, and P.W. Wagacha. 2006. Data-driven part-of-speech tagging of

Kiswahili. In P. Sojka, I. Kopeček, and K. Pala, editors, *Proceedings of Text, Speech and Dialogue, 9th International Conference*, volume 4188 of *Lecture Notes in Computer Science*, pages 197–204, Berlin, Germany. Springer Verlag.

G. De Pauw, P.W. Wagacha, and D.A. Abade. 2007a. Unsupervised induction of Dholuo word classes using maximum entropy learning. In K. Getao and E. Omwenga, editors, *Proceedings of the First International Computer Science and ICT Conference*, pages 139–143, Nairobi, Kenya. University of Nairobi.

G. De Pauw, P.W. Wagacha, and G.-M. de Schryver. 2007b. Automatic diacritic restoration for resource-scarce languages. In Václav Matoušek and Pavel Mautner, editors, *Proceedings of Text, Speech and Dialogue, Tenth International Conference*, volume 4629 of *Lecture Notes in Computer Science*, pages 170–179, Heidelberg, Germany. Springer Verlag.

G.-M. de Schryver and G. De Pauw. 2007. Dictionary writing system (DWS) + corpus query package (CQP): The case of Tshwanelex. *Lexikos*, 17:226–246.

S. Goldwater and D. McClosky. 2005. Improving statistical MT through morphological analysis. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 676–683, Vancouver, Canada.

Google. 2009. *Google Translate*. Available at http://www.google.com/translate (Accessed: 14 January 2009.

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

W.J. Hutchins. 1986. *Machine translation: past, present, future*. Ellis, Chichester.

R. Hwa and N. Madnani. 2004. *The UMIACS Word Alignment Interface.* Available at: http://www.umiacs.umd.edu/~nmadnani/alignment/forclip.htm (Accessed: 14 January 2009).

R. Hwa, Ph. Resnik, A. Weinberg, and O. Kolak. 2002. Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 392–399, Philadelphia, PA, USA.

Ph. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, Prague, Czech Republic.

A.M. McEnery and R.Z Xiao. 2007. Parallel and comparable corpora: What are they up to? In *Incorporating Corpora: Translation and the Linguist. Translating Europe. Multilingual Matters*, Clevedon, UK.

R.C. Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, volume 2499 of *Lecture Notes in Computer Science*, pages 135–144, Berlin, Germany. Springer Verlag.

F.J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

K. Oflazer. 2008. Statistical machine translation into a morphologically complex language. In *Computational Linguistics and Intelligent Text Processing*, pages 376–388, Berlin, Germany. Springer Verlag.

A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In E. Brill and K. Church, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142. Association for Computational Linguistics.

Ph. Resnik and N.A. Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29(1):349–380.

S. Roukos, D. Graff, and D. Melamed. 1997. *Hansard French/English.* Available at: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC95T20 (Accessed: 14 January 2009).

S. Stymne, M. Holmqvist, and L. Ahrenberg. 2008. Effects of morphological analysis in translation between German and English. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 135–138, Columbus, USA.

P.W. Wagacha, G. De Pauw, and K. Getao. 2006a. Development of a corpus for Gĩkũyũ using machine learning techniques. In J.C. Roux, editor, *Proceedings of LREC workshop - Networking the development of language resources for African languages*, pages 27–30, Genoa, Italy, May, 2006. European Language Resources Association, ELRA.

P.W. Wagacha, G. De Pauw, and P.W. Githinji. 2006b. A grapheme-based approach for accent restoration in Gĩkũyũ. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, pages 1937–1940, Genoa, Italy, May, 2006. European Language Resources Association, ELRA.

Yahoo! 2009. *Babelfish*. Available at http://babelfish.yahoo.com (Accessed: 14 January 2009).

Information Structure in African Languages: Corpora and Tools

Christian Chiarcos*, Ines Fiedler, Mira Grubic*, Andreas Haida**, Katharina Hartmann**, Julia Ritz*, Anne Schwarz**, Amir Zeldes**, Malte Zimmermann***

***** Universität Potsdam
Potsdam, Germany
{chiarcos|grubic|julia|malte}@ling.uni-potsdam.de

****** Humboldt-Universität zu Berlin
Berlin, Germany
{ines.fiedler|andreas.haida|k.hartmann|anne.schwarz|amir.zeldes}@rz.hu-berlin.de

Abstract

In this paper, we describe tools and resources for the study of African languages developed at the Collaborative Research Centre "Information Structure". These include deeply annotated data collections of 25 subsaharan languages that are described together with their annotation scheme, and further, the corpus tool ANNIS that provides a unified access to a broad variety of annotations created with a range of different tools. With the application of ANNIS to several African data collections, we illustrate its suitability for the purpose of language documentation, distributed access and the creation of data archives.

1 Information Structure

The Collaborative Research Centre (CRC) "Information structure: the linguistic means for structuring utterances, sentences and texts" brings together scientists from different fields of linguistics and neighbouring disciplines from the University of Potsdam and the Humboldt-University Berlin. Our research comprises the use and advancement of corpus technologies for complex linguistic annotations, such as the annotation of information structure (IS). We define IS as the structuring of linguistic information in order to optimize information transfer within discourse: information needs to be prepared ("packaged") in different ways depending on the goals a speaker pursues within discourse.

Fundamental concepts of IS include the concepts `topic', `focus', `background' and `information status'. Broadly speaking, the topic is the entity a specific sentence is construed *about*, focus represents the *new* or *newsworthy* information a sentence conveys, background is that part of the sentence that is *familiar* to the hearer, and information status refers to different *degrees of familiarity* of an entity.

Languages differ wrt. the means of realization of IS, due to language-specific properties (e.g., lexical tone). This makes a typological comparison of traditionally less-studied languages to existing theories, mostly on European languages, very promising. Particular emphasis is laid on the study of focus, its functions and manifestations in different subsaharan languages, as well as the differentiation between different types of focus, i.e., term focus (focus on arguments/adjuncts), predicate focus (focus on verb/verb phrase/TAM/truth value), and sentence focus (focus on the whole utterance).

We describe corpora of 25 subsaharan languages created for this purpose, together with ANNIS, the technical infrastructure developed to support linguists in their work with these data collections. ANNIS is specifically designed to support corpora with rich and deep annotation, as IS manifests itself on practically all levels of linguistic description. It provides user-friendly means of querying and visualizations for different kinds of linguistic annotations, including flat, layer-based annotations as used for linguistic glosses, but also hierarchical annotations as used for syntax annotation.

2 Research Activities at the CRC

Within the Collaborative Research Centre, there are several projects eliciting data in large amounts and great diversity. These data, originating from different languages, different modes (written and spoken language) and specific research questions characterize the specification of the linguistic database ANNIS.

2.1 Linguistic Data Base

The project "Linguistic database for information structure: Annotation and Retrieval", further

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 17–24,
Athens, Greece, 31 March 2009.

database project, coordinates annotation activities in the CRC, provides service to projects in the creation and maintenance of data collections, and conducts theoretical research on multi-level annotations. Its primary goals, however, are the development and investigation of techniques to process, to integrate and to exploit deeply annotated corpora with multiple kinds of annotations. One concrete outcome of these efforts is the linguistic data base ANNIS described further below. For the specific facilities of ANNIS, its application to several corpora of African languages and its use as a general-purpose tool for the publication, visualization and querying of linguistic data, see Sect. 5.

2.2 Gur and Kwa Languages

Gur and Kwa languages, two genetically related West African language groups, are in the focus of the project “Interaction of information structure and grammar in Gur and Kwa languages”, henceforth *Gur-Kwa project*. In a first research stage, the precise means of expression of the pragmatic category *focus* were explored as well as their functions in Gur and Kwa languages. For this purpose, a number of data collections for several languages were created (Sect. 3.1). Findings obtained with this data led to different subquestions which are of special interest from a cross-linguistic and a theoretical point of view. These concern (i) the analysis of syntactically marked focus constructions with features of narrative sentences (Schwarz & Fiedler 2007), (ii) the study of verb-centered focus (i.e., focus on verb/TAM/truth value), for which there are special means of realization in Gur and Kwa (Schwarz, forthcoming), (iii) the identification of systematic *focus-topic*-overlap, i.e., coincidence of focus and topic in sentence-initial nominal constituents (Fiedler, forthcoming). The project's findings on IS are evaluated typologically on 19 selected languages. The questions raised by the project serve the superordinate goal to expand our knowledge of linguistically relevant information structural categories in the less-studied Gur and Kwa languages as well as the interaction between IS, grammar and language type.

2.3 Chadic Languages

The project “Information Structure in the Chadic Languages”, henceforth *Chadic project*, investigates focus phenomena in Chadic languages. The Chadic languages are a branch of the Afro-Asiatic language family mainly spoken in northern Nigeria, Niger, and Chad. As tone languages, the Chadic languages represent an interesting subject for research into focus because here, intonational/tonal marking – a commonly used means for marking focus in European languages – is in potential conflict with lexical tone, and so, Chadic languages resort to alternative means for marking focus.

The languages investigated in the Chadic project include the western Chadic languages Hausa, Tangale, and Guruntum and the central Chadic languages Bura, South Marghi, and Tera. The main research goals of the Chadic project are a deeper understanding of the following asymmetries: (i) subject focus is obligatorily marked, but marking of object focus is optional; (ii) in Tangale and Hausa there are sentences that are ambiguous between an object-focus interpretation and a predicate-focus interpretation, but in intonation languages like English and German, object focus and predicate focus are always marked differently from each other; (iii) in Hausa, Bole, and Guruntum there is only a tendency to distinguish different types of focus (new-information focus vs. contrastive focus), but in European languages like Hungarian and Finnish, this differentiation is obligatory.

2.4 Focus from a Cross-linguistic Perspective

The project "Focus realization, focus interpretation, and focus use from a cross-linguistic perspective", further *focus project*, investigates the correspondence between the realization, interpretation and use of with an emphasis on focus in African and south-east Asian languages. It is structured into three fields of research: (i) the relation between differences in realization and differences in semantic meaning or pragmatic function, (ii) realization, interpretation and use of predicate focus, and (iii) association with focus.

The relation between differences in realization and semantic/pragmatic differences (i) particularly pertains the semantic interpretation of focus: For Hungarian and Finnish, a differentiation between two semantic types of foci corresponding to two different types of focus realization was suggested, and we investigate whether the languages studied here have a similar distinction between two (or more) semantic focus types, whether this may differ

from language to language, and whether differences in focus realization correspond to semantic or pragmatic differences.

The investigation of realization, interpretation and use of predicate focus (ii) involves the questions why different forms of predicate focus are often realized in the same way, why they are often not obligatorily marked, and why they are often marked differently from term focus.

Association with focus (iii) means that the interpretation of the sentence is influenced by the focusing of a particular constituent, marked by a focus-sensitive expression (e.g., particles like `only', or quantificational adverbials like `always'), while usually**,** focus does not have an impact on the truth value of a sentence. The project investigates which focus-sensitive expressions there are in the languages studied, what kinds of constituents they associate with, how this association works, and whether it works differently for focus particles and quantificational adverbials.

3 Collections of African Language Data at the CRC

3.1 Gur and Kwa Corpora

The Gur and Kwa corpora currently comprise data from 19 languages.

Due to the scarceness of information available on IS in the Gur and Kwa languages, data had to be elicited, most of which was done during field research, mainly in West Africa, and some in Germany with the help of native speakers of the respective languages. The typologically diverse languages in which we elicited data by ourselves are: Baatonum, Buli, Byali, Dagbani, Ditammari, Gurene, Konkomba, Konni, Nateni, Waama, Yom (Gur languages) and Aja, Akan, Efutu, Ewe, Fon, Foodo, Lelemi, Anii (Kwa languages).

The elicitation of the data based mainly on the questionnaire on information structure, developed by our research group (QUIS, see Section 4.2). This ensured that comparable data for the typological comparison were obtained. Moreover, language-specific additional tasks and questionnaires tailored to a more detailed analysis or language-specific traits were developed.

As the coding of IS varies across different types of texts, different text types were included in the corpus, such as (semi-)spontaneous speech, translations, mono- and dialogues. Most of the languages do not have a long literacy tradition, so that the corpus data mainly represents oral communication.

In all, the carefully collected heterogeneous data provide a corpus that gives a comprehensive picture of IS, and in particular the focus systems, in these languages.

3.2 Hausar Baka Corpus

In the Chadic project, data from 6 Chadic languages are considered.

One of the larger data sets annotated in the Chadic project is drawn from *Hausar Baka* (Randell, Bature & Schuh 1998), a collection of videotaped Hausa dialogues recording natural interaction in various cultural milieus, involving over fifty individuals of different age and gender. The annotated data set consists of approximately 1500 sentences.

The corpus was annotated according to the guidelines for Linguistic Information Structure Annotation (LISA, see Section 4.2). The Chadic languages show various forms of syntactic displacement, and in order to account for this, an additional annotation level was added: constituents are marked as `ex-situ="+"` if they occur displaced from their canonical, unmarked position.

An evaluation of the focus type and the displacement status reveals tendencies in the morphosyntactic realization of different focus types, see Sect. 5.2.

3.3 Hausa Internet Corpus

Besides these data collections that are currently available in the CRC and in ANNIS, further resources are continuously created. As such, a corpus of written Hausa is created in cooperation with another NLP project of the CRC.

The corpora previously mentioned mostly comprise elicited sentences from little-documented languages with rather small language communities. Hausa, in contrast, is spoken by more than 24 million native speakers, with large amounts of Hausa material (some of it parallel to material in other, more-studied languages) available on the internet. This makes Hausa a promising language for the creation of resources that enable a quantitative study of information structure.

The Hausa internet corpus is designed to cover different kinds of written language, including news articles from international radio stations (e.g., http://www.dw-world.de), religious texts, literary prose but also material similar to spontaneous spoken language (e.g., in chat logs).

Parallel sections of the corpus comprise excerpts from the novel Ruwan Bagaja by Abubakar Imam, Bible and Qur'an sections, and the Declaration of Human Rights. As will be described in Section 4.3, these parallel sections open the possibility of semiautomatic morphosyntactic annotation, providing a unique source for the study of information structure in Hausa. Sect. 5.2 gives an example for bootstrapping `ex-situ` constituents in ANNIS only on the basis of morphosyntactic annotation.

4 Data Elicitation and Annotation

4.1 Elicitation with QUIS

The questionnaire on information structure (Skopeteas et al., 2006) provides a tool for the collection of natural linguistic data, both spoken and written, and, secondly, for the elaboration of grammars of IS in genetically diverse languages. Focus information, for instance, is typically elicited by embedding an utterance in a question context. To avoid the influence of a mediator (working) language, the main body of QUIS is built on the basis of pictures and short movies representing a nearly culture- and language-neutral context. Besides highly controlled experimental settings, less controlled settings serve the purpose of eliciting longer, cohesive, natural texts for studying categories such as focus and topic in a near-natural environment.

4.2 Transcription and Manual Annotation

In the CRC, the annotation scheme LISA has been developed with special respect to applicability across typologically different languages (Dipper et al., 2007). It comprises guidelines for the annotation of phonology, morphology, syntax, semantics and IS.

The data mentioned above is, in the case of speech, transcribed according to IPA conventions, otherwise written according to orthographic conventions, and annotated with glosses and IS, a translation of each sentence into English or French, (optionally) additional notes, references to QUIS experiments, and references to audio files and metadata.

4.3 (Semi-)automatic Annotation

As to the automization of annotation, we pursue two strategies: (i) the training of classifiers on annotated data, and (ii) the projection of annotations on texts in a source language to parallel texts in a target language.

Machine Learning. ANNIS allows to export query matches and all their annotated features to the table format ARFF which serves as input to the data mining tool WEKA (Witten & Frank, 2005), where instances can be clustered, or used to train classifiers for any annotation level.

Projection. Based on (paragraph-, sentence- or verse-) aligned sections in the Hausa internet corpus, we are about to project annotations from linguistically annotated English texts to Hausa, in a first step parts of speech and possibly nominal chunks. On the projected annotation, we will train a tagger/chunker to annotate the remaining, non-parallel sections of the Hausa internet corpus. Existing manual annotations (e.g. of the Hausar Baka corpus) will then serve as a gold standard for evaluation purposes.

Concerning projection techniques, we expect to face a number of problems: (i) the question how to assign part of speech tags to categories existing only in the target language (e.g., the person-aspect complex in Hausa that binds together information about both the verb (aspect) and its (pronominal subject) argument), (ii) issues of orthography: the official orthography Hausa (*Boko*) is systematically underspecified wrt. linguistically relevant distinctions. Neither vowel length nor different qualities of certain consonants (*r*) are represented, and also, there is no marking of tones (see Examples 1 and 2, fully specified word forms in brackets). To distinguish such homographs, however, is essential to the appropriate interpretation and linguistic analysis of utterances.

(1) **ciki** – 1. [cíkìi, noun] stomach, 2. [cíkí, prep.] inside

(2) **dace** – 1. [dàacée, noun] coincidence, 2. [dáacèe, verb] be appropriate

We expect that in these cases, statistical techniques using context features may help to predict correct vowelization and tonal patterns.

5 ANNIS – the Linguistic Database of Information Structure Annotation

5.1 Conception and Architecture

ANNIS (ANNotation of Information Structure) is a web-based corpus interface built to query and visualize multilevel corpora. It allows the user to formulate queries on arbitrary, possibly nested annotation levels, which may be

conflictingly overlapping or discontinuous. The types of annotations handled by ANNIS include, among others, flat, layer-based annotations (e.g., for glossing) and hierarchical trees (e.g., syntax).

Source data. As an architecture designed to facilitate diverse and integrative research on IS, ANNIS can import formats from a broad variety of tools from NLP and manual annotation, the latter including EXMARaLDA (Schmidt, 2004), annotate (Brants and Plaehn, 2000), Synpathy (www.lat-mpi.eu/tools/synpathy/), MMAX2 (Müller and Strube, 2006), RSTTool (O'Donnell, 2000), PALinkA (Orasan, 2003), Toolbox (Busemann & Busemann, 2008) etc. These tools allow researchers to annotate data for syntax, semantics, morphology, prosody, phonetics, referentiality, lexis and much more, as their research questions require.

All annotated data are merged together via a **general interchange format** PAULA (Dipper 2005, Dipper & Götze 2005), a highly expressive standoff XML format that specifically allows further annotation levels to be added at a later time without disrupting the structure of existing annotations. PAULA, then, is the native format of ANNIS.

Backend. The ANNIS server uses a relational database that offers many advantages including full Unicode support and regular expression searches. Extensive search functionalities are supported, allowing complex relations between individual word forms and annotations, such as all forms of overlapping, contained or adjacent annotation spans, dominance axes (children, ancestors etc., as well as common parent, left- or right-most child and more), etc.

Search. In the user interface, queries can be formulated using the ANNIS Query Language (AQL). It is based on the definition of nodes to be searched for and the relationships between these nodes (see below for some examples). A graphical query builder is also included in the web interface to make access as easy as possible.

Visualization. The web interface, realized as a window-based AJAX application written in Java, provides visualization facilities for search results. Available visualizations include token-based annotations, layered annotations, tree-like annotations (directed acyclic graphs), and a discourse view of entire texts for, e.g., coreference annotation. Multimodal data is represented using an embedded media player.

Special features. By allowing queries on multiple, conflicting annotation levels simultaneously, the system supports the study of interdependencies between a potentially limitless variety of annotation levels.

At the same time, ANNIS allows us to integrate and to search through heterogeneous resources by means of a unified interface, a powerful query language and a intuitive graphical query editor and is therefore particularly well-suited for the purpose of language documentation. In particular, ANNIS can serve as a tool for the publication of data collections via internet. A fine-grained user management allows granting privileged users access to specific data collections, to make a corpus available to the public, or to seal (but preserve) a resource, e.g., until legal issues (copyright) are settled. This also makes publishing linguistic data collections possible without giving them away.

Moreover, ANNIS supports deep links to corpora and corpus queries. This means that queries and query results referred to in, e.g., a scientific paper, can be reproduced and quoted by means of (complex) links (see following example).

5.2 Using ANNIS. An Example Query

As an illustration for the application of ANNIS to the data collections presented above, consider a research question previously discussed in the study of object focus in Hausa.

In Hausa, object focus can be realized in two ways: either ex-situ or in-situ (Section 3.2). It was found that these realizations do not differ in their semantic type (Green & Jaggar 2003, Hartmann & Zimmermann 2007): instead, the marked form signals that the focused constituent (or the whole speech act) is unexpected for the hearer (Zimmermann 2008). These assumptions are consistent with findings for other African languages (Fiedler et al. 2006).

In order to verify such claims on corpora with morphosyntactic and syntactic annotation for the example of Hausa, a corpus query can be designed on the basis of the Hausar Baka corpus that comprises not only annotations for grammatical functions and information-structural categories, but also an annotation of ex-situ elements.

Figure 1: ANNIS partitur view, Hausar Baka corpus.

So, in (3), we look for ex-situ constituents (variable `#1`) in declarative sentences in the Hausa Bakar corpus, i.e., sentences that are not translated as questions (variable `#2`) such that `#1` is included in `#2 (#1 _i_ #2).`

```
(3) EX-SITU="+" &
TRANSLATION=".*[^?]" &      #1
_i_ #2
```

Considering the first 25 matches for this query on Hausar Baka, 16 examples reveal to be relevant (excluding interrogative pronouns and elliptical utterances). All of these are directly preceded by a period (sentence-initial) or a comma (preceded by *ee* 'yes', interjections or exclamations), with one exception, preceded by a sentence initial negation marker.

Only seven examples are morphologically marked by focus particles (*nee, cee*), focus-sensitive adverbs (*kawài* 'only') or quantifiers (*koomee* 'every'). In nine cases, a personal pronoun follows the ex-situ constituent, followed by the verb. Together, these constraints describe all examples retrieved, and as a generalization, we can now postulate a number of patterns that only make use of morphosyntactic and syntactic annotation (token tok, morphological segmentation MORPH, parts of speech CLASS, nominal chunks CHUNK) with two examples given below:

```
(4)      tok=/[,.!?]/ &
CHUNK="NC" & MORPH=/[cn]ee/ &
#1 . #2 & #2 . #3
(5)      tok=/[,.!?]/ &
CHUNK="NC" & CLASS=/PRON.*/ &
CLASS=/V/ & #1 . #2 &   #2 .
#3 & #3 . #4
```

In (4), we search for a nominal chunk following a punctuation sign and preceding a focus particle (*cee* or *nee*), in (5), we search for a nominal chunk preceding a sequence of pronoun/aspect marker and verb.

One example matching template (5) from the Hausar Baka corpus is given in Fig. 1.

While AQL can be used in this way to help linguists understand the grammatical realization of certain phenomena, and the grammatical context they occur in, patterns like (5) above are probably not too readable to an interested user. This deficit, however, is compensated by the graphical query builder that allows users to create AQL queries in a more intuitive way, cf. Fig. 2.

Of course, these patterns are not exhaustive and overgenerate. However, they can be directly evaluated against the manual ex-situ annotation in the Hausar Baka corpus and further refined.

So, the manual annotation of ex-situ constituents in the Hausar Baka corpus provides templates for the semi-automatic detection of ex-situ constituents in a morphosyntactically annotated corpus of Hausa: The patterns generate a set of candidate examples from which a human annotator can then choose real ex-situ constituents. Indeed, for a better understanding of ex-situ object focus, a study with a larger database of more natural language would be of great advantage, and this pattern-based approach represents a way to create such a database of ex-situ constructions in Hausa.

Finally, it would also help find instances of predicate focus. When a V(P) constituent is focused in Hausa, it is nominalized, and fronted like a focused nominal constituent (Hartmann & Zimmermann 2007).

5.3 Related Corpus Tools

Some annotation tools come with search facilities, e.g. **Toolbox** (Busemann & Busemann, 2008), a system for annotating, managing and

analyzing language data, mainly geared to lexicographic use, and **ELAN** (Hellwig et al., 2008), an annotation tool for audio and video data.

In contrast, ANNIS is not intended to provide annotation functionality. The main reason behind this is that both Toolbox and ELAN are problem-specific annotation tools with limited capabilities for application to different phenomena than they were designed for. Toolbox provides an intuitive annotation environment and search facilities for flat, word-oriented annotations; ELAN, on the other hand, for annotations that stand in a temporal relation to each other.

These tools – as well as the other annotation tools used within the CRC – are tailored to a particular type of annotation, neither of them being capable of sufficiently representing the data from all other tools. Annotation on different levels, however, is crucial for the investigation of information structural phenomena. In order to fill in this gap, ANNIS was designed primarily with the focus on visualization and querying of multi-layer annotations. In particular, ANNIS allows to integrate annotations originating from *different tools* (e.g., syntax annotation created with Synpathy, coreference annotation created with MMAX2, and flat, time-aligned annotations created with ELAN) that nevertheless refer to the *same primary data*. In this respect, ANNIS, together with the data format PAULA and the libraries created for the work with both, is best compared to general annotation frameworks such as ATLAS, NITE and LAF.

Taking the **NITE XML Toolkit** as a representative example for this kind of frameworks, it provides an abstract data model, XML-based formats for data storage and metadata, a query language, and a library with JAVA routines for data storage and manipulation, querying and visualization. Additionally, a set of command line tools and simple interfaces for corpus querying and browsing are provided, which illustrates how the libraries can be used to create one's own, project-specific corpus interfaces and tools.

Similarly to ANNIS, NXT supports time-aligned, hierarchical and pointer-based annotation, conflicting hierarchies and the embedding of multi-modal primary data. The data storage format is based on the bundling of multiple XML files similar to the standoff concept employed in LAF and PAULA.

One fundamental difference between NXT and ANNIS, however, is to be seen in the primary clientele it targets: The NITE XML Toolkit is aimed at the developer and allows to build more specialized displays, interfaces, and analyses as required by their respective end users when working with highly structured data annotated on multiple levels.

As compared to this, ANNIS is directly targeted at the end user, that is, a linguist trying to explore and to work with a particular set of corpora. Therefore, an important aspect of the ANNIS implementation is the integration with a data base and convenient means for visualization and querying.

6 Conclusion

In this paper, we described the Africanist projects of the CRC „Information Structure“ at the University of Potsdam and the Humboldt University of Berlin/Germany, together with their data collections from currently 25 subsaharan languages. Also, we have presented the linguistic database ANNIS that can be used to publish, access, query and visualize these data collections. As one specific example of our work, we have described the design and ongoing construction of a corpus of written Hausa, the Hausa internet corpus, sketched the relevant NLP techniques for (semi)automatic morphosyntactic annotation, and the application of the ANNIS Query Language to filter out ex-situ constituents and their contexts, which are relevant with regard to our goal, a better understanding of focus and information structure in Hausa and other African languages.

References

T. Brants, O. Plaehn. 2000. Interactive Corpus Annotation. In: *Proc. of LREC 2000*, Athens, Greece.

A. Busemann, K. Busemann. 2008. Toolbox Self-Training. Technical Report (Version 1.5.4 Oct 2008) http://www.sil.org/

J. Carletta, S. Evert, U. Heid, J. Kilgour, J. Robertson, H. Voormann. 2003. The NITE XML Toolkit: Flexible Annotation for Multi-modal Language Data. *Behavior Research Methods, Instruments, and Computers* 35(3), 353-363.

J. Carletta, S. Evert, U. Heid, J. Kilgour. 2005. The NITE XML Toolkit: data model and query. *Language Resources and Evaluation Journal*, 313-334.

S. Dipper. 2005. XML-based Stand-off Representation and Exploitation of Multi-Level Linguistic Annotation. In: *Proc. of Berliner XML Tage 2005 (BXML 2005)*, Berlin, Germany, 39-50.

S. Dipper, M. Götze. 2005. Accessing Heterogeneous Linguistic Data - Generic XML-based Representation and Flexible Visualization. In *Proc. of the 2nd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*. Poznan, Poland, 206-210.

S. Dipper, M. Götze, S. Skopeteas (eds.). 2007. *Information structure in cross-linguistic corpora: Annotation guidelines for phonology, morphology, syntax, semantics and information structure.* Interdisciplinary Studies on Information Structure 7, 147-187. Potsdam: University of Potsdam.

I. Fiedler. forthcoming. Contrastive topic marking in Gbe. In *Proc. of the 18th International Conference on Linguistics, Seoul, Korea, 21. - 26. July 2008.*

I. Fiedler, K. Hartmann, B. Reineke, A. Schwarz, M. Zimmermann.. forthcoming. Subject Focus in West African Languages. In M. Zimmermann, C. Féry (eds.), *Information Structure. Theoretical, Typological, and Experimental Perspectives*. Oxford: Oxford University Press. .

M. Green, P. Jaggar. 2003. Ex-situ and in-situ focus in Hausa: syntax, semantics and discourse. In J. Lecarme et al. (eds.), *Research in Afroasiatic Grammar 2 (Current Issues in Linguistic Theory)*. Amsterdam: John Benjamins. 187-213.

K. Hartmann, M. Zimmermann. 2004. Nominal and Verbal Focus in the Chadic Languages. In F. Perrill et al. (eds.), *Proc. of the Chicago Linguistics Society.* 87-101.

K. Hartmann, M. Zimmermann. 2007. In Place - Out of Place? Focus in Hausa. In K. Schwabe, S. Winkler (eds.), *On Information Structure, Meaning and Form: Generalizing Across Languages*. Benjamins, Amsterdam: 365-403.

B. Hellwig, D. Van Uytvanck, M. Hulsbosch. 2008. ELAN – Linguistic Annotator. Technical Report (as of 2008-07-31). http://www.lat-mpi.eu/tools/elan/

É. Kiss. 1998. Identificational Focus Versus Information Focus. Language 74: 245-273.

M. Krifka. 1992. A compositional semantics for multiple focus constructions, in Jacobs, J: *Informationsstruktur und Grammatik*, Opladen, Westdeutscher Verlag, 17-53.

C. Müller, M. Strube. 2006. Multi-Level Annotation of Linguistic Data with MMAX2. In: S. Braun et al. (eds.), *Corpus Technology and Language Pedagogy. New Resources, New Tools, New Methods*. Frankfurt: Peter Lang, 197–214.

M. O'Donnell. 2000. RSTTool 2.4 – A Markup Tool for Rhetorical Structure Theory. In: *Proc. of the International Natural Language Generation Conference (INLG'2000), 13-16 June 2000*, Mitzpe Ramon, Israel, 253–256.

C. Orasan. 2003. Palinka: A Highly Customisable Tool for Discourse Annotation. In: *Proc. of the 4th SIGdial Workshop on Discourse and Dialogue*, Sapporo, Japan.

R. Randell, A. Bature, R. Schuh. 1998. Hausar Baka. http://www.humnet.ucla.edu/humnet/aflang/hausarbaka/

T. Schmidt. 2004. Transcribing and Annotating Spoken Language with Exmaralda. In: *Proc. of the LREC-Workshop on XML Based Richly Annotated Corpora, Lisbon 2004*. Paris: ELRA.

A. Schwarz. Verb and Predication Focus Markers in Gur. forthcoming. In I. Fiedler, A. Schwarz (eds.), *Information structure in African languages (Typological Studies in Language)*, 307-333. Amsterdam, Philadelphia: John Benjamins.

A. Schwarz, I. Fiedler. 2007. Narrative focus strategies in Gur and Kwa. In E. Aboh et al. (eds.): *Focus strategies in Niger-Congo and Afroasiatic – On the interaction of focus and grammar in some African languages*, 267-286. Berlin: Mouton de Gruyter.

S. Skopeteas, I. Fiedler, S. Hellmuth, A. Schwarz, R. Stoel, G. Fanselow, C. Féry, M. Krifka. 2006. *Questionnaire on information structure (QUIS).* Interdisciplinary Studies on Information Structure 4. Potsdam: University of Potsdam.

I. H. Witten, E. Frank, Data mining: *Practical machine learning tools and techniques*, 2nd edn, Morgan Kaufman, San Francisco, 2005.

M. Zimmermann. 2008. Contrastive Focus and Emphasis. In *Acta Linguistica Hungarica* 55: 347-360.

A computational approach to Yorùbá morphology

Raphael Finkel
Department of Computer Science, University of Kentucky, USA
raphael@cs.uky.edu

Ọdẹ́túnjí Àjàdí, ỌDẸ́JỌBÍ
Cork Constraint Computation Center, University College Cork, Cork, Ireland.
t.odejobi@4c.ucc.ie

Abstract

We demonstrate the use of default default inheritance hierarchies to represent the morphology of Yorùbá verbs in the KATR formalism, treating inflectional exponences as markings associated with the application of rules by which complex word forms are deduced from simpler roots or stems. In particular, we suggest a scheme of slots that together make up a verb and show how each slot represents a subset of the morphosyntactic properties associated with the verb. We also show how we can account for the tonal aspects of Yorùbá, in particular, the tone associated with the emphatic ending. Our approach allows linguists to gain an appreciation for the structure of verbs, gives teachers a foundation for organizing lessons in morphology, and provides students a technique for generating forms of any verb.

1 Introduction

Recent research into the nature of morphology has demonstrated the feasibility of several approaches to the definition of a language's inflectional system. Central to these approaches is the notion of an inflectional paradigm. In general terms, the **inflectional paradigm** of a lexeme L can be regarded as a set of cells, where each cell is the pairing of L with a set of morphosyntactic properties, and each cell has a word form as its realization; for instance, the paradigm of the lexeme walk includes cells such as <WALK, {3rd singular present indicative}> and <WALK, {past}>, whose realizations are the word forms *walks* and *walked*.

Given this notion, one approach to the definition of a language's inflectional system is the **realizational** approach (Matthews 1972, Zwicky 1985, Anderson 1992, Corbett & Fraser 1993, Stump 2001); in this approach, each word form in a lexeme's paradigm is deduced from the lexical and morphosyntactic properties of the cell that it realizes by means of a system of morphological rules. For instance, the word form *walks* is deduced from the cell <WALK, {3rd singular present indicative}> by means of the rule of *-s* suffixation, which applies to the root *walk* of the lexeme WALK to express the property set {3rd singular present indicative}.

We apply the realizational approach to the study of Yorùbá verbs. Yorùbá, an Edekiri language of the Niger-Congo family (Gordon 2005), is the native language of more than 30 million people in West Africa. Although it has many dialects, all speakers can communicate effectively using Standard Yorùbá (SY), which is used in education, mass media and everyday communication (Adéwọ́lé 1988).

We represent our realizational analysis of SY in the KATR formalism (Finkel, Shen, Stump & Thesayi 2002). KATR is based on DATR, a formal language for representing lexical knowledge designed and implemented by Roger Evans and Gerald Gazdar (Evans & Gazdar 1989). Our information about SY is primarily due to the expertise of the second author.

This research is part of a larger effort aimed at elucidating the morphological structure of natural languages. In particular, we are interested in identifying the ways in which default-inheritance relations describe a language's morphology as well as the theoretical relevance of the traditional notion of principal parts. To this end, we have applied similar techniques to Hebrew (Finkel & Stump 2007), Latin (Finkel & Stump to appear, 2009b), and French (Finkel & Stump to appear, 2009a).

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 25–31,
Athens, Greece, 31 March 2009.

1.1 Benefits

As we demonstrate below, the realizational approach leads to a KATR theory that provides a clear picture of the morphology of SY verbs. Different audiences might find different aspects of it attractive.

- A **linguist** can peruse the theory to gain an appreciation for the structure of SY verbs, with all exceptional cases clearly marked either by morphophonological diacritics or by rules of sandhi, which are segregated from all the other rules.
- A **teacher** of the language can use the theory as a foundation for organizing lessons in morphology.
- A **student** of the language can suggest verb roots and use the theory to generate all the appropriate forms, instead of locating the right paradigm in a book and substituting consonants.

2 SY phonetics

SY has 18 consonants (*b, d, f, g, gb, h, j, k, l, m, n, p, r, s, ṣ, t, w, y*), 7 simple vowels (*a, e, ẹ, i, o, ọ, u*), 5 nasalized vowels (*an, en, in, ọn, un*), and 2 syllabic nasals (*m, n*). SY has 3 phonologically contrastive tones: High, Mid and Low. Phonetically, there are also two tone variants, rising and falling (Laniran & Clements 2003). SY orthography employs two transcription formats for these tones. In one format, the two tones are marked on one vowel. For example, the vowel *a* with a low tone followed by a high tone is written as *ǎ* and with a high tone followed by a low tone as *â*. This paper follows the alternative orthography, in which each tone is carried by exactly one vowel. We write *ǎ* as *àá* and *â* as *áà*.

3 A Realizational KATR theory for SY

The purpose of the KATR theory described here is to generate verb forms for SY, specifically, the realizations of all combinations of the morphosyntactic properties of tense (present, continuous, past, future), polarity (positive, negative), person (1, 2 older, 3 older, 2 not older, 3 not older), number (singular, plural), and strength (normal, emphatic). The combinations form a total of 160 **morphosyntactic property sets** (MPSs).

Our analysis posits that SY verbs consist of a sequence of morphological formatives, arranged in six slots:

- Person, which realizes the person and number but is also influenced by tense and polarity,
- Negator marker 1, which appears only in the negative, but is slightly influenced by person and number,
- Tense, which realizes the tense, influenced by polarity,
- Negator marker 2, which appears only in the negative, influenced by tense,
- Stem, which realizes the verb's lexeme,
- Ending, which appears only for emphatic verbs.

Unlike many other languages, SY does not distinguish conjugations of verbs, making its KATR theory simpler than ones for languages such as Latin and Hebrew. However, the tonality of SY adds a small amount of complexity.

A theory in KATR is a network of **nodes**. The network of nodes constituting SY verb morphology is very simple: every lexeme is represented by a node that specifies its stem and then refers to the node `Verb`. The node `Verb` refers to nodes for each of the slots. We use rules of Sandhi as a final step before emitting verb forms.

Each of the nodes in a theory houses a set of **rules**. We represent the verb *mún* 'take' by a node:

```
Take:
1     <stem> = m ún
2      = Verb
```

The node, named `Take`, has two rules, which we number for discussion purposes only. KATR syntax requires that a node be terminated by a single period (full stop), which we omit here. Our convention is to name the node for a lexeme by a capitalized English word (here `Take`) representing its meaning.

Rule 1 says that a query asking for the stem of this verb should produce a two-atom result containing `m` and `ún`. Rule 2 says that all other queries are to be referred to the `Verb` node, which we introduce below.

A **query** is a list of atoms, such as `<stem>` or `<normal positive past 3Older`

sg>, addressed to a node such as Take. In our theory, the atoms in queries either represent **morphological formatives** (such as stem) or **morphosyntactic properties** (such as 3Older and sg).

A query addressed to a given node is matched against all the rules housed at that node. A rule **matches** if all the atoms on its left-hand side match the atoms in the query. A rule can match even if its atoms do not exhaust the entire query. In the case of Take, the query <stem past> is matched by Rules 1 and 2; the query <positive past> is only matched by Rule 2.

Left-hand sides expressed with **path notation** (<pointed brackets>) only match if their atoms match an initial substring of the query. Left-hand sides expressed with **set notation** ({braces}) match if their atoms are all expressed, in whatever position, in the query. We usually use set notation for queries based on morphological formatives and morphosyntactic properties, where order is insignificant.

When several rules match, KATR picks the best match, that is, the one whose left-hand side "uses up" the most of the query. This choice embodies Pāṇini's principle, which entails that if two rules are applicable, the more restrictive rule applies, to the exclusion of the more general rule. We sometimes speak of a rule's **Pāṇini precedence**, which is the cardinality of its left-hand side. If a node in a KATR theory houses two applicable rules with the same Pāṇini precedence, we consider that theory malformed.

In our case, Rule 2 of Take only applies when Rule 1 does not apply, because Rule 1 is always a better match if it applies at all. Rule 2 is called a **default rule**, because it applies by default if no other rule applies. Default rules define a hierarchical relation among some of the nodes in a KATR theory.

KATR generates output based on queries directed to nodes representing individual lexemes. Since these nodes, such as Take, are not referred to by other nodes, they are called **leaves**, as opposed to nodes like Verb, which are called **internal nodes**. The KATR theory itself indicates the list of queries to be addressed to all leaves. Here is the output that KATR generates for several queries directed to the Take node.

normal,positive,present,1,sg
mo mún

normal,positive,present,1,pl
a mún

normal,positive,present,2Older,sg
ẹ mún

normal,positive,present,2Older,pl
ẹ mún

normal,positive,present,3Older,sg
wọ́n mún

normal,positive,present,3Older,pl
wọ́n mún

normal,positive,present,2NotOlder,sg
o mún

normal,positive,present,2NotOlder,pl
ẹ mún

normal,positive,present,3NotOlder,sg
ó mún

normal,positive,present,3NotOlder,pl
wọ́n mún

normal,positive,past,2NotOlder,sg
o ti mún

normal,positive,continuous,2NotOlder,sg
ò ńmún

normal,positive,future,2NotOlder,sg
o óò mún

normal,negative,present,2NotOlder,sg
o (k)ò mún

normal,negative,past,2NotOlder,sg
o (k)ò tî mún

normal,negative,continuous,2NotOlder,sg
o (k)ò mún

normal,negative,future,2NotOlder,sg
o (k)ò ní (kìóò) mún

emphatic,positive,present,2NotOlder,sg
o múnun

emphatic,positive,past,2NotOlder,sg
o ti múnun

The rule for Take illustrates the strategy we term **provisioning** (Finkel & Stump 2007): It provides information (here, the letters of the verb's stem) needed by a more general node (here, Verb).

3.1 The Verb node

We now turn to the Verb node, to which the Take node refers.

```
Verb:
1    {continuous negative} = <present
       negative>
2    {} = Person Negator1 Tense Negator2
       , "<stem>" Ending
```

Rule 1 of Verb reflects the continuous negative to the present negative, because they have identical forms.

Rule 2 is a default rule that composes the surface form by referring to a node for each slot except the stem. This rule directs an query that does not satisfy Rule 1 to each of the nodes mentioned. In this way, the theory computes values for each of the slots that represent the morphological formatives. The KATR phrase "<stem>" directs a new query to the original node (in our case, Take), which has provisioned information about the stem (in our case, m ún). The comma in the right-hand side of rule 2 is how we represent a word division; our post-processing removes ordinary spaces.

3.2 Auxiliary nodes

The Verb node invokes several auxiliary nodes to generate the surface forms for each slot.

```
Person:
1     {1 sg} = mo
2     {1 sg negative} = mi
3     {1 sg future} = m
4     {1 pl} = a
5     {2Older} = ẹ
6     {2Older continuous} = ẹ̀
7     {2Older continuous pl} = w ọ́n
8     {3Older positive !future} = w ọ́n
9     {3Older} = w ọn
10    {2NotOlder sg} = o
11    {2NotOlder pl} = ẹ
12    {2NotOlder continuous sg} = ò
13    {2NotOlder continuous pl} = ẹ̀
14    {3NotOlder} = ó
15    {3NotOlder negative sg} =
16    {3NotOlder future} = yí
17    {3NotOlder pl ++} = <3Older>
```

Generally, the Person slot depends on person and number, but it depends to a small extent on polarity and tense. For example, the exponence[1] of 1 sg is *m*, but it takes an additional vowel in the negative and the non-future positive. On the other hand, the exponence of 1 pl is always *a*. Rule 8 applies to tenses other than future, as marked by the notation !future; in the future, the more general Rule 9 applies. Rule 17 reflects any query involving 3NotOlder pl to the same node (Person) and 3Older forms, to which it is identical. The ++ notation increases the Pāṇini precedence of this rule so that it applies in preference to Rules 15 and 16, even if one of them should apply.

```
Negator1:
1     {negative} = , (k)ò
2     {negative 3NotOlder sg} = kò
3     {} =
```

The first negation slot introduces the exponence *ò* for negative forms (Rules 1 and 2) and the null exponence for positive forms. In most situations, this vowel starts a new word (represented by the comma), and careful speech may place an optional *k* before the vowel (represented by the parenthetical *k*); in 3NotOlder sg, this consonant is mandatory.

```
Tense:
1     {} =
2     {past} = , t i
3     {continuous positive} = , ń -
4     {future positive} = , óò
5     {future 1 sg positive} = , àá
6     {future 3NotOlder positive} =
         <future 3Older positive>
```

The Tense slot is usually empty, as indicated by Rule 1. However, for both negative and positive past, the word *ti* appears here. In the positive continuous, the following slot (the stem) is prefixed by *ń*. We use the hyphen (−) to remove the following word break by a spelling rule (shown later). Similarly, future positive forms have a tense marker, with a special form for 1 sg. As often happens, the 3NotOlder form reflects to the 3Older form.

```
Negator2:
1     {future negative} = , ní
2     {past negative} = ´  ì
3     {} =
```

The second negator slot adds the word *ní* in the future (Rule 1). In the past (Rule 2), it changes the tone of the tense slot from *ti* to *tî*. In all other cases, Rule 3 gives a null default. Rule 2 follows an assumption that tone and vowel can be specified independently in SY; without this assumption, this slot would be more cumbersome to specify. Such floating tones are in keeping with theories of autosegmental phonology (Goldsmith 1976) and are seen in other Niger-Congo languages, such as Bambara (Mountford 1983).

```
Ending:
1     {} =
2     {emphatic} = ↓
```

The Ending slot is generally null (Rule 1), but in emphatic forms, it reduplicates the final vowel with a mid tone, unless the vowel already has a

[1] An **exponence** is a surface form or part of a surface form, that is, the way a given lexeme appears when it is attached to morphosyntactic properties.

mid tone, in which case the tone becomes low. (We disagree slightly with Akinlabi and Liberman, who suggest that this suffix is in low tone except after a low tone, in which case it becomes mid (Akinlabi & Liberman 2000).) For this case, we introduce a *jer*[2], represented by "↓", for postprocessing in the Sandhi phase, discussed below. Such forms are important as a way to simplify presentation, covering many cases in one rule. When we tried to develop a SY KATR theory without a *jer*, we needed to separate the stem of each word into onset and coda so we could repeat the coda in emphatic forms, but we had no clear way to indicate the regular change in tone. The *jer* accomplishes both reduplication and tone change with a single, simple mechanism. It also suggests that the emphatic ending is really a matter of tone Sandhi, not a matter of default inheritance.

4 Postprocessing: Sandhi, Spelling and Alternatives

After the rules produce a surface form, we postprocess that form to account for Sandhi (language-specific rules dictating sound changes for euphony), spelling conventions, and alternative exponence. We have only one Sandhi matter to account for, the *jer* "↓". We accomplish this postprocessing with these rules:

```
1 #vars $vowel:  a e ẹ i o ọ u .
2 #vars $tone:   ´  ̀  .
3 #sandhi $vowel ↓ => $1 $1 ̀  .
4 #sandhi $vowel $tone ↓ => $1 $2 $1 .
5 #sandhi $vowel n ↓ => $1 n $1 ̀  n .
6 #sandhi $vowel $tone n ↓ => $1 $2 n $1
    n .
```

The first two lines introduce shorthands so we can match arbitrary vowels and tone marks. Sandhi rules are applied in order, although in this case, at most one of them will apply to any surface form.

Rules 3–6 represent tone Sandhi by showing how to replace intermediate surface strings with final surface strings. Each rule has a left-hand side that is to be replaced by the information on the right-hand side. Numbers like `$1` on the right-hand side refer to whatever a variable (in this case, the first variable) on the left-hand side has matched.

[2]A **jer**, also called a **morphophoneme**, is a phonological unit whose phonemic expression depends on its context. It is an intermediate surface form that is to be replaced in a context-sensitive way during postprocessing.

Rule 3 indicates that if we see a vowel without a tone mark (indicating mid tone) followed by the *jer*, we replace it with the vowel (represented by `$1`) repeated with low tone. This specification follows our assumption that tone and vowel may be treated independently. Rule 4 indicates that a vowel followed by a tone mark and the *jer* is repeated with mid tone (without a mark). Rules 5 and 6 are similar, but they deal with nasalized vowels.

There is one spelling rule to remove word breaks that would otherwise be present. We have used "–" to indicate that a word break should disappear. We use the following rule to enforce this strategy:

```
#sandhi - , => .
```

That is, a hyphen before a comma removes both.

SY allows the negative future forms *(k)ò ní* and *kò ní* to be expressed instead as *kìóò*. We provide rules of alternation for this purpose:

```
#alternative \(k\)ò , ní => kìóò .
#alternative kò , ní => kìóò .
```

These alternation rules effectively collapse the three slots, Negator1, Tense, and Negator2 into a single exponence.

5 Processing

The interested reader may see the entire SY theory and run it through our software by directing a browser to `http://www.cs.uky.edu/~raphael/KATR.html`, where theories for several other languages can also be found. Our software runs in several steps:

1. A Perl script converts the KATR theory into two files: a Prolog representation of the theory and a Perl script for post-processing.

2. A Prolog interpreter runs a query on the Prolog representation.

3. The Perl post-processing script treats the Prolog output.

4. Another Perl script either generates a textual output for direct viewing or HTML output for a browser.

This software is available from the first author under the GNU General Public License (GPL).

6 Discussion and Conclusions

This exercise demonstrates that the realizational approach to defining language morphology leads to an effective description of SY verbs. We have applied language-specific knowledge and insight to create a default inheritance hierarchy that captures the morphological structure of the language, with slots pertaining to different morphosyntactic properties. In particular, our KATR theory nicely accounts for the slot structure of SY verbs, even though most slots are dependent on multiple morphosyntactic properties, and we are easily able to deal with the tone shifts introduced by the emphatic suffix.

This work is not intended to directly address the problem of parsing, that is, converting surface forms to pairings of lexemes with morphosyntactic properties. We believe that our KATR theory for SY correctly covers all verb forms, but there may certainly be exceptional cases that do not follow the structures we have presented. Such cases are usually easy to account for by introducing information in the leaf node of such lexemes. Further, this work is not in the area of automated learning, so questions of precision and ability to deal with unseen data are not directly relevant.

We have constructed the SY theory in KATR instead of DATR for several reasons.

- We have a very fast KATR implementation, making for speedy prototyping and iterative improvement in morphological theories. This implementation is capable of taking standard DATR theories as well.

- KATR allows bracket notation ({ and }) on the left-hand side of rules, which makes it very easy to specify morphosyntactic properties for queries in any order and without mentioning those properties that are irrelevant to a given rule. Rules in DATR theories tend to have much more complicated left-hand sides, obscuring the morphological rules.

- KATR has a syntax for Sandhi that separates its computation, which we see as postprocessing of surface forms, from the application of morphological rules. It is possible to write rules for Sandhi in DATR, but the rules are both unpleasant to write and difficult to describe.

As we have noted elsewhere (Finkel & Stump 2007), writing KATR specifications requires considerable effort. Early choices color the structure of the resulting theory, and the author must often discard attempts and rethink how to represent the target morphology. The first author, along with Gregory Stump, has built KATR theories for verbs in Hebrew, Slovak, Polish, Spanish, Irish, Shughni (an Iranian language of the Pamir) and Lingala (a Bantu language of the Congo), as well as for parts of Hungarian, Sanskrit, and Pali.

Acknowledgments

We would like to thank Gregory Stump, the first author's collaborator in designing KATR and applying it to many languages. Lei Shen and Suresh Thesayi were instrumental in implementing our Java™ version of KATR. Nancy Snoke assisted in implementing our Perl/Prolog version.

Development of KATR was partially supported by the US National Science Foundation under Grants IIS-0097278 and IIS-0325063 and by the University of Kentucky Center for Computational Science. The second author is supported by Science Foundation Ireland Grant 05/IN/I886 and Marie Curie Grant MTKD-CT-2006-042563. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Adéwọlé, L. O. (1988). *The categorical status and the function of the Yorùbá auxiliary verb with some structural analysis in GPSG*, PhD thesis, University of Edinburgh, Edinburgh.

Akinlabi, A. & Liberman, M. (2000). The tonal phonology of Yoruba clitics, *in* B. Gerlach & J. Grizenhout (eds), *Clitics in phonology, morphology and syntax*, John Benjamins Publishing Company, Amsterdam/Philadelphia, pp. 64–66.

Anderson, S. R. (1992). *A-morphous morphology*, Cambridge University Press.

Corbett, G. G. & Fraser, N. M. (1993). Network Morphology: A DATR account of Russian nominal inflection, *Journal of Linguistics* **29**: 113–142.

Evans, R. & Gazdar, G. (1989). Inference in DATR, *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, pp. 66–71.

Finkel, R., Shen, L., Stump, G. & Thesayi, S. (2002). KATR: A set-based extension of DATR, *Technical Report 346-02*, University of Kentucky Department of Computer Science, Lexington, KY. `ftp://ftp.cs.uky.edu/cs/techreports/346-02.pdf`.

Finkel, R. & Stump, G. (2007). A default inheritance hierarchy for computing Hebrew verb morphology, *Literary and Linguistic Computing* **22**(2): 117–136. `dx.doi.org/10.1093/llc/fqm004`.

Finkel, R. & Stump, G. (to appear, 2009a). Stem alternations and principal parts in French verb inflection, *Cascadilla Proceedings Project* .

Finkel, R. & Stump, G. (to appear, 2009b). What your teacher told you is true: Latin verbs have four principal parts, *Digital Humanities Quarterly* .

Goldsmith, J. A. (1976). *Autosegmental phonology*, PhD thesis, Massachusetts Institute of Technology, Boston, MA.

Gordon, R. G. (2005). *Ethnologue: Languages of the World*, 15^{th} edn, SIL International, Dallas, Texas.

Laniran, Y. O. & Clements, G. N. (2003). Downstep and high rising: interacting factors in Yorùbá tone production, *J. of Phonetics* **31**(2): 203 – 250.

Matthews, P. H. (1972). *Inflectional morphology*, Cambridge University Press.

Mountford, K. W. (1983). *Bambara declarative sentence intonation*, PhD thesis, Indiana University, Bloomington, IN.

Stump, G. T. (2001). *Inflectional morphology*, Cambridge University Press, Cambridge, England.

Zwicky, A. M. (1985). How to describe inflection, *Proceedings of the 11th annual meeting of the Berkeley Linguistics Society*, pp. 372–386.

Using Technology Transfer to Advance Automatic Lemmatisation for Setswana

Hendrik J. Groenewald
Centre for Text Technology (CTexT)
North-West University
Potchefstroom 2531, South Africa
handre.groenewald@nwu.ac.za

Abstract

South African languages (and indigenous African languages in general) lag behind other languages in terms of the availability of linguistic resources. Efforts to improve or fast-track the development of linguistic resources are required to bridge this ever-increasing gap. In this paper we emphasize the advantages of technology transfer between two languages to advance an existing linguistic technology/resource. The advantages of technology transfer are illustrated by showing how an existing lemmatiser for Setswana can be improved by applying a methodology that was first used in the development of a lemmatiser for Afrikaans.

1 Introduction

South Africa has eleven official languages. Of these eleven languages, English is the only language for which ample HLT resources exist. The rest of the languages can be classified as so-called "resource scarce languages", i.e. languages for which few digital resources exist. However, this situation is changing, since research in the field of Human Language Technology (HLT) has enjoyed rapid growth in the past few years, with the support of the South African Government. Part of this development is a strong focus on the development of core linguistic resources and technologies. One such a technology/resource is a lemmatiser.

The focus of this article is on how technology transfer between two languages can help to improve and fast track the development of an existing linguistic resource. This is illustrated in the way that an existing lemmatiser for Setswana is improved by applying the method that was first used in the development of a lemmatiser for Afrikaans.

The rest of this paper is organised as follows: The next section provides general introductory information about lemmatisation. Section 3 provides specific information about lemmatisation and the concept of a lemma in Setswana. Section 4 describes previous work on lemmatisation in Afrikaans. Section 5 gives an overview of memory based learning (the machine learning techniques used in this study) and the generic architecture developed for machine learning based lemmatisation. Data requirements and the data preparation process are discussed in Section 6. The implementation of a machine learning based lemmatiser for Setswana is explained in Section 7, while some concluding remarks and future directions are provided in Section 8.

2 Lemmatisation

Automatic Lemmatisation is an important process for many applications of text mining and natural language processing (NLP) (Plisson *et al*, 2004). Within the context of this research, lemmatisation is defined as a simplified process of morphological analysis (Daelemans and Strik, 2002), through which the inflected forms of a word are converted/normalised under the lemma or base-form.

For example, the grouping of the inflected forms 'swim', 'swimming' and 'swam' under the base-form 'swim' is seen as an instance of lemmatisation. The last part of this definition applies to this research, as the emphasis is on recovering the base-form from the inflected form of the word. The base-form or lemma is the simplest form of a word as it would appear as headword in a dictionary (Erjavec and Džeroski, 2004).

Lemmatisation should, however, not be confused with stemming. Stemming is the process whereby a word is reduced to its stem by the removal of both inflectional and derivational morphemes (Plisson *et al*, 2004). Stemming can thus be viewed as a "greedier" process than lemmatisation, because a larger number of morph-

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 32–37,
Athens, Greece, 31 March 2009.

emes are removed by stemming than lemmatisation. Given this general background, it would therefore be necessary to have a clear understanding of the inflectional affixes to be removed during the process of lemmatisation for a particular language.

There are essentially two approaches that can be followed in the development of lemmatisers, namely a rule-based approach (Porter, 1980) or a statistically/data-driven approach (Chrupala, 2006). The rule-based approach is a traditional method for stemming/lemmatisation (i.e. affix stripping) (Porter 1980; Gaustad and Bouma, 2002) and entails the use of language-specific rules to identify the base-forms (i.e. lemmas) of word forms.

3 Lemmatisation in Setswana

The first automatic lemmatiser for Setswana was developed by Brits (2006). As previously mentioned, one of the most important aspects of developing a lemmatiser in any language is to define the inflectional affixes that need to be removed during the transformation from the surface form to the lemma of a particular word. In response to this question, Brits (2006) found that only stems (and not roots) can act independently as words and therefore suggests that only stems should be accepted as lemmas in the context of automatic lemmatisation for Setswana.

Setswana has seven different parts of speech. Brits (2006) indicated that five of these seven classes cannot be extended by means of regular morphological processes. The remaining two classes, namely nouns and verbs, require the implementation of alternation rules to determine the lemma. Brits (2006) formalized rules for the alterations and implemented these rules as regular expressions in FSA 6 (Van Noord, 2002), to create finite state transducers. These finite state transducers generated C++ code that was used to implement the Setswana lemmatiser. This lemmatiser achieved a linguistic accuracy figure of 62,17%, when evaluated on an evaluation subset of 295 randomly selected Setswana words. Linguistic accuracy is defined as the percentage of words in the evaluation set that was correctly lemmatised.

4 *Lia*: Lemmatiser for Afrikaans

In 2003, a rule-based lemmatiser for Afrikaans (called *Ragel* – "*Reëlgebaseerde Afrikaanse Grondwoord- en Lemma-identifiseerder*") [Rule-Based Root and Lemma Identifier for Afrikaans] was developed at the North-West University (RAGEL, 2003). *Ragel* was developed by using traditional methods for stemming/lemmatisation (i.e. affix stripping) (Porter, 1980; Kraaij and Pohlmann, 1994) and consists of language-specific rules for identifying lemmas. Although no formal evaluation of *Ragel* was done, it obtained a disappointing linguistic accuracy figure of only 67% in an evaluation on a random 1,000 word data set of complex words. This disappointing result motivated the development of another lemmatiser for Afrikaans.

This "new" lemmatiser (named *Lia* – "Lemma-identifiseerder vir Afrikaans" [Lemmatiser for Afrikaans]) was developed by Groenewald (2006). The difference between *Ragel* and *Lia* is that *Lia* was developed by using a so-called data driven machine learning method. Machine learning requires large amounts of annotated data. For this purpose, a data set consisting of 73,000 lemma-annotated words were developed. *Lia* achieves a linguistic accuracy figure of 92,8% when trained on this data set. This result confirms that the machine learning based approach outperforms the rule-based approach for lemmatisation in Afrikaans.

The increased linguistic accuracy figure obtained with the machine learning based approach motivated the research presented in this paper. Since *Ragel* and the rule-based Setswana lemmatiser obtained comparable linguistic accuracy figures, the question arises whether the application of machine learning techniques, together with the methodology and architecture developed for *Lia*, can also be utilised to improve on the linguistic accuracy figure obtained by the Setswana rule-based lemmatiser.

5 Methodology

5.1 Memory Based Learning

Memory based learning (Aha *et al*, 1991) is based on the classical *k*-NN classification algorithm. *k*-NN has become known as a powerful pattern classification algorithm (Daelemans *et al*, 2007), and is considered the most basic instance-based algorithm. The assumption here is that all instances of a certain problem correspond to points in the *n*-dimensional space (Aha *et al*, 1991). The nearest neighbours of a certain instance are computed using some form of distance metric (X,Y). This is done by assigning the most frequent category within the found set of most

similar example(s) (the *k*-nearest neighbours) as the category of the new test example. In case of a tie amongst categories, a tie-breaking resolution method is used.

The memory based learning system on which *Lia* is based, is called TiMBL (Tilburg Memory-Based Learner). TiMBL was specifically developed with NLP tasks in mind, but it can be used successfully for classification tasks in other domains as well (Daelemans *et al*, 2007).

5.2 Architecture

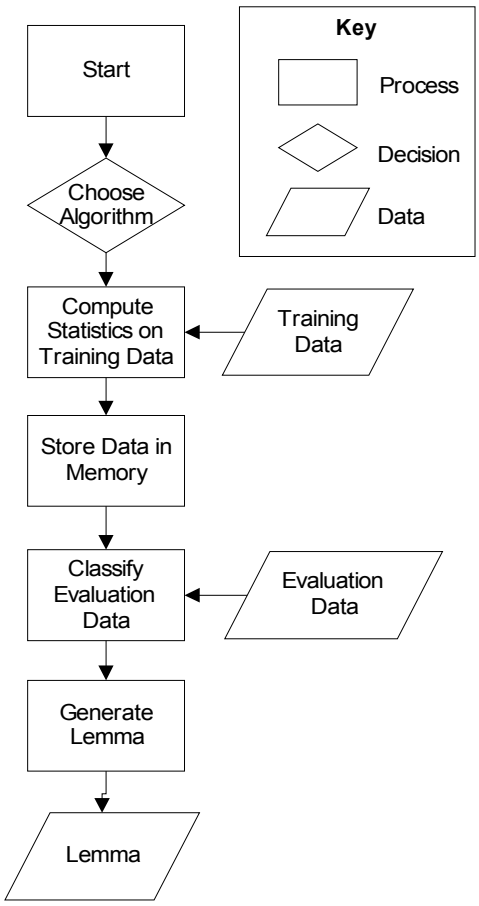


Figure 1. Generic Architecture of the Machine Learning Based Lemmatiser.

The architecture presented in this subsection was first developed and implemented for *Lia*, the machine learning based lemmatiser for Afrikaans. This same architecture was used for the development of the machine learning based lemmatiser for Setswana. The first step in this "generic" architecture consists of training the system with data. During this phase, the training data is examined and various statistical calculations are computed that aid the system during classification. This training data is then stored in memory as sets of data points. The evaluation instance(s) are then presented to the system and their class is computed by interpolation to the stored data points according to the selected algorithm and algorithm parameters. The last step in the process consists of generating the correct lemma(s) of the evaluation instance(s), according to the class that was awarded during the classification process. The generic architecture of the machine learning based lemmatiser is illustrated in Figure 1.

6 Data

6.1 Data Size

A negative aspect of the Machine Learning method for developing a lemmatiser is that a large amount of lemma-annotated training data is required. Currently, there is a data set available that contains only 2,947 lemma-annotated Setswana words. This is the evaluation data set constructed by Brits (2006) to evaluate the performance of the rule-based Setswana lemmatiser. A data set of 2,947 words is considered to be very small in machine learning terms.

6.2 Data Preparation

Memory based learning requires that lemmatisation be performed as a classification task. The training data should therefore consist of feature vectors with assigned class labels (Chrupala, 2006). The feature vectors for each instance consist of the letters of the inflected word. The class labels contain the information required to transform the involved word form from the inflected form to the lemma.

The class labels are automatically derived by determining the character string (and the position thereof) to be removed and the possible replacement string during the transformation from word-form to lemma. This is determined by firstly obtaining the longest common substring between the inflected word and the manually identified lemma. Once the longest common substring is known, a comparison of the remaining strings in the inflected word form and the lemma indicates the strings that need to be removed (as well as the possible replacement strings) during the transformation from word form to lemma. The positions of the character string to be removed are annotated as *L* (left) or *R* (right).

If a word-form and its lemma are identical, the class awarded will be *"0"*, denoting that the word should be left in the same form. This annotation scheme yields classes like in column four of Table 1.

Inflected Word-Form	Manually Identified Lemma	Longest Common Substring	Automatically Derived Class
matlhwao	letlhwao	hwao	Lma>le
menoganya	menoga	menoga	Rya>
itebatsa	lebala	ba	Lit>lRtsa>la

Table 1. Data Preparation and Classes.

For example, Table 1 shows that the class of "matlhwao" is *Lma>le*. This means that the string "ma" needs to be replaced by the string "le" (at the left hand side of the word) during the transformation from the inflected form "matlhwao" to the lemma "letlhwao". Accordingly, the class of the word "menoganya" is *Rya>*, denoting the string "ya" should be removed at the right-hand side of the inflected form during lemmatisation. In this particular case, there is no replacement string. Some words like "itebatsa" undergo alterations to both sides of the inflected form during lemmatisation. The class *Lit>lRtsa>la* indicates that the string "it" must be replaced at the left-hand side of the word with the letter "l", while the string "tsa" should be replaced with the string "la" at the right-hand side of the word.

An example of the training of data of the lemmatiser is shown in Figure 2. The data is presented in C4.5 format (Quinlan, 1993) to the memory based learning algorithm, where each feature is separated by a comma. The algorithm requires that every instance must have the same number of features. In order to achieve this, it was decided that each instance should contain 20 features. 20 features were chosen, since less than 1% of the words in the data contained more than 20 letters. All instances were formatted to contain 20 features by adding underscores to the words that contained less than 20 features. The result of this process is displayed in Figure 2.

_,_,_,_,_,_,_,_,_,_,t,s,o,g,a,t,s,o,g,a,0
_,_,_,_,_,_,_,_,_,_,e,d,i,m,o,l,a,n,y,a,Rnya>
_,_,_,_,_,_,_,_,_,_,_,_,_,_,d,i,n,y,e,p,o,Ldi>
_,_,_,_,_,_,_,_,_,_,t,s,i,s,e,d,i,t,s,e,Ltsisedi>Rse>la

Figure 2. Training Data in C4.5 Format.

7 Implementation

Each of the 2,947 lemma-annotated words in the evaluation data of the rule-based Setswana lemmatiser was formatted in C4.5 format. The data was then split up further into a training data set, consisting of 90% of all the data, with an evaluation set consisting of 10% of all the data. A machine learning based lemmatiser was trained (by utilising default parameter settings) and evaluated with these two datasets. This lemmatiser obtained an accuracy figure of 46.25%. This is a disappointing result when compared to the linguistic accuracy figure of 62.71% obtained with the rule-based Setswana lemmatiser when evaluated on the same data set. Algorithmic parameter optimisation with *PSearch* (Groenewald, 2008) resulted in an improved accuracy figure of 58.98%. This represents an increase of 12.73%, but is still less than the accuracy figure obtained by the rule-based lemmatiser.

Error analysis indicated that in some cases the class predicted by TiMBL is conspicuously wrong. This is evident from instances shown in Table 2, where the assigned classes contain strings that need to be removed that is not present in the inflected forms.

Inflected Word	Correct Class	Assigned Class
tlamparele	*Re>a*	Lmm>bRele>a
phologileng	*Rileng>a*	Regileng>a

Table 2. Instances with Incorrectly Assigned Classes.

Inflected Word	Assigned Class	Class Distribution
tlamparele	*Lmm>bRele>a*	0 0.934098 Re>a 1.82317 Rele>a 0.914829 Lmm>bRele>a 1.96103
phologileng	*Regileng>a*	Rileng>a 3.00014 Relang>a 1.24030 Regileng>a 4.20346

Table 3. Instances Containing Additional Class Distribution Information.

For example, the class assigned to the second instance in Table 2, is *Regileng>a*. This means that the string "egileng" must be replaced with the

character “a” at the right-hand side of the word. However, the inflected word “phologileng” does not contain the string “egileng”, which means that the assigned class is sure to be incorrect. This problem was overcome by utilizing the TiMBL option (+v db) that adds class distribution in the nearest neighbour set to the output file. The result of this is an additional output that contains the class distribution information shown in Table 3. The class distribution information contains the nearest classes with their associated distances from the involved evaluation instance.

A post-processing script that automatically recognises this type of incorrectly assigned class and replaces the incorrect class with the second most likely class (according to the class distribution) was developed. The result of this was a further increase in accuracy to 64.06%. A summary of the obtained results is displayed in Table 4.

Method	Linguistic Accuracy
Rule-based	62.17%
Machine Learning with default parameter settings	46.25%
Machine Learning with optimised parameter settings	58.9%
Machine Learning with optimised parameter settings and class distributions	64.06%.

Table 4. Summary of Results.

8 Conclusion

The best results obtained by the machine learning based Setswana lemmatiser was a linguistic accuracy figure of 64.06%. This represents an increase of 1.9% on the accuracy figure obtained by the rule-based lemmatiser. This seems to be a small increase in accuracy compared to the 25.8% increase obtained when using a machine learning based method for Afrikaans lemmatisation. The significance of this result becomes evident when considering the fact that it was obtained by training the machine learning based Setswana lemmatiser with a training data set consisting of only 2,652 instances. This data set is very small in comparison with the 73,000 instances contained in the training data of *Lia.*

The linguistic accuracy figure of 64.06% furthermore indicates that a machine learning based lemmatiser for Setswana that yields better results than a rule-based lemmatiser can be developed with a relatively small data set. We are confident that further increases in the linguistic accuracy figure will be obtained by enlarging the training data set. Future work will therefore entail the employment of bootstrapping techniques to annotate more training data for improving the linguistic accuracy of the machine learning based Setswana lemmatiser.

The most important result of the research presented in this paper is, however, that existing methodologies and research can be applied to fast-track the development of linguistic resources or improve existing linguistic resources for resource-scarce languages, a result that is especially significant in the African context.

Acknowledgments

The author wishes to acknowledge the work of Jeanetta H. Brits, performed under the supervision of Rigardt Pretorius and Gerhard B. van Huyssteen, on developing the first automatic lemmatiser for Setswana.

References

David W. Aha, Dennis Kibler and Marc K. Albert. 1991. Instance-Based Learning Algorithms. *Machine Learning*, 6:37-66.

Jeanetta H. Brits. 2006. *Outomatiese Setswana Lemma-identifisering ‘Automatic Setswana Lemmatisation’*. Master’s Thesis. North-West University, Potchefstroom, South Africa.

Gregorz Chrupala. 2006. Simple Data-Driven Context-Sensitive Lemmatization. *Proceedings of SEPLN 2006*.

Walter Daelemans, Antal Van den Bosch, Jakub Zavrel and Ko Van der Sloot. 2007. *TiMBL: Tilburg MemoryBased Learner*. Version 6.1, Reference Guide. ILK Technical Report 07-03.

Walter Daelemans and Helmer Strik. 2002. Actieplan Voor Het Nederlands in de Taal- en Spraaktechnologie: Prioriteiten Voor Basisvoorzieningen. *Report for the Nederlandse Taalunie*. Nederlandse Taalunie.

Tomaž Erjavec and Saso Džeroski. 2004. Machine Learning of Morphosyntactic Structure: Lemmatising Unknown Slovene Words. *Applied Artificial Intelligence*, 18(1):17-40.

Tanja Gaustad and Gosse Bouma. 2002. Accurate Stemming of Dutch for Text Classification. *Language and Computers,* 45 (1):104-117.

Hendrik J. Groenewald. 2007. *Automatic Lemmatisation for Afrikaans*. Master’s Thesis. North-West University, Potchefstroom, South Africa.

Hendrik J. Groenewald. 2008. *PSearch 1.0.0*. North-West University, Potchefstroom, South Africa.

Wessel Kraaij and Renee Pohlmann. 1994. Porter's Stemming Algorithm for Dutch. *Informatiewetenschap 1994: Wetenschaplike bijdraen aan de derde STINFON Conferentie*. 1(1):167-180.

Joel Plisson, Nada Lavrac and Dunja Mladenić. 2004. A Rule-based Approach to Word Lemmatization. *Proceedings C of the 7th International Multi-Conference Information Society IS 2004*, 1(1):83-86.

Martin Porter. 1980. An Algorithm for Suffix Stripping. *Program,* 14 (3):130-137.

John R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, USA.

RAGEL. 2003. Reëlgebaseerde Afrikaanse Grondwoord- En Lemma-identifiseerder 'Rule-based Afrikaans Stemmer and Lemmatiser. http://www.puk.ac.za/opencms/export/PUK/html/fakulteite/lettere/ctext/ragel.html.> 11 January 2009.

Gertjan Van Noord. 2002. Finite State Utilities. < http://www.let.rug.nl/~vannoord/Fsa/>. 12 January 2009.

Part-of-Speech tagging of Northern Sotho: Disambiguating polysemous function words

Gertrud Faaß[†‡] **Ulrich Heid**[†] **Elsabé Taljard**[‡] **Danie Prinsloo**[‡]
[†] Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Germany
faaszgd@ims.uni-stuttgart.de
heid@ims.uni-stuttgart.de

[‡] University of Pretoria
South Africa
elsabe.taljard@up.ac.za
danie.prinsloo@up.ac.za

Abstract

A major obstacle to part-of-speech (=POS) tagging of Northern Sotho (Bantu, S 32) are ambiguous function words. Many are highly polysemous and very frequent in texts, and their local context is not always distinctive.
With certain taggers, this issue leads to comparatively poor results (between 88 and 92 % accuracy), especially when sizeable tagsets (over 100 tags) are used.
We use the RF-tagger (Schmid and Laws, 2008), which is particularly designed for the annotation of fine-grained tagsets (e.g. including agreement information), and we restructure the 141 tags of the tagset proposed by Taljard et al. (2008) in a way to fit the RF tagger. This leads to over 94 % accuracy. Error analysis in addition shows which types of phenomena cause trouble in the POS-tagging of Northern Sotho.

1 Introduction

In this paper, we discuss issues of the part-of-speech (POS) tagging of Northern Sotho, one of the eleven official languages of South Africa, spoken in the North-east of the country. Northern Sotho is a Bantu language belonging to the Sotho family (Guthrie, 1967: S32). It is written disjunctively (contrary to e.g. Zulu), i.e. certain morphemes appear as character strings separated by blank spaces. It makes use of 18 noun classes (1, 1a, 2, 2b, 3 to 10, 14, 15, and the locative classes 16, 17, 18, *N-*, which may be summarized as LOC for their identical syntactic features). A concordial system helps to verify agreement or resolve ambiguities.

We address questions of the ambiguity of function words, in the framework of an attempt to use "standard" European-style statistical POS taggers on Northern Sotho texts.

In the remainder of this section, we briefly discuss our objectives (section 1.1) and situate our work within the state of the art (section 1.2). Section 2 deals with the main issues at stake, the handling of unknown open class words, and the polysemy of Northern Sotho function words. In section 3, we discuss our methodology, summarizing the tagset and the tagging technique used, and reporting results from other taggers. Section 4 is devoted to details of our own results, the effects of the size of training material (4.2), the effects of polysemy and reading frequency (4.3), and it includes a discussion of proposals for quality improvement (Spoustová et al., 2007). We conclude in section 5.

1.1 Objectives

The long term perspective of our work is to support information extraction, lexicography, as well as grammar development of Northern Sotho with POS-tagged and possibly parsed corpus data. We currently use the 5.5 million word *University of Pretoria Sepedi Corpus* (PSC, cf. de Schryver and Prinsloo (2000)), as well as a 45,000 words training corpus. We aim at high accuracy in the POS-tagging, and at minimizing the amount of unknown word forms in arbitrary unseen corpora, by using guessers for the open word classes.

1.2 Recent work

A few publications, so far, deal with POS-tagging of Northern Sotho; most prominently, de Schryver and de Pauw (2007) have presented the MaxTag method, a tagger based on Maximum Entropy

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 38–45,
Athens, Greece, 31 March 2009.

Learning (Berger et al., 1996) as implemented in the machine learning package Maxent (Le, 2004). When trained on manually annotated text, it extracts features such as the first and last letter of each word, or the first two and last two letters or the first three and last three letters of each word; it takes the word and the tag preceding and following the item to be tagged, etc., to decide about word/tag probabilities. De Schryver and de Pauw report an accuracy of 93.5 % on unseen data, using a small training corpus of only ca. 10,000 word forms.

Other work is only partly engaged in POS-tagging, e.g. Kotzé's (2008) finite state analysis of the verb complex of Northern Sotho. This study does not cover all parts of speech and can thus not be directly compared with our work. Taljard et al. (2008) and Van Rooy and Pretorius (2003) present tagsets for Northern Sotho and the closely related language Setswana, but they focus on the definition of the tagsets without discussing their automatic application in detail. In (Prinsloo and Heid, 2005), POS-tagging is mentioned as a step in a corpus processing pipeline for Northern Sotho, but no experimental results are reported.

2 Challenges in tagging Northern Sotho

POS-tagging of Northern Sotho and of any disjunctively written Bantu language has to deal especially with two major issues which are consequences of their morphology and their syntax. One is the presence, in any unseen text, of a number of lexical items which are not covered by the lexicon of the tagger ("unknown words"), and the other is an extraordinarily high number of ambiguous function words.

2.1 Unknown words

In Northern Sotho, nouns, verbs and adverbs are open class items; all other categories are closed word classes: their items can be listed. The open classes are characterized in particular by a rich morphology: nouns can form derivations to express diminutives and augmentatives, as well as locative forms, to name just a few. Adding the suffix *-ng* to *toropo* 'town', for example, forms *toropong*, 'in/at/to town'. For verbs, tense, voice, mood and many other dimensions, as well as nominalization, lead to an even larger number of derived items. Prinsloo (1994) distinguishes 18 clusters of verbal suffixes which give rise to over 260 individual derivation types per verb. Only a few of these derivations are highly frequent in corpus text; however, due to productivity, a large number of verbal derivation types can potentially appear in any unseen text.

For tagging, noun and verb derivations show up as unknown items, and an attempt to cover them within a large lexicon will partly fail due to productivity and recursive applicability of certain affixes. The impact of the unknown material on tagging quality is evident: de Schryver and de Pauw (2007) report 95.1 % accuracy on known items, but only 78.9 % on unknowns; this leads to a total accuracy of 93.5 % on their test corpus. We have carried out experiments with a version of the memory-based tagger, MBT (Daelemans et al., 2007), which arrives at 90.67 % for the known items of our own test corpus (see section 3.2), as opposed to only 59.68 % for unknowns.

To counterbalance the effect of unknown items, we use rule-based and partly heuristic guessers for noun and verb forms (cf. Prinsloo et al. (2008) and (Heid et al., 2008)) and add their results to the tagger lexicon before applying the statistical tagger: the possible annotations for all words contained in the text are thus part of the knowledge available to the tagger.

Adverbs are also an open word class in Northern Sotho; so far, we have no tools for identifying them. In high quality tagging, the suggestions of our guessers are examined manually, before they are added to the tagger lexicon.

2.2 Polysemous function words and ambiguity

Function words of Northern Sotho are highly ambiguous, and because of the disjunctive writing system of the language, a number of bound morphemes are written separately from other words.

A single form can have several functions. For example, the token *-a-* is nine-ways ambiguous: it can be a subject concord of noun class 1 or 6, an object concord of class 6, a possessive concord of class 6, a demonstrative of class 6, a hortative or a question particle or a verbal morpheme indicating present tense or past tense (Appendix A illustrates the ambiguity of *-a-* with example sentences). Furthermore, the most polysemous function words are also the most frequent word types in corpora. The highly ambiguous item *go*[1] alone ac-

[1] 11 different functions of *go* may be distinguished: object

counts for over 5 % of all occurrences in our training corpus, where 88 types of function words, with an average frequency of well over 200, make up for about 40 % of all occurrences.

The different readings of the function words are not evenly distributed: some are highly frequent, others are rare. Furthermore, many ambiguous function words appear in the context of other function words; thus the local context does not necessarily disambiguate individual function words. This issue is particularly significant with ambiguities between concords which can have the same function (e.g. object) in different noun classes. As mentioned, *-a-* can be a subject concord of either noun class 1 or 6: though there are some clearcut cases, like the appearance of a noun of class 6 (indicating class 6), or an auxiliary or the conjunction *ge* in the left context (both rather indicating class 1) there still remain a number of occurrences of *-a-* in the corpora only where a broader context, sometimes even information from preceding sentences, may help to disambiguate this item.

Consequently, comparing tagging performance across different tagsets does not give very clear results: if a tagset, like the one used by de Schryver and de Pauw (2007), does not distinguish noun classes, obviously a large number of difficult disambiguation cases does not appear at all (their tagset distinguishes, for example, subject and object concord, but gives no information on noun class numbers). For the lexicographic application we are interested in, and more generally as a preparatory step to chunking or parsing of Northern Sotho texts, an annotation providing information on noun classes is however highly desirable.

3 Methodology

3.1 Tagset

There are several proposals for tagsets to be used for Northern Sotho and related languages. Van Rooy and Pretorius (2003) propose a detailed tagset for Setswana, which is fully in line with the guidelines stated by the EAGLES project, cf. Leech and Wilson (1999). This tagset encodes a considerable number of semantic distinctions in its nominal and verbal tags. In Kotzé's work on the Northern Sotho verb complex, (Kotzé, 2008), a number of POS tags are utilized to distinguish the elements of the verb, however, due to Kotzé's objectives, her classification does not cover other items. De Schryver and de Pauw (2007) use a tagset of only 56 different tags, whereas the proposal by Van Rooy and Pretorius leads to over 100 tags. Finally, Taljard et al. (2008) propose a rather detailed tagset: contrary to the other authors mentioned, they do encode noun classes in all relevant tags, which leads to a total of 141 tags. Furthermore, they encode a number of additional morphosyntactic distinctions on a second level of their tagset, which leads to a total of 262 different classifications of Northern Sotho morphemes.

Our current tagset is inspired by Taljard et al. (2008). However, we disregard some of their second level information for the moment (which in many cases encodes lexical properties of the items, e.g. the subdivision of particles: hortative, question, instrumental, locative, connective, etc.). We use the RF-tagger (Schmid and Laws, 2008) (cf. section 3.3), which is geared towards the annotation of structured tagsets, by separating information which partitions the inventory of forms (e.g. broad word classes) from feature-like information possibly shared by several classes, such as the Sotho noun classes. With this method, we are able to account for Taljard et al.'s (2008) 141 tags by means of only 25 toplevel tags, plus a number of feature-like labels of lower levels. We summarize the properties of the tagsets considered in table 1.

3.2 Training corpus

Our training corpus consists of ca. 45.000 manually annotated word forms, from two text types. Over 20.000 word forms come from a novel of the South African author Oliver K. Matsepe (Matsepe, 1974); over 10.000 forms come from a Ph.D. dissertation by Raphehli M. Thobakgale (Thobakgale, 2005), and another 10.000 from a second Ph.D. dissertation, by Ramalau R. Maila (Maila, 2006). Obviously, this is not a balanced corpus; it was indeed chosen because of its easy accessibility. We use this corpus to train our taggers and to test them; in a ten-fold cross validation, we split the text into ten slices of roughly equal size, train on 9 of them and test on the tenth. In this article, we give figures for the median of these results.

concord of class 15, object concord of the locative classes, object concord of the 2nd person singular, subject concord of class 15, indefinite subject concord, subject concord of the locative classes, class prefix of class 15, locative particle, copulative indicating either an indefinite subject, or a subject of class 15 or a locative subject.

Authors	No. of tags	± noun class	tool?
(van Rooy and Pretorius, 2003)	106	- noun class	no
(De Schryver and De Pauw, 2007)	56	- noun class	yes
(Kotzé, 2008)	partial	N.R.	yes
(Taljard et al., 2008)	141/262	+ noun class	no
This paper	25/141	+ noun class	yes

Table 1: Tagsets for N. Sotho: authors, # of tags, consideration of the noun class system, use in tools

3.3 Tagging techniques: the RF-tagger

We opt for the RF-tagger (Schmid and Laws, 2008), because it is a Hidden-Markov-Model (HMM) tagger which was developed especially for POS tagsets with a large number of (fine-grained) tags. Tests with our training corpus have shown that this tagger outperforms the Tree-tagger ((Schmid, 1994) and (Schmid, 1995)), as shown in figure 1. An additional external lexicon may serve as input, too. The development of the RF-tagger was based on the widely shared opinion that for languages like German or Czech, agreement information (e.g. case, number or person) should preferably appear as part of all appropriate part of speech tags. However, as tagsets increase immensely in size when such information is part of the tags, the data are decomposed, i.e. split into several levels of processing. The probability of each level is then calculated separately (the joint probability of all levels is afterwards calculated as their product). With such methodology, a tag of the German determiner *das* may contain five levels of information, e.g. ART.Def.Nom.Sg.Neut to define a definite, nominative singular, neutral determiner (article) that appears in the nominative case.

This approach makes sense for the Bantu-languages as well, since information on noun class numbers should be part of the noun tags, too, as in Taljard et al's (2008) tagset. A noun here is not only tagged "N", but `Nclass`, e.g. *mohumagadi* '(married) woman' as N01. All concords, pronouns or other types that concordially agree with nouns are also labelled with a class number, e.g. *o*, the subject concord of class 1, is labelled CS01. This approach makes sense, especially in the view of chunking/parsing and reference resolution, because any of those elements can acquire a pronominal function when the noun that they refer to is deleted (Louwrens, 1991).

To utilize the RF-tagger, we split all tags containing noun class numbers into several levels (e.g. the tag N01 becomes N.01). Emphatic and possessive pronouns are represented on three levels (e.g. PROPOSSPERS becomes PRO.POSS.PERS)[2].

4 Results

In a preliminary experiment, we compared several taggers[3] on our manually annotated data. Apart from the RF-tagger (Schmid and Laws, 2008), we also used the Tree-Tagger (Schmid, 1994), the TnT tagger (Brants, 2000) and MBT (Daelemans et al., 2007).

4.1 Comparing taggers

The results give a relatively homogenous picture, with the RF-tagger achieving a median of 94.16 % when utilising a lexicon containing several thousand nouns and verbs. It leads to 91 % accuracy without this lexicon (to simulate similar conditions as for TnT or MBT where no external lexicon was offered). TnT achieves 91.01 %, and MBT 87.68 %. Data from the Tree-Tagger were not comparable for they had been obtained at an earlier stage using the lexicon (92.46 %).

4.2 Effects of the size of the training corpus on the tagging results

All probabilistic taggers are in need of training data the size of which depends on the size of the tagset and on the frequencies of occurrence of each context. De Schryver and de Pauw (2007) demonstrated that when utilizing a tagset that contains only about a third of the tags (56) contained in Taljard et al.'s (2008) tagset (141), their MaxTag POS-tagger reaches a 93.5 % accuracy with a training corpus of only about 10,000 tokens.

Figure 1 depicts the effects of the size of the training corpus on the accuracy figures of the Tree-tagger and the RF-tagger. Tests with training corpora of the sizes 15,000, 30,000 and 45,000 tokens

[2] Tests have shown that the quantitative pronouns should be treated separately, their tags are thus only split into two levels.

[3] Check the References section for the availability of these taggers.

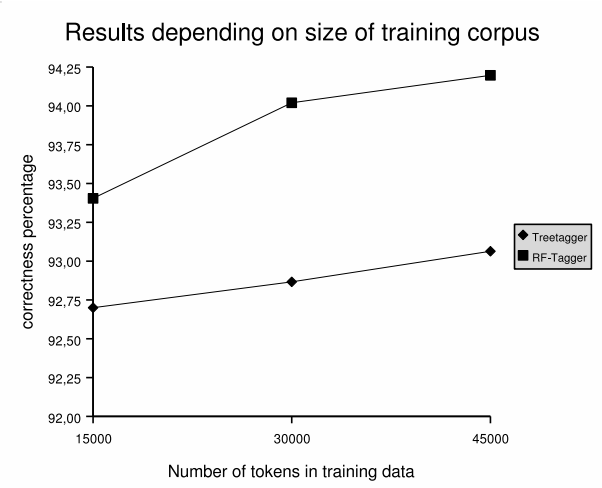


Figure 1: Effects of the size of the training corpus on tagger results.

showed that the results might not improve much if more data is added. The RF-tagger already reaches more than 94 % correctness when utilizing the current 45,000 token training corpus.

4.3 Effects of the highly polysemous function words of Northern Sotho

The less frequently a token-label pair appears in the corpus, the lower is its probability (leading to the sparse data problem, when probability guesses become unreliable because of low numbers of occurrences). This issue poses a problem for Northern Sotho function words: if they occur very frequently with a certain label, the chances of them being detected with another label are fairly low. This effect is demonstrated in table 2, which describes the detection of the parts of speech of the highly ambiguous function word -*a*-. The word -*a*- as PART(icle) occurs only 45 times while -*a*- as CS.01 occurs 1,182 times. More than 50 % of the particle occurrences (23) are wrongly labelled CS.01 by the tagger. In table 2, we list the correct tags of all occurrences of -*a*-, as well as the assigned tags to each of them by our tagger. Each block of table 2 is ordered by decreasing numbers of occurrence of each tag in the output of the RF-tagger. For easier reference, the correct tags assigned by the RF-tagger are printed in bold. Table 2 also clearly shows the effect of ambiguous local context on the tagging result: the accuracy of the CS.06-annotation (subject concord of class 6) is considerably lower than that of the more frequent CS.01 (96.45 % vs. 63.08 %), and CS.01 is the most frequent error in the CS.06 assignment process.

***a* as**	**freq**	**RF-tagger**	**sums**	%
CS.01	1182	**CS.01**	1140	96.4
		CS.06	19	1.6
		MORPH	14	1.2
		CDEM.06	3	0.3
		PART	2	0.2
		CPOSS.06	2	0.2
		CO.06	2	0.2
CS.06	176	**CS.06**	111	63.1
		CS.01	43	24.4
		CPOSS.06	10	5.7
		CDEM.06	5	2.8
		MORPH	3	1.7
		PART	3	1.7
		CO.06	1	0.6
CO.06	18	MORPH	7	38.9
		CS.01	6	33.3
		CO.06	3	16.7
		CS.06	2	11.11
PART	45	CS.01	23	51.1
		MORPH	11	24.4
		CDEM.06	5	11.1
		PART	5	11.1
		CPOSS.06	1	2.2
CDEM.06	97	**CDEM.06**	89	91.8
		CPOSS.06	4	4.1
		CS.06	2	2.1
		CS.01	1	1.0
		PART	1	1.0
CPOSS.06	209	**CPOSS.06**	186	89.0
		CDEM.06	12	5.7
		CS.06	6	2.9
		PART	4	1.9
		CS.01	1	0.5
MORPH	89	**MORPH**	44	49.4
		CO.06	26	29.2
		CS.01	15	16.9
		CPOSS.06	4	4.5
sums	1816		1816	

Table 2: RF-tagger results for -***a***-

4.4 Suggestions for increasing correctness percentages

Spoustová et al. (2007) describe a significant increase of accuracy in statistical tagging when utilizing rule-based macros as a preprocessing, for Czech. We have contemplated, in an earlier stage of our work (Prinsloo and Heid, 2005) to adopt a

similar strategy, i.e. to design rule-based macros for the (partial) disambiguation of high-frequency function words. However, the fact that the local context of many function words is similar (i.e. the ambiguity of this local context (see above)), is a major obstacle to a disambiguation of single function words by means of rules. Rules would interact in many ways, be dependent on the application order, or disambiguate only partially (i.e. leave several tag options). An alternative would be to design rules for the disambiguation of word or morpheme sequences. This would however amount to partial parsing. The status of such rules within a tagging architecture would then be unclear.

4.5 Effects of tagset size and structure

While a preprocessing with rule-based disambiguation does not seem to be promising, there are other methods of improving accuracy, such as, e.g., the adaptation of the tagset. Obviously, types appearing in different contexts should have different labels. For example, in the tagset of Taljard et al. (2008), auxiliary verbs are a sub-class of verbs (V_aux). In typical Northern Sotho contexts, however, auxiliaries are surrounded by subject concords, while verbs are only preceded by them. When 'promoting' the auxiliaries to the first level by labelling them VAUX, the RF-tagger result increases by 0.13 % to 94.16 % accuracy. We still see room for further improvement here. For example, *ga* as PART (either locative particle PART_loc or hortative particle PART_hort) is identified correctly in only 29.2 % of all cases at the moment. The hortative particle usually appears at the beginning of a verbal segment, while the locative in most cases follows the segment. Results may increase to an even higher accuracy when 'promoting' these second level annotations, hort(ative) and loc(ative) to the first annotation level.

5 Conclusions and future work

This article gives an overview of work on POS-tagging for Northern Sotho. Depending on the place of tagging in an overall NLP chain for this language, different choices with respect to the tagset and to the tagging technology may prove adequate.

In our work, which is part of a detailed linguistic annotation of Northern Sotho corpora for linguistic exploration with a view to lexicons and grammars, it is vital to provide a solid basis for chunking and/or parsing, by including information on noun class numbers in the annotation. We found that the RF-tagger (Schmid and Laws, 2008) performs well on this task, partly because it allows us to structure the tagset into layers, and to deal with noun class information in the same way as with agreement features for European languages. We reach over 94 % correctness, which indicates that at least a first attempt at covering the PSC corpus may now be in order.

Our error analysis, however, also highlights a few more general aspects of the POS annotation of Northern Sotho and related languages: obviously, frequent items and items in distinctive local contexts are tagged quite well. When noun class information is part of the distinctions underlying the tagset, function words usable for more than one noun class tend however, to appear in non-distinctive local contexts and thus to lead to a considerable error rate. Furthermore, we found a few cases of uses of, e.g., subject concords that are anaphoric, with antecedents far away and thus not accessible to tagging procedures based on the local context. These facts raise the question whether, to achieve the highest quality of lexical classification of the words and morphemes of a text, chunking/parsing might be required altogether, rather than tagging.

Our experiments also showed that several parameters are involved in fine-tuning a Sotho tagger. The size and structure of the tagset is one such a prominent parameter. Tendencies towards simpler and smaller tagsets obviously conflict with the needs of advanced processing of the texts and of linguistically demanding applications. It seems that tagset design and tool development go hand in hand.

We intend to apply the current version of the RF-tagger to the PSC corpus and to evaluate the results carefully. We expect a substantial gain from the use of the guessers for nouns and verbs, cf. (Prinsloo et. al, 2008) and (Heid et al., 2008). Detailed error analysis should allow us to also design specific rules to correct the output of the tagger. Instead of preprocessing (as proposed by Spoustová et al. (2007)), a partial postprocessing may contribute to further improving the overall quality. Rules would then probably have to be applied to particular sequences of words and/or morphemes which cause difficulties in the statistical process.

References

Adam L. Berger, Stephen Della Peitra, and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics* 22(1): pp. 39 – 71.

Thorsten Brants. 2000. TnT - A Statistical Part-of-Speech Tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000, Seattle, WA*.

Walter Daelemans, Jakub Zavrel, Antal van den Bosch. 2007. *MBT: Memory-Base Tagger, version 3.1*. Reference Guide. ILK Technical Report Series 07–08 [online]. Available : `http://ilk.uvt.nl/mbt` (10th Jan, 2009).

Gilles-Maurice de Schryver and Guy de Pauw. 2007. Dictionary Writing System (DWS) + Corpus Query Package (CQP): The Case of TshwaneLex. *Lexikos 17 AFRILEX-reeks/series* 17:2007: pp. 226 – 246. [Online tagger:] `http://aflat.org/?q=node/177`. (10th Feb, 2009)

Gilles-Maurice de Schryver and Daan J. Prinsloo. 2000. The compilation of electronic corpora with special reference to the African languages. *Southern African Linguistics and Applied Language Studies* 18(1-4): pp. 89 – 106.

Malcolm Guthrie. 1971. *Comparative Bantu: an introduction to the comparative linguistics and prehistory of the Bantu languages, vol 2,* Farnborough: Gregg International.

Ulrich Heid, Daan J. Prinsloo, Gertrud Faaß, and Elsabé Taljard. 2008 *Designing a noun guesser for part of speech tagging in Northern Sotho* (33 pp). ms: University of Pretoria.

Petronella M. Kotzé. 2008. Northern Sotho grammatical descriptions: the design of a tokeniser for the verbal segment. *Southern African Linguistics and Applied Language Studies* 26(2): pp. 197 – 208.

Zhang Le. 2004. *Maximum Entropy Modeling Toolkit for Python and C++* (Technical Report) [online]. Available: `http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html` (10th Jan, 2009).

Geoffrey Leech, Andrew Wilson. 1999. Standards for Tagsets. in van Halteren (Ed.) *Syntactic world-class tagging*: pp. 55 – 80 Dordrecht/Boston/London: Kluwer Academic Publishers.

Louis J. Louwrens. 1991. *Aspects of the Northern Sotho Grammar* p. 154. Pretoria:Via Afrika.

Ramalau A. Maila. 2006. *Kgolo ya tiragatso ya Sepedi.* [="Development of the Sepedi Drama"]. Doctoral thesis. University of Pretoria, South Africa.

Oliver K. Matsepe. 1974. Tša Ka Mafuri. [="From the homestead"]. Pretoria: Van Schaik.

Daan J. Prinsloo. 1994. Lemmatization of verbs in Northern Sotho. *SA Journal of African Languages* 14(2): pp. 93 – 102.

Daan J. Prinsloo and Ulrich Heid. 2005. Creating word class tagged corpora for Northern Sotho by linguistically informed bootstrapping. in: Isabella Ties (Ed.): *LULCL, Lesser used languages and computational linguistics*, 27/28-10-2005, Bozen/Bolzano, (Bozen: Eurac) 2006: pp. 97 – 113.

Daan J. Prinsloo, Gertrud Faaß, Elsabé Taljard, and Ulrich Heid. 2008. Designing a verb guesser for part of speech tagging in Northern Sotho. *Southern African Linguistics and Applied Languages Studies (SALALS) 26(2)*.

Helmut Schmid and Florian Laws. 2008. *Estimation of Conditional Probabilities with Decision Trees and an Application to Fine-Grained POS Tagging* [online]. COLING 2008. Manchester, Great Britain. Available: `http://www.ims.uni-stuttgart.de/projekte/corplex/RFTagger/` (10th Jan, 2009).

Helmut Schmid. September 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of the International Conference on New Methods in Language Processing*[online]. Available: `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/` (10th Jan, 2009).

Helmut Schmid. March 1995. Improvements in Part-of-Speech Tagging with an Application to German. *Proceedings of the ACL SIGDAT-Workshop*.

Drahomíra "johanka" Spoustová, Jan Hajič, Jan Votrubec, Pavel Krbec, and Pavel Květoň. Jun 29, 2007. The best of two worlds: Cooperation of Statistical and Rule-based Taggers for Czech. *Balto-Slavonic Natural Language Processing*: pp. 67 – 74 [online]. Available: `http://langtech.jrc.it/BSNLP2007/m/BSNLP-2007-proceedings.pdf` (10th Jan, 2009).

Elsabé Taljard, Gertrud Faaß, Ulrich Heid and Daan J. Prinsloo. 2008. On the development of a tagset for Northern Sotho with special reference to standardisation. *Literator 29(1) 2008*. Potchefstroom, South Africa.

Raphehli M. Thobakgale. *Khuetšo ya OK Matsepe go bangwadi ba Sepedi* [="Influence of OK Matsepe on the writers of Sepedi"]. Doctoral thesis. University of Pretoria, South Africa.

Bertus van Rooy and Rigardt Pretorius. 2003. A word-class tagset for Setswana. *Southern African Linguistics and Applied Language Studies* 21(4): pp. 203 – 222.

Appendix A. The polysemy of *-a-*

	Description	Example
1	Subject concord of	*ge monna **a** fihla* conjunctive + noun cl. 1 + **subject concord cl. 1** + verb stem if/when + man + **subj-cl1** + arrive ”when the man arrives”
2	Subject concord of nominal cl. 6	*masogana **a** thuša basadi* noun cl. 6 + **subject concord cl. 6** + verb stem + noun cl.2 young men + **subj-cl6** + help women ”the young men help the women”
3	Possessive concord of nominal cl. 6	*maoto **a** gagwe* noun cl. 6 + **possessive concord cl. 6** + possessive pronoun cl. 1 feet + **of** + his ”his feet”
4	Present tense morpheme	*morutiši o **a** bitša* noun cl. 1 + subject concord cl.1 + **present tense marker** + verb stem teacher + subj-cl1 + **pres** + call ”the teacher is calling”
5	Past tense morpheme	*morutiši ga o **a** bitša masogana* noun cl. 1 + negation morpheme + subject concord cl.1 + **past tense marker** + verb stem + noun cl. 6 teacher + neg + subj-cl1 + **past** + call + young men ”the teacher did not call the young men”
6	Demonstrative concord of nominal cl. 6	*ba nyaka masogana **a*** subject concord cl. 2 + verb stem + noun cl. 6 + **demonstrative concord** they + look for + young men + **these** ”they are looking for these young men”
7	Hortative particle	***a** ba tsene* **hortative particle** + subject concord cl. 2 + verb stem let + subj-cl2 + come in ”let them come in”
8	Interrogative particle	***a** o tseba Sepedi* **interrogative particle** + subject concord 2nd pers sg. + verb stem + noun cl. 7 **ques** + subj-2nd-pers-sg + know + Sepedi ”do you know Sepedi”
9	Object concord of	*moruti o **a** biditše* noun cl. 1 + subject concord cl. 1 + **object concord cl. 6** + verb stem teacher + subj-cl1 + **obj-cl6** + called ”the teacher called them”

Development of an Amharic Text-to-Speech System Using Cepstral Method

Tadesse Anberbir
ICT Development Office, Addis Ababa University, Ethiopia
tadanberbir@gmail.com

Tomio Takara
Faculty of Engineering, University of the Ryukyus, Okinawa, Japan
takara@ie.u-ryukyu.ac.jp

Abstract

This paper presents a speech synthesis system for Amharic language and describes and how the important prosodic features of the language were modeled in the system. The developed Amharic Text-to-Speech system (AmhTTS) is parametric and rule-based that employs a cepstral method. The system uses a source filter model for speech production and a Log Magnitude Approximation (LMA) filter as the vocal tract filter. The intelligibility and naturalness of the system was evaluated by word and sentence listening tests respectively and we achieved 98% correct-rates for words and an average Mean Opinion Score (MOS) of 3.2 (which is categorized as good) for sentences listening tests. The synthesized speech has high intelligibility and moderate naturalness. Comparing with previous similar study, our system produced considerably similar quality speech with a fairly good prosody. In particular our system is mainly suitable for building new languages with little modification.

1 Introduction

Text-to-Speech (TTS) synthesis is a process which artificially produces synthetic speech for various applications such as services over telephone, e-document reading, and speaking system for handicapped people etc.

The two primary technologies for generating synthetic speech are concatenative synthesis and formant (Rule-based) synthesis methods. Concatenative synthesis produces the most natural-sounding synthesized speech. However, it requires a large amount of linguistic resources and generating a various speaking style is a challenging task. In general the amount of work required to build a concatenative system is enormous. Particularly, for languages with limited linguistic resources, it is more difficult. On the other hand, formant synthesis method requires small linguistic resources and able to generate various speaking styles. It is also suitable for mobile applications and easier for customization. However, this method produced less natural-sounding synthesized speech and the complex rules required to model the prosody is a big problem.

In general, each method has its own strengths and weaknesses and there is always a tradeoff. Therefore, which approach to use will be determined by the intended applications, the availability of linguistic resources of a given language etc. In our research we used formant (rule-based) synthesis method because we are intending to prepare a general framework for Ethiopian Semitic languages and apply it for mobile devices and web embedded applications.

Currently, many speech synthesis systems are available mainly for 'major' languages such as English, Japanese etc. and successful results are obtained in various application areas. However, thousands of the world's 'minor' languages lack such technologies, and researches in the area are quite very few. Although recently many localization projects (like the customization of Festvox[1]) are being undergoing for many languages, it is quite inadequate and the localization process is not an easy task mainly because of the lack of linguistic resources and absence of similar works in the area. Therefore, there is a strong demand for the development of a speech synthesizer for many of the African minor languages such as Amharic.

Amharic, the official language of Ethiopia, is a Semitic language that has the greatest number of speakers after Arabic. According to the 1998 census, Amharic has 17.4 million speaker as a mother thong language and 5.1 million speakers as a second language. However, it is one of the

[1] Festvox is a voice building framework which offers general tools for building unit selection voices for new languages.

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 46–52,
Athens, Greece, 31 March 2009.

least supported and least researched languages in the world. Although, recently, the development of different natural language processing (NLP) tools for analyzing Amharic text has begun, it is often very far comparing with other languages (Alemu et al., 2003). Particularly, researches conducted on language technologies like speech synthesis and the application of such technologies are very limited or unavailable. To the knowledge of the authors, so far there is only one published work (Sebsibe, 2004) in the area of speech synthesis for Amharic. In this study they tried to describe the issues to be considered in developing a concatenative speech synthesizer using Festvox and recommended using syllables as a basic unit for high quality speech synthesis.

In our research we developed a syllabic based TTS system with prosodic control method which is the first rule-based system published for Amharic. The designed Amharic TTS (AmhTTS) is parametric and rule-based system that employs a Cepstral method and uses a Log Magnitude Approximation (LMA) filter. Unlike the previous study, Sebsibe (2004), our study provides a total solution on prosodic information generation mainly by modeling the durations. The system is expected to have a wide range of applications, for example, in software aids to visually impaired people, in mobile phones and can also be easily customized for other Ethiopian languages.

2 Amharic Language's Overview

Amharic (**አማርኛ**) is a Semitic language and it is one of the most widely spoken languages in Ethiopia. It has its own non Latin based syllabic script called "Fidel" or "Abugida". The orthographic representation of the language is organized into orders (derivatives) as shown in Fig.1. Six of them are CV (C is a consonant, V is a vowel) combinations while the sixth order is the consonant itself. In total there are 32 consonants and 7 vowels with 7x32= 224 Syllables. But since there are redundant sounds that represent the same sounds, the phonemes are only 28 (see the Appendix).

1st	2nd	3rd	4th	5th	6th	7th
C/e/ ቸ	C/u/ ቹ	C/i/ ቺ	C/a/ ቻ	C/ie/ ቼ	C ች	C/o/ ቾ

Figure 1: Amharic Syllables structure

Like other languages, Amharic also has its own typical phonological and morphological features that characterize it. The following are some of the striking features of Amharic phonology that gives the language its characteristic sound when one hears it spoken: the weak, indeterminate stress; the presence of glottalic, palatal, and labialized consonants; the frequent gemination of consonants and central vowels; and the use of an automatic helping vowel (Bender et al., 1976).

Gemination in Amharic is one of the most distinctive characteristics of the cadence of the speech, and also caries a very heavy semantic and syntactic functional weight (Bender and Fulass, 1978). Amharic gemination is either lexical or morphological. Gemination as a lexical feature cannot be predicted. For instance, **አለ** may be read as *alä* meaning 'he said', or *allä* meaning 'there is'. Although this is not a problem for Amharic speakers, it is a challenging problem in speech synthesis. As a morphological feature gemination is more predictable in the verb than in the noun, Bender and Fulass (1978). However, this is also a challenging problem in speech synthesis because to automatically identify the location of geminated syllables, it requires analysis and modeling of the complex morphology of the language. The lack of the orthography of Amharic to show geminates is the main problem. In this study, we used our own manual gemination mark (') insertion techniques (see Section 3.3.1).

The sixth order syllables are the other important features of the language. Like geminates, the sixth order syllables are also very frequent and play a key role for proper pronunciation of speech. In our previous study, (Tadesse and Takara, 2006) we found that geminates and sixth order syllables are the two most important features that play a key role for proper pronunciation of words. Therefore, in our study we mainly consider these language specific features to develop a high quality speech synthesizer for Amharic language.

3 AmhTTS System

Amharic TTS synthesis system is a parametric and rule based system designed based on the general speech synthesis system. Fig.2. shows the scheme of Amharic speech synthesis system. The design is based on the general speech synthesis system (Takara and Kochi, 2000). The system has three main components, a text analysis subsystem, prosodic generation module and a speech synthesis subsystem. The following three sub-sections discuss the details of each component.

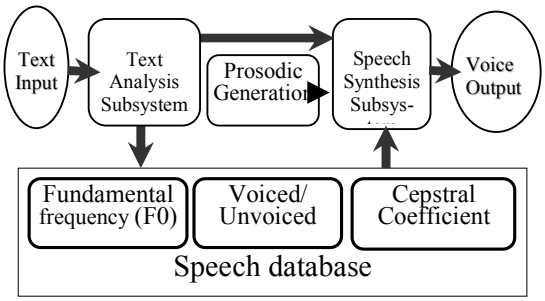


Figure 2: Amharic Speech Synthesis System

3.1 Text Analysis

The text analysis subsystem extracts the linguistic and prosodic information from the input text. The program iterates through the input text and extracts the gemination and other marks, and the sequences of syllables using the syllabification rule. The letter-to-sound conversion has simple one-to-one mapping between orthography and phonetic transcription (see Apendix). As defined by (Baye, 2008; Dawkins, 1969) and others, Amharic can be considered as a phonetic language with relatively simple relationship between orthography and phonology.

3.2 Speech Analysis and Synthesis systems

First, as a speech database, all Amharic syllables (196) were collected and their sounds were prepared by recording on digital audio tape (DAT) at a 48 kHz sampling rate and 16-bit value. After that, they were down-sampled to 10 kHz for analyzing. All speech units were recorded with normal speaking rate.

Then, the speech sounds were analyzed by the analysis system. The analysis system adopts short-time cepstral analysis with frame length 25.6 ms and frame shifting time of 10 ms. A time-domain Hamming window with a length of 25.6 ms is used in analysis. The cepstrum is defined as the inverse Fourier transform of the short-time logarithm amplitude spectrum (Furui, 2001). Cepstral analysis has the advantage that it could separate the spectral envelope part and the excitation part. The resulting parameters of speech unit include the number of frames and, for each frame, voiced/unvoiced (V/UV) decision, pitch period and cepstral coefficients c[m], $0 \leq m \leq 29$. The speech database contains these parameters as shown in fig.2.

Finally, the speech synthesis subsystem generates speech from pre-stored parameters under the control of the prosodic rules. For speech synthesis, the general source-filter model is used as a speech production model as shown in fig.3. The synthetic sound is produced using Log Magnitude Approximation (LMA) filter (Imai, 1980) as the system filter, for which cepstral coefficients are used to characterize the speech sound.

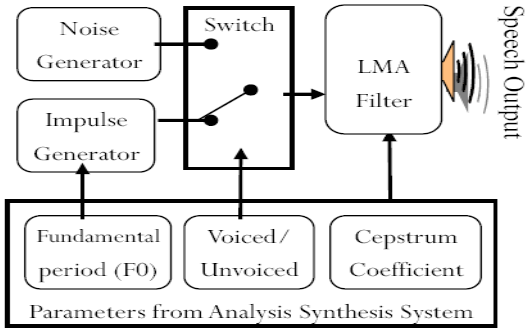


Figure 3: Diagram of Speech Synthesis Model

The LMA filter presents the vocal tract characteristics that are estimated in 30 lower-order quefrency elements. The LMA filter is a pole-zero filter that is able to efficiently represent the vocal tract features for all speech sounds. The LMA filter is controlled by cepstrum parameters as vocal tract parameters, and it is driven by fundamental period impulse series for voiced sounds and by white noise for unvoiced sounds. The fundamental frequency (F0) of the speech is controlled by the impulse series of the fundamental period. The gain of the filter or the power of synthesized speech is set by the 0th order cepstral coefficient, c [0].

3.3 Prosody Modeling

For any language, appropriate modeling of prosody is the most important issue for developing a high quality speech synthesizer.

In Amharic language segments duration is the most important and useful component in prosody control. It is shown that, unlike English language in which the rhythm of the speech is mainly characterized by stress (loudness), rhythm in Amharic is mainly marked by longer and shorter syllables depending on gemination of consonants, and by certain features of phrasing (Bender et al., 1976). Therefore it is very important to model the syllables duration in AmhTTS system. In this paper we propose a new segmental duration control methods for synthesizing a high quality speech. Our rule-based TTS system uses a compact rule-based prosodic generation method in three phases:

- modeling geminated consonants duration

- controlling of sixth order syllables duration
- assignment of a global intonation contour

Prosody is modeled by variations of pitch and relative duration of speech elements. Our study deals only with the basic aspects of prosody such as syllables duration and phrase intonation. Gemination is modeled as lengthened duration in such a way geminated syllables are modeled on word level. Phrase level duration is modeled as well to improve prosodic quality. Prosodic phrases are determined in simplified way by using text punctuation. To synthesize F0 contour Fujisaki pitch model that superimpose both word level and phrase level prosody modulations is used (Takara and Jun, 1988).

The following sub-sections discuss the prosodic control methods employed in our system.

3.3.1 Gemination rule

Accurate estimation of segmental duration for different groups of geminate consonants (stops, nasals, liquids, glides, fricatives) will be crucial for natural sounding of AmhTTS system. In our previous study, Tadesse and Takara (2006), we studied the durational difference between singletons vs. geminates of contrastive words and determined the threshold duration for different groups of consonants. Accordingly the following rule was implemented based on the threshold durations we obtained in our previous study.

The gemination rule is programmed in the system and generates geminates from singletons by using a simple durational control method. It generates geminates by lengthening the duration of the consonant part of the syllables following the gemination mark. Two types of rules were prepared for two groups of consonants, continuant (voiced and unvoiced) and non-continuant (stops and glottalized) consonants. If a gemination mark (‘) is followed by syllable with voiced or unvoiced consonant then, the last three frames of the cepstral parameters (c[0]) of vowel is adjusted linearly and then 120 ms of frame 1, 2 and 3 of second syllable is added. Then the second syllable is connected after frame 4. Totally 90 ms of cepstral parameters is added. Otherwise, if, a gemination mark (‘) is followed by syllable with glottal or non-glottal consonant then, the last three frames of the cepstral parameters (c[0]) of vowel is adjusted linearly and then 100 ms of silence is added. Finally the second syllable is directly connected.

Since Amharic orthography does not use gemination mark, in our study we used our own gemination mark and a manual gemination insertion mechanism for input texts. Although some scholars make use of two dots (¨ which is proposed in UNICODE 5.1 version as 135F) over a consonant to show gemination, so far there is no software which supports this mark.

3.3.2 Sixth order syllables rule

As mentioned earlier the sixth order syllables are very frequent and play a major role for proper pronunciation of words. The sixth order orthographic syllables, which do not have any vowel unit associated to it in the written form, may associate the helping vowel (epenthetic vowel /ix/, see the Apendix) in its spoken form (Dawkins, 1969). The sixth order syllables are ambiguous; they can stand for either a consonant in isolation or a consonant with the short vowel. In our study we prepared a simple algorithm to control the sixth order syllables duration. The algorithm to model the sixth order syllables duration uses the following rules:

1. The sixth order syllables at the beginning of word are always voweled (see /sxix/ in fig 5).
2. The sixth order syllables at the end of a word are unvoweled (without vowel) but, if it is geminated, it becomes voweled.
3. The sixth order syllables in the middle of words are always unvoweled (see /f/ in fig.5). But, if there is a cluster of three or more consonants, it is broken up by inserting helping vowel /ix/.

The following figures shows sample words synthesized by our system by applying the prosodic rules. Fig.5 and fig.7 shows the waveform and duration of words synthesized by applying both the gemination and sixth order syllables rules. Fig.4 and fig.6 shows the waveform of original words just for comparison purpose only. The two synthesized words are comparative words which differ only by presence or absence of gemination. In the first word /sxixfta/ ሽፍታ meaning ‘bandit’, the sixth order syllable /f/ is unvoweled (see fig.5). However, in the second word /sxixffixta/ ሽፍታ meaning ‘rash’, the sixth order syllable /f/ is voweled and longer /ffix/ (see fig.7) because it is geminated. In our previous study, Tadesse and Takara (2006), we observed that vowels are needed for singletons to be pronounced as geminates.

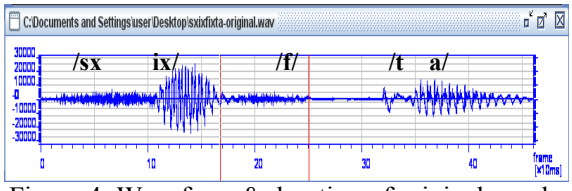


Figure 4: Waveform & duration of original word ሽፍታ/sxixfta/, meaning 'bandit'

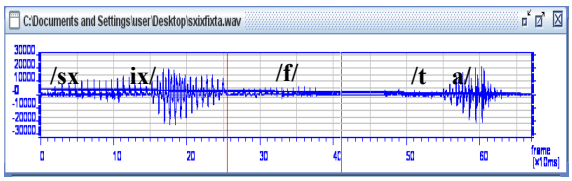


Figure 5: Waveform & duration of synthesized word ሽፍታ/sxixfta/, meaning 'bandit'

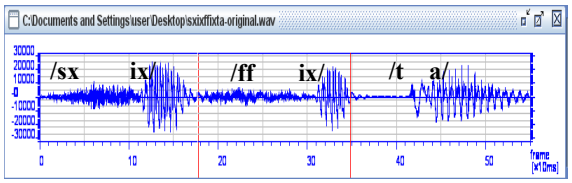


Figure 6: Waveform & duration of original word ሽፍታ/sxixffixta/, meaning 'rash'

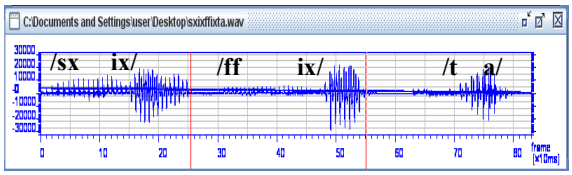


Figure 7: Waveform & duration of synthesized word ሽፍታ/sxixffixta/, meaning 'rash'

3.3.3 Syllables connection rules

For syllables connections, we prepared four types of syllables connection-rules based on the type of consonants. The syllable units which are represented by cepstrum parameters and stored in the database are connected based on the types of consonants joining with vowels. The connections are implemented either by smoothing or interpolating the cepstral coefficients, F0 and amplitude at the boundary. Generally, we drive two types of syllabic joining patterns. The first pattern is smoothly continuous linkage where the pitch, amplitude and spectral assimilation occur at the boundary. This pattern occurs when the boundary phonemes of joining syllables are unvoiced. Another joining pattern is interpolation, this pattern occurs when one or both of the boundary phonemes of joining syllables is voiced. If the boundary phonemes are plosive or glottal stop then the pre-plosive or glottal stop closure pauses with 40ms in length is inserted between them.

3.3.4 Intonation

The intonation for a sentence is implemented by applying a simple declination line in the log frequency domain adopted from similar study for Japanese TTS system by Takara and Jun (1988). Fig.8 shows the intonation rule. The time ti is the initial point of syllable, and the initial value of F0 (circle mark) is calculated from this value. This is a simple linear line, which intends to experiment the very first step rule of intonation of Amharic. In this study, we simply analyzed some sample sentences and take the average slope = -0.0011. But as a future work, the sentence prosody should be studied more.

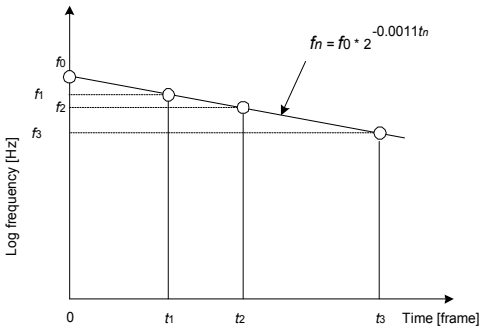


Figure 8: Intonation rule

4 Evaluation and Discussion

In order to evaluate the intelligibility and naturalness of our system, we performed two types of listening tests. The first listening test was performed to evaluate the intelligibility of words synthesized by the system and the second listening test was to evaluate the naturalness of synthesized sentences. The listening tests were used to evaluate the effectiveness of the prosodic rule employed in our system.

4.1 Recordings

For both listening tests the recording was done in a soundproof room, with a digital audio tape recorder (SONY DAT) and SONY ECM-44S Electrets Condenser Microphone. Sampling rate of DAT was set at 48kHz then from DAT the recorded data were transferred to a PC via a digital audio interface (A/D, D/A) converter. Finally, the data was converted from stereo to mono; down sampled to 10 kHz and the amplitude was normalized using Cool Edit program. All recording was done by male native speaker of the language who is not included in the listening tests.

4.2 Speech Materials

The stimuli for the first listening test were consisted of 200 words which were selected from Amharic-English dictionary. The selected words are commonly and frequently used words in the day-to-day activities adopted from (Yasumoto and Honda, 1978). Among the 200 words we selected, 80 words (40% of words) contain one or more geminated syllables and 75% of the words contain sixth order syllables. This shows that how geminates and sixth order syllables are very frequent.

Then, using these words, two types of synthesized speech data were prepared: Analysis/synthesis sounds and rule-based synthesized sounds using AmhTTS system. The original speech sounds were also added in the test for comparison purpose.

For the second listening test we used five sentences which contains words with either geminated syllables or sixth order syllables or both. The sentences were selected from Amharic grammar book, Baye (2008) which are used as an example. We prepared three kinds of speech data: original sentences, analysis/synthesis sentences, and synthesized sentences by our system by applying prosodic rules. In total we prepared 15 speech sounds.

4.3 Methods

Both listening tests were conducted by four Ethiopian adults who are native speakers of the language (2 female and 2 male). All listeners are 20-35 years old in age, born and raised in the capital city of Ethiopia. For both listening tests we prepared listening test programs and a brief introduction was given before the listening test.

In the first listening test, each sound was played once in 4 second interval and the listeners write the corresponding Amharic scripts to the word they heard on the given answer sheet.

In the second listening test, for each listener, we played all 15 sentences together and randomly. And each subject listens to 15 sentences and gives their judgment score using the listening test program by giving a measure of quality as follows: (5 – Excellent, 4 - Good, 3 - Fair, 2 - Poor, 1 – Bad). They evaluated the system by considering the naturalness aspect. Each listener did the listening test fifteen times and we took the last ten results considering the first five tests as training.

4.4 Results and discussion

After collecting all listeners' response, we calculated the average values and we found the following results.

In the first listening test, the average correct-rate for original and analysis-synthesis sounds were 100% and that of rule-based synthesized sounds was 98%. We found the synthesized words to be very intelligible.

In the second listening test the average Mean opinion score (MOS) for synthesized sentences were 3.2 and that of original and analysis/synthesis sentences were 5.0 and 4.7 respectively. The result showed that the prosodic control method employed in our system is effective and produced fairly good prosody. However, the durational modeling only may not be enough to properly generate natural sound. Appropriate syllable connections rules and proper intonation modeling are also important. Therefore studying typical intonation contour by modeling word level prosody and improving syllables connection rules by using quality speech units is necessary for synthesizing high quality speech.

5 Conclusions and future works

We have presented the development of a syllabic based AmhTTS system capable of synthesizing intelligible speech with fairly good prosody. We have shown that syllables produce reasonably natural quality speech and durational modeling is very crucial for naturalness. However the system still lacks naturalness and needs automatic gemination assignment mechanisms for better durational modeling.

Therefore, as a future work, we will mainly focus on improving the naturalness of the synthesizer. We are planning to improve the duration model using the data obtained from the annotated speech corpus, properly model the co-articulation effect of geminates and to study the typical intonation contour. We are also planning to integrate a morphological analyzer for automatic gemination assignment and sophisticated generation of prosodic parameters.

References

Atelach Alemu, Lars Asker and Mesfin Getachew. 2003. *Natural Language Processing For Amharic: Overview And Suggestions for a Way Forward*, Proc. 10th Conference 'Traitement Automatique Des Languages Naturalles', pp. 173-182, Vol.2, Batz-Sur-Mer, France.

Sebsibe H/Mariam, S P Kishore, Alan W Black, Rohit Kumar, and Rajeev Sangal. 2004. *Unit Selection Voice for Amharic Using Festvox*, 5th ISCA Speech Synthesis Workshop, Pittsburgh.

M.L Bender, J.D.Bowen, R.L. Cooper and C.A. Ferguson. 1976. *Language in Ethiopia*, London, Oxford University Press.

M. Lionel Bender, Hailu Fulass. 1978. *Amharic Verb Morphology: A Generative Approach*, Carbondale.

Tadesse Anberbir and Tomio Takara. 2006. *Amharic Speech Synthesis Using Cepstral Method with Stress Generation Rule*, INTERSPEECH 2006 ICSLP, Pittsburgh, Pennsylvania, pp. 1340-1343.

T. Takara and T. Kochi. 2000. *General speech synthesis system for Japanese Ryukyu dialect*, Proc. of the 7th WestPRAC, pp. 173-176.

Baye Yimam. 2008. **የአማርኛ ሰዋሰው** (“*Amharic Grammer”)*, Addis Ababa. (in Amaharic).

C.H DAWKINS. 1969. *The Fundamentals of Amharic*, Bible Based Books, SIM Publishing, Addis Ababa, Ethiopia, pp.5-7.

S. Furui, *Digital Speech Processing, Synthesis, and Recognitio*n, Second Edition, Marcel Dekker, Inc., 2001, pp. 266-270.

S. Imai. 1980. Log *Magnitude Approximation (LMA) filter*, Trans. of IECE Japan, J63-A, 12, PP. 886-893. (in Japanese).

Takara, Tomio and Oshiro, Jun. 1988. *Continuous Speech Synthesis by Rule of Ryukyu Dialect*, Trans. IEEE of Japan, Vol. 108-C, No. 10, pp. 773-780. (in Japanese)

B. Yasumoto and M. Honda. 1978. *Birth Of Japanses*, pp.352-358, Taishukun-Shoten. (in Japanese).

Appendix

Amharic Phonetic List, IPA Equivalence and its ASCII Transliteration Table. (Mainly adopted from (Sebsibe, 2004; Baye, 2008)

IPA Equivalent	Transcription	Amharic Scripts
Consonants		
[p]	[p]	ፐ
[t]	[t]	ተ
[k]	[k]	ከ
[ʔ]	[ax]	ዐ
[b]	[b]	በ
[d]	[d]	ድ
[g]	[g]	ግ
[p’]	[px]	ጵ
[t’]	[tx]	ጥ
[c’]	[cx]	ፂ
[q]	[q]	ቅ
[f]	[f]	ፍ
[s]	[s]	ስ
[ʃ]	[sh]	ሽ
[h]	[h]	ሀ
[s’]	[sx]	ጽ
[tʃ]	[c]	ች
[g’]	[j]	ጅ
[m]	[m]	ም
[n]	[n]	ን
[n’]	[nx]	ኝ
[l]	[l]	ል
[r]	[r]	ር
[j]	[y]	ይ
[w]	[w]	ው
[v]	[v]	ቭ
[z]	[z]	ዝ
[z’]	[zx]	ዠ
Vowels		
[ə]	[e]	ኧ
[ʊ]	[u]	ኡ
[ɪ]	[ii]	ኢ
[a]	[a]	አ
[e]	[ie]	ኤ
[i]	[ix]	እ
[o]	[o]	ኦ

Building Capacities in Human Language Technology for African Languages

Tunde Adegbola
African Languages Technlogy Initiative (Alt-i), Ibadan, Nigeria
taintransit@hotmail.com

Abstract

The development of Human Language Technology (HLT) is one of the important by-products of the information revolution. However, the level of knowledge and skills in HLT for African languages remain unfortunately low as most scholars continue to work within the frameworks of knowledge production for an industrial society while the information age dawns. This paper reports the work of African Languages Technology Initiative (Alt-i) over a five-year period, and thereby presents a proposal for the acceleration of the development of knowledge and skills in HLT for African languages.

1 Introduction

The world is undergoing a transformation from industrial economies to an information economy, in which the indices of value are shifting from material to non-material resources. This transformation has been rightly described as a revolution because of the height of its pace and intensity. Of necessity therefore, the response to the changes brought about by the information revolution has to be commensurate, both in pace and intensity.

At the root of the information revolution is the development of digital technology which has brought about a major shift in the way we conceptualize, describe and anticipate our world.

One of the salient social imperatives of the information revolution is the need for humans to communicate through and with machines. This has brought about the need to make machines capable of handling natural language used by humans as against formal language used by machines. The field of Human Language Technology (HLT) was developed to provide the necessary knowledge and skills that will enhance the effectiveness and efficiency with which machines mediate communication between humans as well as facilitate communication between humans and machines.

So far, developments in HLT have not sufficiently addressed African languages. This is attributable to a low level of awareness of the importance of and lack of interest in HLT and among scholars of African languages and scholars of technology on the African continent. Furthermore, there is little or no immediate economic incentive in working in HLT. Consequently, there is a dearth of scholars with the requisite impetus, knowledge and skills to support the development of HLT for African languages. Hence, even though there are pockets of activities in Africa, the level of development of HLT for African languages remains low.

The circumstances that necessitated the development of HLT have been described, and rightly so as a revolution. The response therefore has to be commensurate both in pace and intensity. Hence, the need for accelerated development of HLT for African languages is urgent.

This paper presents a proposal aimed at accelerating the development of HLT for African languages based on the experiences gathered at African Languages Technology Initiative (Alt-i) over a five-year period of activities mainly in Nigeria.

2 The State of Language technology in Africa

The application of Language technology to African languages is relatively new and most efforts seem to be incidental. The most consistent efforts motivated and guided by national policy come from South Africa while projects in other countries are based primarily on private initiatives. In a report on HLT development in Sub-Saharan Africa, Justus Roux (2008) reported nine organizations involved in HLT activities in South Africa, one organization in West Africa and two in East Africa. Seven out of the nine organizations in South Africa are based in universities, one is a Semi-Government institution and one is an agency of the Government.

The seven universities with HLT projects are:

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 53–58,
Athens, Greece, 31 March 2009.

- University of Cape Town
- University of Limpopo
- University of the North West (Potchefstroom)
- University of Pretoria
- University of South Africa
- University of Stellenbosch
- University of the Witwatersrand (Johannesburg)

The Meraka institute is semi-governmental while there is a Human Language Technology Unit under the Department of Arts and Culture of the Government of South Africa.

In West Africa the only organization reported is Africa Languages Technology Initiative (Alt-i), while the two reported in East Africa are The Djibouti Center for Speech Research and Technobyte Speech Technologies in Kenya.

Apart from the organisations reported above, there are individual efforts in some universities. These include Dr. Odetunji Odejobi working on Text to Speech Synthesis at Obafemi Awolowo University, Ile Ife, Nigeria, Dr. Wanjiku Ng'ang'a working on Machine Translation and Dr. Peter Wagacha working on Machine Learning, both at the at the University of Nairobi, Kenya.

Apart from these organizations and individuals that are strictly located in Africa, there are a number other efforts in various parts of the world that address HLT for Africa languages, usually in cooperation with some organizations in Africa. Examples include:

- Local Language Speech Technology Initiative (LLSTI), a project of Outside Echo in the UK
- West African Language Documentation, a project of the University of Bielefeld, Germany in collaboration with the University of Uyo, Nigeria and the University of Cocody, Cote D'Ivoire.

Also, there are significant short-term activities on language technology for African languages both within and outside Africa which have not been sufficiently publicized. For example, in 2002, there was an undergraduate project in Yoruba-English machine translation at the St Mary's College of Maryland, USA.[1]

[1] It is also necessary to mention the efforts of Bisharat and SIL. Even though both organizations are not strictly founded for developing language technology for African languages, they have both done important work in making various resources and tools for developing language technology for African languages available.

3 On-going Alt-i Activities

Since its inception, Alt-i has done more work in Yoruba than in other African languages. This is due primarily to the ready availability of intellectual and other resources for Yoruba at Alt-i's base in Ibadan. However, work in a few other languages with available local resources have also been undertaken. The main projects undertaken so far are:

3.1 Automatic Speech Recognition (ASR) of Yoruba

The ASR project started in 2001, and is still ongoing. A PhD thesis with the title of Application of Tonemic Information for Search-Space Reduction in the Automatic Speech Recognition of Yoruba is one of the results of the ASR project. The project approaches ASR of Yoruba from the point of view that Yoruba tones carry so much information that "talking drums" can "speak" the Yoruba language, hence, ASR of Yoruba (and probably other African tone languages) should be based primarily on tones or at least should address considerable computational resources towards correct identification of tones. Experiments have shown that a tone-guided search of the recognition space as proposed in the above PhD thesis leads to improvement in recognition speed and accuracy. ASR efforts are continuing within the project "Redefining Literacy"; Alt-i's main on-going project funded by the Open Society Initiative for West Africa (OSIWA).

3.2 Text to Speech (TTS) synthesis of Yoruba

This project was conceived and initiated in 2002. TTS is a major component of the "Redefining Literacy" project but we have not yet succeeded in attracting funding for this component of the project. Hence, it is in abeyance. However, one of our Associates, Dr. Odetunji Odejobi of the Department of Computer Science and Engineering of the Obafemi Awolowo University, Ile Ife is actively working on TTS and we are collaboration with him on this project. Dr. Odejobi applies fuzzy logic to formalise Yoruba prosody.

3.3 Machine Translation

Work is on-going on Igbo-English and Yoruba-English Machine Translation. The machine translation projects are not funded at present, but

they are taking advantage of the efforts of student volunteers from the Department of Linguistics and African Languages as well as the Africa Regional Center for information Science, both at the University of Ibadan. The main thrust of the present stage of the project is identifying and developing formal specifications for various challenges of a rule-based machine translation as it relates to translation between Igbo/Yoruba and English.

3.4 Yoruba Spelling checker

As a member of the African Network of Localizers (AnLoc), Alt-i is developing a spelling checker for Yoruba in Open Office. This has provided the opportunity to undertake a computational study of Yoruba morphology. Staff and students of the Department of Linguistics and African Languages at the University of Ibadan are playing an active role in this project. The work is producing new insights for interpreting the existing literature of Yoruba morphology and is already leading to interests in similar projects for other languages. As at the time of writing, a dictionary file of about 5000 Yoruba root words and over 100 highly productive affix rules have been developed. Even though some important Yoruba morphological rules cannot be efficiently coded in Hunspell (the software on which the spelling checker is based) the modest dictionary and affix files have produced a useful spelling checker. The project is funded by the International Development Research Center (IDRC) of Canada.

3.5 Automatic diacritic application for Yoruba

As a by-product of the Yoruba spelling checker project, Alt-i is developing an automatic diacritic application program, using the Bayesian learning approach. This project is not funded, but it is taking due advantage of some of the resources, particularly the corpus produced in the IDRC funded spelling checker project. Work is ongoing to expand the corpus used for the automatic diacritic application program.

3.6 Localization of Microsoft Vista and Office Suite

Alt-i was appointed by Microsoft as moderators for the localization of Microsoft Vista and Office Suite into Hausa, Igbo and Yoruba. This project is making steady progress.

3.7 Academic assistance to the University of Ibadan

Apart from the above projects, Alt-i offers academic support to the University of Ibadan. The Executive Director of Alt-i teaches post-graduate courses in Artificial Intelligence and Information Networking as well as supervises post-graduate projects at the Africa Regional Center for Information Science (ARCIS) in the University of Ibadan. He also gives various levels of support in the supervision of post-graduate projects, particularly in the area of acoustic analysis of speech at the Department of Linguistics and African Languages.

Some of the HLT issues addressed in the Master in Information Science projects between 2002 and the present include:
Statistical Language Model (SLM) of Yoruba, Man-Machine Communication in Yoruba, Machine Translation between spoken and signed Yoruba for the deaf and impaired in hearing, Yoruba phonology multimodal learning courseware and phonetically motivated automatic language identification.

Many PhD students of the Department of Linguistics and African Languages have also enjoyed intellectual support and use of Alt-i's speech laboratory in their studies

3.8 Support to other universities and scholarly associations

Many staff members and students from far and wide travel to Ibadan to use Alt-i's speech laboratory. PhD students as well as faculty members from the University of Lagos, University of Ilorin, University of Benin and University of Abuja come regularly to use the facilities. At present a teaching staff of the Department of Systems Engineering of the University of Lagos is undertaking a PhD programme on ASR of Yoruba. This PhD candidate visits Ibadan frequently and regularly to use Alt-i's library and speech laboratory, as well as consult Alt-i staff.

Alt-i is involved in the activities of various scholarly associations such as the West African Linguistic Society (WALS), Linguistics Association of Nigeria (LAN) and the Yoruba Studies Association of Nigeria (YSAN). In 2004, Alt-i collaborated with the West African Linguistics Society to organize the West African Languages Congress with the theme: Globalisation and the Future of African Languages. Alt-i's collaboration in the organization of this congress influenced the proceedings towards language technol-

ogy which brought about great awareness of language technology issues among West African linguists.

3.9 Bridge building seminar series

Alt-i runs a Bridge Building Seminar series as a way of encouraging cross-disciplinary studies in universities and research centers. These seminars have so far been run in eight Nigerian universities and the National Institute for Nigerian Languages (NINLAN). The one-day seminars bring together scholars from Linguistics, Literature, Psychology, Mathematics, Physics, Computer Science and other relevant departments to build awareness of language technology problems and the need for knowledge and skills from a wide range of departments for their solutions

4 Observations

While undertaking the above activities in the last five years between 2003 and 2008, the following observation were made:

- The intellectual resources needed for developing knowledge, skills and academic programmes in Language Technology are largely available in Nigerian universities.
- The lack of awareness of the need to address language technology problems has made it difficult to harness and direct these resources towards the development of language technology for African languages.
- Strong sentimental attachments to departmental traditions makes it extremely difficult for scholars to venture far outside their departmental cocoons.
- The importance of linguistics as a field of study and the role of linguists in society are not properly understood. Hence students of linguistic may not be sufficiently motivated to aspire to their important roles in society.
- Inappropriate admission criteria, limited curricular, and low level of formal interaction between different faculties in the universities make it extremely difficult for students of the pure sciences, technology and linguistics to share courses and thereby have opportunities for academic interaction beneficial to the development of HLT.

5 Recommendations

- Intensive and sustained awareness building programmes on the importance of linguistics and language technology should be undertaken in institutions of higher learning. This will make it possible to harness some of the available intellectual resources that remain yet untapped for the development of language technology for African languages in these institutions.
- Admission criteria and curricular should be reviewed in order to encourage and capacitate students to widen their intellectual horizon beyond the artificial traditional departmental boundaries.
- Modern techniques for the management of learning resources should be employed in order to address the logistic challenges that discourage students from taking courses across the faculties of science, technology and arts.

6 Alt-i: a historical perspective

Initial interests in language technology that ultimately led to the founding of Alt-i date back to 1978, but it was in 1985 that a small group of one Electrical Engineer and two Physicist started to investigate Text to Speech synthesis of Yoruba in Ibadan, Nigeria. All they were armed with was a book (Electronic Speech Synthesis by Geof Bristow), a microphone and storage oscilloscope. It was extremely difficult to get the required materials in the Nigeria of those days.

Unfortunately, neither the Physicists nor the Engineer realized the relevance of linguistics in their work because the academic environment within which they grew did not provide the necessary impetus for interdisciplinary or multidisciplinary studies between certain fields of study, certainly not between linguistics, physics and engineering. This brought about a lot of misdirected efforts and frustration. By 2001 however, with better access to the scientific literature of computational linguistics and HLT, it had become clear to what was left of the group that HLT is as much an issue in language as it is an issue in technology.

A careful review of the relevant aspects of the scientific literature of linguistics was then undertaken. Contacts were made with some of the known scholars of Yoruba phonology and their insights brought new impetus to the work, leading to the founding of Alt-i in 2001.

The salient point in this historical perspective is that the academic environment that produced the members of the original group did not provide the necessary impetus for the level of cross-disciplinary cooperation demanded by the solutions to the problems the group was addressing. Even though there were many papers that illustrated fruitful connections between linguistics and computer science in the mid 1980's, the Nigerian economy was in such a bad state that the universities could not afford to keep their libraries updated with current publications in any field. Nigerian universities now have noticeably better access to the global academic literature but a systemic weaknesses that does not encourage inter-disciplinary scholarship still subsists and needs to be addressed.

7 Proposal

A project with advocacy and service components aimed at accelerating the development of language technology for African languages is hereby proposed. The aim of the proposed project is to produce lecturers, researchers and other experts in language technology for African languages.

The advocacy component will identify and develop policy thrusts that will encourage the development of language technology and raise awareness at various levels of the importance of linguistics and language technology. These include raising awareness among secondary school students, university undergraduates, cultural activists and relevant policy makers.

Within the service component, in affiliation with a university, a post-graduate course of study aimed at producing a number of PhDs in language technology/computational linguistics within the space of about five to six years is to be developed. The candidates for this post-graduate course shall be university graduates of various relevant fields. The programme shall start with a one year diploma programme of intensive course work in linguistics, computational and cognitive sciences. These will serve to widen the knowledge-base of the participants thereby creating the necessary connections between their backgrounds and various aspects of language technology. Those that attain a high level of achievement in the diploma course may stay on for another six-months to undertake a practical project in language technology. Success in this project will earn such candidates a master degree in language technology or computational linguistics. Graduates of the master programme that attain a high level of performance in the project will be encouraged to stay on for the PhD programme.

The main faculty for the programme shall be drawn from relevant departments in the university. They shall undergo induction courses (locally and overseas) to re-orientate their knowledge towards applications in language technology.

To kick-start the programme, the support of scholars in the international language technology community shall be sought for curricular development as well as teaching. Occasional or short-term visiting lectureships will be accommodated within sabbatical, fellowship and exchange programmes.

As an on-going experiment in this regard, two students of the university of Ibadan are at present working together on Yoruba-English machine translation. One student is a graduate of Computer Science, working towards a master degree in Information Science, while the other is a graduate of Linguistics working towards a master degree in Linguistics. The two students are jointly supervised by a lecturer in Information Science and a lecturer in Linguistics. Even though the computer science graduate has never had any formal training in Linguistics and the Linguistics graduate has never had any formal training in computing, their collaboration has served to widen their knowledge-bases. The student of linguistics is approaching the project from the point of view of comparative syntax and is now able to express syntax rules in the form of context-free-grammar in Prolog, while the Information Science student is approaching the project from the point of view of predicate logic as a knowledge representation formalism and now has a fair understanding of the principles of Yoruba and English grammars.

Even though the experiment is still on-going, the emerging results suggest that the one year of formal study in the proposed diploma programme will provide adequate knowledge and skills for graduates of the physical sciences, computer science, technology, linguistics and psychology to undertake productive research in language technology.

8 Conclusion

The development of language technology for African languages is at a rather embryonic stage. Apart from the efforts in South Africa, there are

little or no coherent programmes on language technology in African universities. National language policies where they exist do not accommodate language technology issues and there is a generally low level of awareness of the benefits derivable from language technology.

With one-third of the world's languages spoken in Africa, there is an urgent need for the development of new techniques that address the peculiar features of these languages and thereby make it possible for the cultures that use them to benefit from the information revolution without having to adopt foreign languages

The implementation of the proposed academic programme in HLT would require active support of the Nigerian government in cooperation and collaboration with other friendly governments as well as various multilateral agencies for funding and other resources. However, in the absence of a language policy and a coherent language technology programme, the advocacy component becomes be the necessary starting point.

Developments in HLT and computational linguistics present avenues for Nigerian universities to re-invigorate the study of linguistics, provide new impetus for students of linguistics and prepare graduates of linguistics for more roles in society than the traditional teaching of local languages at secondary level. Nigerian universities must therefore play an important role in the necessary advocacy.

As the world moves further into the information age, concerted efforts are need to ensure that developments in HLT takes due account of African languages so that African languages and cultures can benefit from the information revolution.

Acknowledgments

Alt-i acknowledges with thanks the funding support of Tiwa Systems Ltd., Bait-al-Hikma, Open Society Initiative for West Africa (OSWIA) and International Development Research Center (IDRC) in its activities.

References

Tunde Adegbola. 2005. Application of tonemic information for search-space reduction in the automatic speech recognition of Yoruba. Unpublished PhD. Thesis, December, 2005, University of Ibadan, Nigeria.

Tunde Adegbola. 2007. Hitting the right tone. ICT Update http://ictupdate.cta.int/en/Feature-Articles/Hitting-the-right-tone

Tunde Adegbola. 2009. The Future of African Languages in a Globalising World. Presented at the Bamako Summit on Multilingualism, 19 - 21 January 2009. Bamako, Mali.

Tunde Adegbola. 2009 Indigenising Human Language Technology for National Development. To be presented at the Africa Regional Center for Information Science (ARCIS), University of Ibadan, Guest Lecture, March 18, 2009.

eLearning Africa. 2007. Interview with Dr. Tunde Adegbola. http://www.elearning-africa.com/newsportal/english/news56_print.php, International Conference on ICT for Development Education

Justus Roux. 2008. HLT Development in Sub-Saharna Africa. Report to COCOSDA/WRITE Workshop, LREC2008, Marrakesh. http://www.ilc.cnr.it/flarenet/documents/lrec2008_cocosda-write_workshop_roux.pdf

Adam Samassekou. 2007. Linguistic Diversity. ICT Update. http://ictupdate.cta.int/en/Regulars/Carte-blanche/Linguistic-diversity

Roger Tucker. 2003. Local language speech technology initiative. http:/www.llsti.org

Initial fieldwork for LWAZI: A Telephone-Based Spoken Dialog System for Rural South Africa

Tebogo Gumede
CSIR
Meraka Institute
PO Box 395
Pretoria, 0001
tgumede@csir.co.za

Madelaine Plauché
CSIR
Meraka Institute
PO Box 395
Pretoria, 0001
mad@brainhotel.org

Abstract

This paper describes sociological fieldwork conducted in the autumn of 2008 in eleven rural communities of South Africa. The goal of the fieldwork was to evaluate the potential role of automated telephony services in improving access to important government information and services. Our interviews, focus group discussions and surveys revealed that Lwazi, a telephone-based spoken dialog system, could greatly support current South African government efforts to effectively connect citizens to available services, provided such services be toll free, in local languages, and with content relevant to each community.

1 Introduction

There is a growing interest in deploying spoken dialog systems (SDSs) in developing regions. In rural communities of developing regions, where infrastructure, distances, language and literacy are barriers to access, but where mobile phones are prevalent, an SDS could be key to unlocking social and economic growth (Barnard et al., 2003). Some notable recent studies in this field include "Tamil Market" (Plauché et al., 2006) and "VoiKiosk" (Agarwal et al., 2008). Both were kiosk-based SDSs providing agricultural information that were tested in rural, semi-literate communities in India. Nasfors (2007) also developed an agricultural information service, aimed at mobile telephone users in Kenya. "Healthline" was evaluated by a small set of community health workers in Pakistan (Sherwani et al., 2007), who had trouble with the voice-based interface, presumably due to their limited literacy. In a more recent study, Sharma et al. (2008) evaluated a SDS designed for caregivers of HIV positive children in Botswana. The researchers found that the users performed equally well using touchtone as speech input, when navigating the system. In the current paper, we expand on this body of work by investigating the potential role for SDSs in connecting rural citizens of South Africa with government services, such as free education opportunities and stipends.

South Africa is the leader in Information and Communications Technology (ICT) in Africa and has the most developed telecommunications network on the continent (SA year book 2006/2007: 131). In particular, mobile phone usage has experienced massive growth due in part to its accessibility by non-literate people and its "leapfrog" development, which skipped the interim solutions adopted in the developed world (Tongia & Subrahmanian, 2006). The amount of mobile phone users in South Africa is an astonishing 30 million people - out of a total population of 47 million (Benjamin, 2007). The percentage of both rural and urban households with mobile phones tripled from 2001 to 2007, while "landline" use declined. The accessibility and widespread use of mobile phones make SDSs a good candidate for low-cost information access.

In South Africa, there are eleven official languages. Private companies, NGOs and government offices who wish to reach South Africans through print or audio, find it extremely costly to do so for each language. Heugh (2007) shows that in terms of speakers' proficiency, there is no single lingua franca for South Africans (see Figure 1). In fact, in 2001, only 3 million of 44 million South Africans were English speakers, the language in which most government messages are currently disseminated (Household survey 2001).

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 59–65,
Athens, Greece, 31 March 2009.

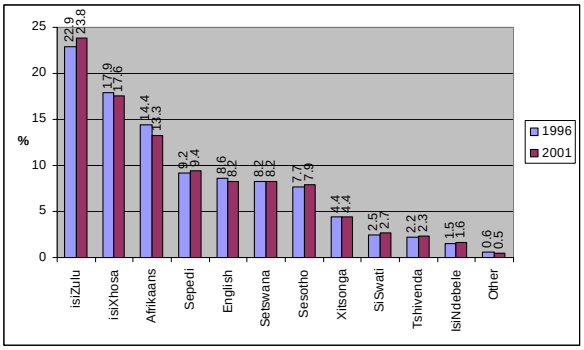


Figure 1: Percentage of speakers per language in South Africa.

Heugh (2007) reports that between 35% and 45% of South Africans above the age of 16 cannot read or write. Illiteracy is disproportionately high for women and for people living in the primarily rural provinces: KwaZulu-Natal, Limpopo and Mpumalanga. Mobile phone use is widespread in these areas among semi-literate citizens and speakers of all languages.

Communities in rural areas struggle to access government services due to their remote locations. Most community members must travel long distances by foot or rare and costly public transport to access basic services. In their longitudinal household study on costs and coping strategies with chronic illnesses, Goudge et al. (2007), for example, found that people in rural areas of South Africa do not go to free health care facilities because they cannot afford transport.

NGOs face the same challenge when trying to reach rural populations. Many produce information to assist households affected by HIV/AIDS, for example, but most of the materials are published on websites; the cost of providing multilingual print materials is often too high. Due to low literacy levels, language, and a lack of infrastructure, the information remains inaccessible to the people who need it, especially those living in rural areas and townships (Benjamin, 2007).

Given the well developed mobile phone network and the relatively sparse alternative options in rural South Africa, the authors believe that multilingual SDSs can provide a low-cost solution to improving the ICT access of citizens who may currently be excluded from government services due to language, literacy and location. However, it is imperative to understand the target users and their environmental context as a first step to designing such a system (Nielsen, 1993). In this paper, we provide background on the current state of rural government service delivery in South Africa and introduce the Lwazi project. We describe our field methods, and finally, we present our findings from the initial field work with design and deployment implications for the Lwazi system.

2. Background

In South Africa, rural citizens are faced with a lack of economic activities and limited access to resources. The South African government is aware of both the need to improve citizen access to services and the specific challenges that rural communities face.

In this section, we note two successful rural initiatives of the South African government (Section 2.1) and we describe Lwazi (Section 2.2), a SDS designed to augment government's accessibility by eligible citizens.

2.1 Rural Initiatives in South Africa

Two national efforts that are successfully connecting rural South African citizens to government services are (1) The Thusong Service Centres (TSCs) and (2) Community Development Workers (CDWs).

Thusong Service Centres (TSCs), formerly known as MPCC (Multi-Purpose Community Centres), were initiated in 1999 as a national initiative to integrate government services into primarily rural communities, where services and participation by citizens was limited due to the long distances they needed to travel (TSC 2008). In June 7, 2008, the 100th TSC was opened. Each TSC is a one-stop centre providing integrated services and information from government to rural community members close to where they live.

Community Development Workers (CDWs) were formed in 2004 as a national initiative to further bridge the gap between government services and eligible citizens, especially the rural poor (CDW 2008). CDWs are members of rural communities who are trained and employed by the Department of Public Service and Administration (DPSA) under the office of the presidency. They work within their communities and coordinate with municipal and provincial offices. The primary responsibilities of a CDW is

to keep informed, notify citizens about events, and inform them about services for which they are eligible then follow up to ensure they successfully receive these services.

2.2 Project Lwazi

As part of the ICT initiative, an ambitious, three-year project is currently being conducted by the Human Language Technology (HLT) research group under the Meraka institute, at the Council for Scientific and Industrial Research (CSIR) in South Africa. This project is funded by the South African Department of Arts and Culture to develop a multilingual telephone-based SDS that would assist South African government service delivery. "Lwazi," derived from the IsiZulu word for *knowledge*, aims to make a positive impact in the daily lives of South Africans by connecting them to government and health services (Lwazi, 2008). The ability of SDSs to overcome barriers of language, literacy, and distances led the Lwazi team to explore a low-cost application that would support the current rural initiatives mentioned in 2.1.

3. Method

First, we consulted previous research on development and technology in South and Southern Africa. We reviewed the most recent census conducted (Statistics SA, 2007) for data on infrastructure, income, language, and technology use.

Then, eleven communities were visited by small, interdisciplinary teams of researchers over a period of 3 months in 2008. Of these eleven centres, two were in peri-urban societies another two in urban and the rest were based in rural communities (Table 1). In each visit, the Lwazi team gained access to the community through the Thusong Service Centres (TSC's) manager. These individuals provided materials, office space, and meetings with CDWs and key people at the TSC.

We conducted between one and five key informant interviews at each site with the TSC employees, CDWs, and community members. In four of the eleven sites, we also conducted a focus group discussion. In two sites, we shadowed a CDW during a typical day. We visited farms, day-care centres, churches, markets, youth centres, clinics, businesses and households.

Community	**Type**	**TSC**	**CDWs**
Sterkspruit	Rural	Yes	Yes
Tshidilamolomo	Rural	Yes	Yes
Botshabelo	Peri-urban	Yes	Yes
Kgautswane	Rural	Yes	Yes
Waboomskraal	Peri-urban	Yes	No
Durban	Urban	No	No
Orhistad	Rural	Yes	Yes
Sediba	Rural	Yes	Yes
Atteridgeville	Urban	Yes	Yes
Laingsburg	Rural	Yes	Yes
Vredendal	Rural	Yes	Yes

Table 1: Sites visited in Spring 2008.

Data collection in these communities was primarily to investigate the suitability of the Lwazi SDS and to determine key user and contextual factors that would drive its design. In particular, we sought to:

- Gather rural community information needs.
- Investigate how people currently get information.
- Determine which cultural factors would impact the Lwazi system.
- Determine level of technical competency.
- Gauge interest in a low-cost SDS that improves access to government services.

4. Results

In this section, we present our overall results from field visits in eleven communities of South Africa (Section 4.1). In particular, we report on factors that influence the design and potential uptake of the Lwazi system in this context: the information needs and sources (Section 4.2), cultural and social factors (Section 4.3), suitability of the technology (Section 4.4), and user experience (Section 4.5).

4.1 Overall Results

The eleven communities we visited were located throughout seven of the nine provinces of South Africa. They varied greatly in available infrastructure and languages spoken. They shared an economic dependency on nearby cities and, in

some cases, reported social problems. These communities also share a dependency on government social grants.

During interviews and focus group discussions with government employees and community members, interviewees identified what they perceived as the primary problems in their communities. Across all eleven sites visited, unemployment was most often reported as the primary problem (Figure 2). In fact, our team observed that in at least 8 of the sites visited, the community's livelihood was entirely sustained by government grants.

After unemployment, access to health and social services was viewed as a primary problem in six of the sites visited. Crime and substance abuse were also reported as community problems.

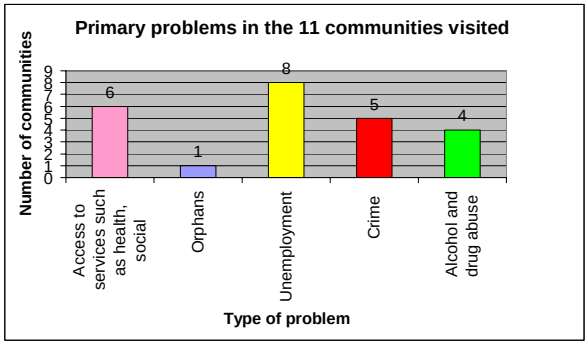


Figure 2: Number of the eleven communities that site these as primary problems in the community, as reported by interviewees.

There are four mobile providers in South Africa namely Cell-C, MTN, Virgin Mobile and Vodacom. The two landline companies, Neo-tell and Telkom are not familiar in the communities visited by Lwazi team. Community members prefer and use mobile phones because of ease of use and accessibility. Figure 3 illustrates the use of mobile providers in the eleven visited communities.

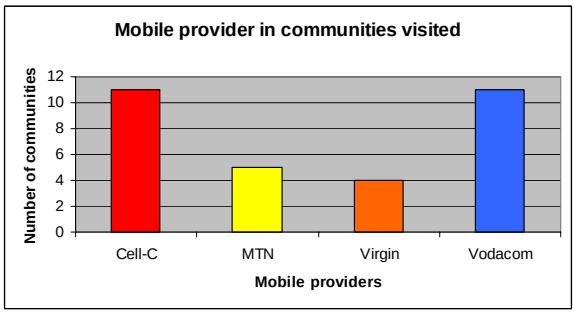


Figure 3: Mobile provider in communities visited

4.2. Information Needs and Sources

The majority of communities visited reported lack of economic activity as the primary problem in the community, and as could be expected, we observed very high levels of unemployment. Grants are offered by the South African government to address the imbalance and stimulate the economy of these areas. There are six types of grants, namely: War Veteran grant, Old Age grant, Disability grant, Care dependency grant, Foster care grant and Child support grant. Citizens can apply for these at their nearest local South African social security agency (SASSA) or district office.

Figure 4 shows that all eleven communities visited received Vukuzenzele magazine, a monthly government information sharing medium. This is however not effective in these communities where literacy levels are low. This, as mentioned earlier, is one of the problems the government is trying to address. The second commonly used source of information was the CDWs. This is a useful source because they are the 'foot soldiers' of the government; they are responsible for door to door visits, collecting and delivering information.

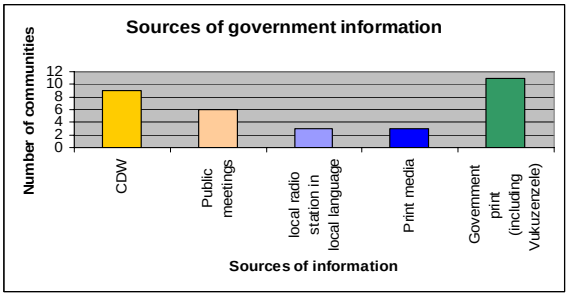


Figure 4: sources of government information

At the time of the visits, the eleven communities received information from the Thusong Service Centres. Government departments, NGOs, municipalities and research institutes such the CSIR, used the TSCs as a platform to disseminate information. At a more grass roots level, African communities in South Africa share information by word of mouth. Local radio stations and newspapers in local languages are also important sources of information.

4.3 Cultural and Social Factors

The population in the communities we visited consists mostly of older people taking care of grandchildren whose parents work in nearby

cities and visit only once or twice a year. As we have previously mentioned, the lack of economic activity means that these communities depend heavily on government social grants. Older people in these communities are often unable to read and write. In some cases, their low literacy levels and possible unfamiliarity with technology restricts their ability to use the cheaper "texting" feature of their mobile phones. In the communities we visited, ten out of the eleven official South African languages were spoken. Each community typically spoke two or more languages. A Lwazi system that delivers information about SASSA services would be more accessible to the older, rural population than current print methods; the proposed system would support a majority of the eleven languages in order to offer a better information channel.

4.4 Suitability of Technology

The research prior to the design of the Lwazi system investigated how to ensure that the proposed system will be suitable to the lifestyle of the community to be served. We do know that currently, communities have other means of accessing government information, including the free "Vukuzenzele" monthly magazine and local radio stations. Like these current means, Lwazi must be free in order to be effective and it must contain content that is locally-relevant to these peri-urban and rural communities. Government departments will find Lwazi to be a very useful and low-cost way of disseminating their information. Rural South Africans will benefit from the alternative source of critical information.

4.5 User Expertise

We also sought to evaluate the current expertise of potential Lwazi system users with telephony and other ICT technologies. The user expertise of telephony systems differ between young and old. As mentioned earlier, households have at least one cell phone. The older members of the community use it to call and receive calls from friends and children in neighbouring urban areas. They do not know how to send text messages. Some children have saved the full 'Please call me' line as a quick dial so that their elder family members can just press it in cases of emergency.

The young people on the other hand are well versed with telephony technology. Most of them are also familiar with the basic use of a computer, despite their limited access to them. The Lwazi system must be as simple as making a phone call to a friend or relative in order to be accessible to all. In most households, however, there is someone who is technically competent. Based on our fieldwork and recent user studies of SDSs in developing regions, a Lwazi system could be useable in the rural context of South Africa, especially among the elderly and those who do not read and write.

5. Discussion

5.1 Potential Uptake

We saw two main areas where a telephony service could be very useful. The first is in supporting communication between community and government. For example, a multilingual, automated service could direct calls from community members to the appropriate TSC office or CDW, or perhaps provide locally-relevant information such as office hours of operation, directions to the office, and eligibility requirements for services. Such a service might reduce the amount of calls that a TSC office or CDW would need to take personally. It could also likely save a community member a trip if they were sure beforehand what paperwork they needed to bring and when the office was open. It is important to mention here that the project will require a buy-in from the local councillors of the communities we will be piloting in.

The second area in which a telephony service could be useful would be in facilitating internal communication among government service providers. CDWs may need to meet community members face to face whenever possible. Coordinating with government staff across the municipality, district, province, or country could happen remotely and efficiently if government staff could use an automated telephony service to send audio messages to several staff members at once. The national coordinator for CDWs, for example, could notify every CDW in the country of upcoming events and policy changes with a single phone call.

Many government officials, including Thusong centre managers, felt the system might assist them in communicating with the communities they serve. There was, however, a concern from one site that there are sections of the population that do not have mobile

connection or a reliable source of electricity to charge their phones. These could be the communities that need government services the most. In these cases, Lwazi will have to play a supportive role to the existing services provided by the CDWs, rather than allowing direct access by community members.

Our field work revealed the effectiveness of national government programs to connect rural citizens to available government services. Our major finding was that although particulars about the communities differed, individuals in the eleven communities visited experienced barriers to information access that could be mitigated with automated telephony services, provided that such services are toll free and localized to the language and information relevant to the particular rural community. Whereas infrastructure such as roads, services, and in some cases, electricity were limited, the mobile phone network of South Africa is reliable and widespread. We feel optimistic that the Lwazi system will build on the available infrastructure to transcend the barriers of geography and improve the connection between citizens and services.

5.2 Challenges

In a large and culturally diverse country such as South Africa, deploying a SDS intended to provide universal access is a great challenge. Designing for any given user group often requires a multiple iterations of testing and user feedback. Our fieldwork revealed a diverse set of end users; a successful design will require a greater investment in time and resource to gather detailed and accurate information about rural South Africans. Although we plan to rely on our partners (TSC and CDWs) on the ground for a great deal of this information, we believe it is an ambitious goal to expect deployment of the Lwazi project in eleven languages country wide by summer 2009.

Not only is the technological aspect very ambitious, this kind of national government sponsored system requires tactful management of stakeholders. The success of the Lwazi project relies on community members, government partners, researchers, NGO's, and corporate interests, all of whom have conflicting needs and interests. Our team recognizes the importance of managing stakeholder interests and has devised a problem structuring method to facilitate feedback and discussion (Plauché et al., submitted).

Community buy-in is critical to the success of an ICT deployment. We found not only that the TSC and CDW national coordinators but also each of the communities visited were all excited about the potential of the proposed system. Generally, rural communities are comfortable with the use of a mobile phone. But there is an age difference in preference of different applications. Because Lwazi is voice-based, the senior citizens of the community will be more likely to be excited about it than the younger generations. Younger South Africans are comfortable with the cheaper, text interface to mobile phones. We recognise that Lwazi may not suit the needs of all South Africans, but we aim to make it accessible to those who are historically excluded. In doing so, we hope to have an overall impact in this country where only 26% of the population has a Matric or tertiary education (Stats SA, 2007).

6. Conclusion

In this paper we evaluated the potential role of Lwazi, a proposed telephone-based SDS, in improving rural access to important government services. The Lwazi project will create an open platform for telephone-based services for government to provide information in all eleven languages. Lwazi will be especially useful if it can reduce cost or the distances that people travel to access government services and that the distances that government workers travel to check in with municipal offices. Our team plans to conduct pilots in two communities in the summer of 2009. A successful pilot in one of these communities will then burgeon into a national service for all South Africans to empower themselves through improved access to information, services and government resources.

References

Aditi Sharma, Madelaine Plauché, Etienne Barnard, Christiaan and Kuun. (To appear). *HIV health information access using spoken dialog systems*: *Touchtone vs. Speech*. In *Proc. of IEEE ICTD'09*, 2009.

Bernhard Suhm. 2008. *IVR Usability Engineering using Guidelines and Analyses of end-to-end calls.* in D. Gardener-Bonneau and H.E. Blanchard (Eds). *Human Factors and Voice Interactive Systems*. pp. 1-41, Second Edition, Springer Science: NY, USA.

CDW 2008: www.info.gov.za/issues/cdw.htm. Accessed August 20, 2008.

Etienne Barnard, Lawrence Cloete and Hina Patel. 2003. *Language and Technology Literacy Barriers to Accessing Government Services*. *Lecture Notes in Computer Science*, vol. 2739, pp. 37-42.

Government Communication and Information System. 2007. South African Year Book 2006/2007.

Jahanzeb *Sherwani* , Nosheen *Ali*, Sarwat *Mirza*, Anjum *Fatma*, Yousuf *Memon*, Mehtab *Karim*, Rahul *Tongia*, Roni *Rosenfeld.* 2007. *Healthline: Speech-based Access to Health Information by low-literate users.* in Proc. of IEEE ICTD'07, Bangalore, India.

Jane Goudge, Tebogo Gumede, Lucy Gilson, Steve Russell, Steve Tollman & Anne Mills. 2007. *Coping with the cost burdens of illness: Combining qualitative and quantitative methods in longitudinal household research*. Scandinavian journal of public health. 35 (Suppl 69), 181 – 185.

Kathleen Heugh. 2007. *Language and Literacy issues in South Africa.* In Rassool, Naz (ed) Global Issues in Language, Education and Development. Perspectives from Postcolonial Countries. Clevedon: Multilingual matters, 187-217.

Lwazi. 2008. http://.meraka.org.za/lwazi. Accessed August 20, 2008.

Madelaine Plauché, Alta De Waal, Aditi Sharma, and Tebogo Gumede. (submitted). 2008. *Morphological Analysis: A method for selecting ICT applications in South African government service delivery*. ITID.

Madelaine Plauché, Udhyakumar Nallasamy, Joyojeet Pal, Chuck Wooters and Divya. Ramachandran. 2006. *Speech Recognition for Illiterate Access to Information and Technology*. in *Proc. of IEEE ICTD'06*, pp. 83-92.

Ministry of Public Service and Administration. 2007. *Community Development Workers Master Plan*.

Nielsen Jakob. 1993. *Usability Engineering*. AP Professional, Boston, MA, USA.

PANSALB. 2001. *Language use and Language Interaction in South Africa: A National Sociolinguistic Survey Summary Report.* Pan South African Language Board. Pretoria

Pernilla Nasfors. 2007. *Efficient Voice Information Services for Developing Countries, Master Thesis, Department of Information technology*, Uppsala University, Sweden.

Peter Benjamin. 2007. *The cellphone information channel for HIV/AIDS.* Unpublished information newsletter.

Rahul Tongia, and Eswaran Subrahmanian. 2006. *Information and Communications Technology for Development (ICT4D) - A design challenge?* In *Proc. of IEEE ICTD'06*, Berkeley, CA.

Sheetal Agarwal, Arun Kumar, AA Nanavati and Nitendra Rajput. 2008. *VoiKiosk: Increasing Reachability of Kiosks in Developing Regions*, in *Proc. of the 17th International Conference on World Wide Web*, pp. 1123-1124, 2008.

Statistics South Africa. 2007. Community Survey: http://www.statsa.gov.za/publications/P0301/P0301.pdf (last accessed 15 Sept 2008).

Tebogo Gumede, Madelaine Plauché, and Aditi Sharma. 2008. *Evaluating the Potential of Automated Telephony Systems in Rural Communities*. CSIR biannual conference.

TCS 2008: www.thusong.gov.za/. Accessed August 20, 2008.

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

Rigardt Pretorius
School of Languages
North-West University
Potchefstroom, South Africa
Rigardt.Pretorius@nwu.ac.za

Ansu Berg
School of Languages
North-West University
Potchefstroom, South Africa
Ansu.Berg@nwu.ac.za

Laurette Pretorius
School of Computing
University of South Africa
and
Meraka Institute, CSIR
Pretoria, South Africa
pretol@unisa.ac.za

Biffie Viljoen
School of Computing
University of South Africa
Pretoria, South Africa
viljoe@unisa.ac.za

Abstract

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. Therefore, 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 such as POS tagging and shallow parsing, etc. Indeed, the approach ensures that an aspect such as orthography does not obfuscate sound linguistics and, ultimately, proper semantic analysis, which remains the ultimate aim of linguistic analysis and therefore also computational linguistic analysis.

1 Introduction

Words, syntactic groups, clauses, sentences, paragraphs, etc. usually form the basis of the analysis and processing of natural language text. However, texts in electronic form are just sequences of characters, including letters of the alphabet, numbers, punctuation, special symbols, whitespace, etc. The identification of word and sentence boundaries is therefore essential for any further processing of an electronic text. *Tokenisation* or word segmentation may be defined as the process of breaking up the sequence of characters in a text at the word boundaries (see, for example, Palmer, 2000). Tokenisation may therefore be regarded as a core technology in natural language processing.

Since disjunctive orthography is our focus, we distinguish between an orthographic word, that is a unit of text bounded by whitespace, but not containing whitespace, and a linguistic word, that is a sequence of orthographic words that together functions as a member of a word category such as, for example, nouns, pronouns, verbs and adverbs (Kosch, 2006). Therefore, tokenisation may also be described as the process of identifying linguistic words, henceforth referred to as tokens.

While the Bantu languages are all agglutinative and exhibit significant inherent structural similarity, they differ substantially in terms of their orthography. The reasons for this difference are both historical and phonological. A detailed

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 66–73,
Athens, Greece, 31 March 2009.

discussion of this aspect falls outside the scope of this article, but the interested reader is referred to Cole (1955), Van Wyk (1958 & 1967) and Krüger (2006).

Setswana, Northern Sotho and Southern Sotho form the Sotho group belonging to the South-Eastern zone of Bantu languages. These languages are characterised by a disjunctive (also referred to as semi-conjunctive) orthography, affecting mainly the word category of verbs (Krüger, 2006:12-28). In particular, verbal prefixal morphemes are usually written disjunctively, while suffixal morphemes follow a conjunctive writing style. For this reason 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, which includes Zulu, Xhosa, Swati and Ndebele.

The following research question arises: 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 (Dixon and Aikhenvald, 2002:8) and allow a uniform approach.

The structure of the paper is as follows: The introduction states and contextualises the research question. The following section discusses tokenisation in the context of the South African Bantu languages. Since the morphological structure of the Setswana verb is central to the tokenisation problem, the next section comprises a brief exposition thereof. The paper then proceeds to discuss the finite-state computational approach that is followed. This entails the combination of two tokeniser transducers and a finite-state (rule-based) morphological analyser. The penultimate section concerns a discussion of the computational results and insights gained. Possibilities for future work conclude the paper.

2 Tokenisation

Tokenisation for alphabetic, segmented languages such as English is considered a relatively simple process where linguistic words are usually delimited by whitespace and punctuation. This task is effectively handled by means of regular expression scripts. Mikeev (2003) however warns that "errors made at such an early stage are very likely to induce more errors at later stages of text processing and are therefore very dangerous." The importance of accurate tokenisation is also emphasised by Forst and Kaplan (2006). While Setswana is also an alphabetic segmented language, its disjunctive orthography causes token internal whitespace in a number of constructions of which the verb is the most important and widely occurring. Since the standard tokenisation issues of languages such as English have been extensively discussed (Farghaly, 2003; Mikeev, 2003; Palmer, 2000), our focus is on the challenge of Setswana verb tokenisation specifically. We illustrate this by means of two examples:

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. In a study on corpus design for Setswana lexicography Otlogetswe (2007) claims that *a* is the most frequent "word" in his 1.3 million "words" Setswana corpus (Otlogetswe, 2007:125). In reality, the orthographic word *a* in Setswana could be any of several linguistic words or morphemes. 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;

*Dintswa/ 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*.

The above examples emphasise the importance of correct tokenisation of corpora, particularly in the light of the increased exploitation of electronic corpora for linguistic and lexicographic research. In particular, the correct tokenisation of verbs in disjunctively written languages is crucial for all reliable and accurate corpus-based research. Hurskeinen et al. (2005:450) confirm this by stating that "a care-

fully designed tokeniser is a prerequisite for identifying verb structure in text".

3 Morphological Structure of the Verb in Setswana

A complete exposition of Setswana verb morphology falls outside the scope of this article (see Krüger, 2006). Main aspects of interest are briefly introduced and illustrated by means of examples.

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. Krüger (2006:36) describes the root as "a lexical morpheme [that] can be defined as that part of a word which does not include a grammatical morpheme; cannot occur independently as in the case with words; constitutes the lexical meaning of a word and belongs quantitatively to an open class".

3.1 Prefixes of the Setswana verb

The verbal root can be preceded by several prefixes (cf. Krüger (2006:171-183):

Subject agreement morphemes: The subject agreement morphemes, written disjunctively, include non-consecutive subject agreement morphemes and consecutive subject agreement morphemes. This is the only modal distinction that influences the form of the subject morpheme. The same subject agreement morpheme therefore has a consecutive as well as a non-consecutive form. 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).

Object agreement morphemes: The object agreement morpheme is written disjunctively in most instances, for example *ba* ***di*** *bona* (they see it).

The reflexive morpheme: The reflexive morpheme *i-* (-self) is always written conjunctively to the root, for example *o* ***i****pona* (he sees himself).

The aspectual morphemes: 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: 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*: 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).

3.2 Suffixes of the Setswana verb

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

Verb-final morphemes: Verbal-final suffixes *a*, *e*, the relative *-ng* and the imperative *–ng,* for example, *ga ba ithut****e*** (they are not learning).

The causative suffix *-is-*: Example, *o rek****is****a* (he sells (he causes to buy)).

The applicative suffix *-el-*: Example, *o bal****el****a* (she reads for).

The reciprocal suffix *-an-*: Example, *re a thus****an****a* (we help each other).

The perfect suffix *-il-*: Example, *ba utlw****il****e* (they heard).

The passive suffix *-w-*: Example, *o rom****i****wa* (he is sent).

3.3 Auxiliary verbs and copulatives

Krüger (2006:273) states that "Syntactically an auxiliary verb is a verb which must be followed by a complementary predicate, which can be a verb or verbal group or a copulative group or an auxiliary verbal group, because it cannot function in isolation". Consider the following example of the auxiliary verb **tlhola**: *re tlhola/ re ba thusa/* (we always help them). For a more detailed discussion of auxiliary verbs in Setswana refer to Pretorius (1997).

Copulatives function as introductory members to non-verbal complements. The morphological forms of copula are determined by the copulative relation and the type of modal category in which they occur. These factors give rise to a large variety of morphological forms (Krüger, 2006: 275-281).

3.4 Formation of verbs

The formation of Setswana verbs is governed by a set of linguistic rules according to which the various prefixes and suffixes may be sequenced

to form valid verb forms (so-called morhotactics) and by a set of morphophonological alternation rules that model the sound changes that occur at morpheme boundaries. These formation rules constitute a model of Setswana morphology that forms the basis of the finite-state morphological analyser, discussed in subsequent sections.

This model, supported by a complete set of known, attested Setswana roots, may be used to recognise valid words, including verbs. It will not recognise either incorrectly formed or partial strings as words. The significance of this for tokenisation specifically is that, in principle, the model and therefore also the morphological analyser based on it can and should recognise only (valid) tokens.

Morphotactics: While verbs may be analysed linearly or hierarchically, our computational analysis follows the former approach, for example:

ba a kwala (they write)
`Verb(INDmode),(PREStense,Pos):AgrSubj-Cl2+AspPre + [kwal]+Term`
o tla reka (he will buy)
`Verb(INDmode),(FUTtense,Pos):AgrSubj-Cl1+TmpPre+[rek]+Term`
ke dirile (I have worked)
`Verb(INDmode),(PERFtense,Pos):AgrSubj-1P-Sg+[dir]+Perf+Term`

The above analyses indicate the part-of-speech (verb), the mode (indicative) and the tense (present, future or perfect), followed by a ':' and then the morphological analyses. The tags are chosen to be self-explanatory and the verb root appears in square brackets. For example the first analysis is *ba*: subject agreement class 2; *a*: aspectual prefix; *kwal*: verb root; *a*: verb terminative (verb-final suffix). The notation used in the presentation of the morphological analyses is user-defined.

In linear analyses the prefixes and suffixes have a specific sequencing with regard to the verbal root. We illustrate this by means of a number of examples. A detailed exposition of the rules governing the order and valid combinations of the various prefixes and suffixes may be found in Krüger (2006).

Object agreement morphemes and the reflexive morpheme always appear directly in front of the verbal root, for example *le a **di** reka* (he buys it). No other prefix can be placed between the object agreement morpheme and the verbal root or between the reflexive morpheme and the verbal root.

The position of the negative morpheme *ga* is always directly in front of the subject agreement morpheme, for example, ***ga ke** di bône*. (I do not see it/them).

The negative morpheme *sa* follows the subject agreement morpheme, for example, *(fa) **le sa** dire* ((while) he is not working).

The negative morpheme *se* also follows the subject agreement morpheme, for example, *(gore) **re se** di je* ((so that) we do not eat it). However, if the verb is in the imperative mood the negative morpheme *se* is used before the verbal root, for example, ***Se** kwale!* (Do not write!).

The aspectual morphemes always follow the subject agreement morpheme, for example, ***ba sa** dira* (they are still working).

The temporal morpheme also follows the subject agreement morpheme, for example, ***ba tla** dira* (they shall work).

Due to the agglutinating nature of the language and the presence of long distance dependencies, the combinatorial complexity of possible morpheme combinations makes the identification of the underlying verb rather difficult. Examples of rules that assist in limiting possible combinations are as follows:

The object agreement morpheme is a prefix that can be used simultaneously with the other prefixes in the verb, for example, *ba a **di** bona* (they see it/them).

The aspectual morphemes and the temporal morpheme cannot be used simultaneously, for example, *le ka ithuta* (he can learn) and l*e tla ithuta* (he will learn).

Since (combinations of) suffixes are written conjunctively, they do not add to the complexity of the disjunctive writing style prevalent in verb tokenisation.

Morphophonological alternation rules: Sound changes can occur when morphemes are affixed to the verbal root.

The prefixes: The object agreement morpheme of the first person singular *ni/n* in combination with the root causes a sound change and this combination is written conjunctively, for example *ba ni-bon-a* > *ba mpona* (they see me). In some instances the object agreement morpheme of the third person singular and class 1 causes sound changes when used with verbal roots beginning with *b-*. They are then written conjunctively, for example, *ba mo-bon-a* > *ba mmona* (they see him).

When the subject agreement morpheme *ke* (the first person singular) and the progressive morpheme *ka* are used in the same verb, the sound change *ke ka* > *nka* appears, for example, *ke ka opela* > *nka opela* (I can sing).

The suffixes: Sound changes also occur under certain circumstances, but do not affect the conjunctive writing style.

Summarising, the processing of electronic Setswana text requires precise tokenisation; the disjunctive writing style followed for verb constructions renders tokenisation on whitespace inappropriate; morphological structure is crucial in identifying valid verbs in text; due to the regularity of word formation, linguistic rules (morphotactics and morphophonological alternation rules) suggest a rule-based model of Setswana morphology that may form the basis of a tokeniser transducer, and together with an extensive word root lexicon, also the basis for a rule-based morphological analyser. Since the Bantu languages exhibit similar linguistic structure, differences in orthography should be addressed at tokenisation / morphological analysis level so that subsequent levels of computational (syntactic and semantic) analysis may benefit optimally from prevalent structural similarities.

4 Facing the Computational Challenge

Apart from tokenisation, computational morphological analysis is regarded as central to the processing of the (agglutinating) South African Bantu languages (Bosch & Pretorius, 2002, Pretorius & Bosch, 2003). Moreover, standards and standardisation are pertinent to the development of appropriate software tools and language resources (Van Rooy & Pretorius, 2003), particularly for languages that are similar in structure. While such standardisation is an ideal worth striving for, it remains difficult to attain. Indeed, the non-standard writing styles pose a definite challenge.

4.1 Other approaches to Bantu tokenisation

Taljard and Bosch (2005) advocate an approach to word class identification that makes no mention of tokenisation as a central issue in the processing of Northern Sotho and Zulu text. For Northern Sotho they propose a hybrid system (consisting of a tagger, a morphological analyser and a grammar) "containing information on both morphological and syntactic aspects, although biased towards morphology. This approach is dictated at least in part, by the disjunctive method of writing." In contrast, Hurskainen et al. (2005) in their work on the computational description of verbs of Kwanjama and Northern Sotho, concludes that "a carefully designed tokeniser is a prerequisite for identifying verb structures in text". Anderson and Kotzé (2006) concur that in their development of a Northern Sotho morphological analyser "it became obvious that tokenisation was a problem that needed to be overcome for the Northern Sotho language as distinct from the ongoing morphological and morpho-phonological analysis".

4.2 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.

4.3 Morphological analyser

The finite-state morphological analyser prototype for Setswana, developed with the Xerox finite state toolkit (Beesley and Karttunen, 2003), 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 (Pretorius et al., 2005 and 2008). 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.

4.4 Tokeniser

Since the focus is on verb constructions, the Setswana tokeniser prototype makes provision for punctuation and alphabetic text, but not yet for the usual non-alphabetic tokens such as dates, numbers, hyphenation, abbreviations, etc. A grammar for linguistically valid verb constructions is implemented with xfst regular expressions. By way of illustration we show a fragment thereof, where SP represents a single blank character, WS is general whitespace and SYMBOL is punctuation. In the fragment of xfst below '...' indicates that other options have been removed for conciseness and is not strict xfst syntax :

```
define WORD [Char]+[SP | SYMBOL];
define WORDwithVERBEnding [Char]+[a | e
| n g] [SP | SYMBOL];
echo >>> define object concords
define OBJ [g o | r e | l o | l e | m o
| b a | o | e | a | s e | d i | b o]
WS+;
echo >>> define subject concords
define SUBJ [k e | o | r e | l o | l e |
a | b a | e | s e | d i | b o | g o]
WS+;
echo >>> define verb prefixes
echo >>> define indicative mode
define INDPREF [(g a WS+) SUBJ ([a | s a
] WS+) ([a | k a | s a] WS+) (t l a WS+)
(OBJ)];
define VPREF [...| INDPREF | ...];
echo >>> define verb groups
define VGROUP [VPREF WORDwithVERBEnd-
ing];
echo >>> define tokens
define Token [VGROUP | WORD | ...];
```

Finally, whitespace is normalised to a single blank character and the right-arrow, right-to-left, longest match rule for verb tokens is built on the template

```
A ->@ B || L _ R;
```

where A, B, L and R are regular expressions denoting languages, and L and R are optional (Beesley and Karttunen, 2003:174).

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? Both these questions are discussed in the subsequent section.

4.5 Methodology

Our methodology is based on a combination of a comprehensive and reliable morphological analyser for Setswana catering for disjunctively written verb constructions (see section 5.3), a verb tokeniser transducer (see section 5.4) and a tokeniser transducer that tokenises on whitespace. The process is illustrated in Figure 1. 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.

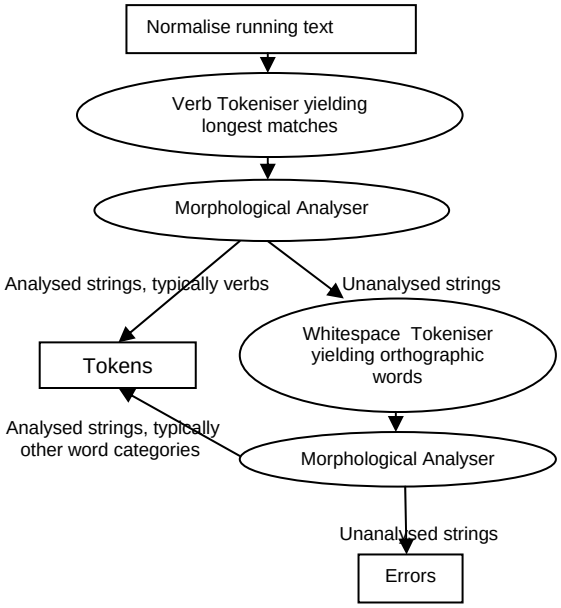


Figure 1: Tokenisation procedure

Tokenisation procedure:

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.

This procedure yields the tokens obtained by computational means. Errors are typically strings that could not be analysed by the morphological analyser and should be rare. These strings should be subjected to human elicitation. 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.

The test data: 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), (FUT-tense,Neg): AgrSubj-Cl5`
ditsebe
`NPre10+[tsebe]`

5 Results and Discussion

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
$H \cap A$	383 (93.6%)	193 (92.8%)
$A \setminus H$	29	9
$H \setminus A$	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

The F-score of 0.93 in Table 2 may be considered a promising result, given that it was obtained on the most challenging aspect of Setswana tokenisation. The approach scales well and may form the basis for a full scale, broad coverage tokeniser for Setswana. A limiting factor is the as yet incomplete root lexicon of the morphological analyser. However, this may be addressed by making use of a guesser variant of the morphological analyser that contains consonant/vowel patterns for phonologically possible roots to cater for absent roots.

It should be noted that the procedure presented in this paper yields correctly tokenised and morphologically analysed linguistic words, ready for subsequent levels of parsing.

We identify two issues that warrant future investigation:

- 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.

6 Future work

- The extension of the morphological analyser to include complete coverage of the so-called closed word categories, as well as comprehensive noun and verb root lexicons;
- The refinement of the verb tokeniser to cater for a more extensive grammar of Setswana verb constructions and more sophisticated ways of reducing the length of invalid longest right-to-left matches;
- The application of the procedure to large text corpora.

Acknowledgements

This material is based upon work supported by the South African National Research Foundation 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.

References

Anderson, W.N. and Kotzé, P.M. Finite state tokenisation of an orthographical disjunctive agglutinative language: The verbal segment of Northern Sotho. In *Proceedings of the 5th International Conference on Language Resources and Evalution*, Genoa, Italy, May 22-28, 2006.

Bosch, S.E. and Pretorius, L. 2002. The significance of computational morphology for Zulu lexicography. *South African Journal of African Languages*, 22(1):11-20.

Cole, D.T. 1955. *An Introduction to Tswana Grammar*. Longman, Cape Town, South Africa.

Dixon, R.M.W. and Aikhenvald, A.Y. 2002. *Word: A Cross-linguistic Typology*. Cambridge University Press, Cambridge, UK.

Forst, M. and Kaplan, R.M. 2006. The importance of precise tokenization for deep grammars. In *Proceedings of the 5th International Conference on Language Resources and Evalution*, Genoa, Italy, May 22-28, 2006.

Hurskeinen, A., Louwrens, L. and Poulos, G. 2005 Computational description of verbs in disjoining writing systems. *Nordic Journal of African Studies*, 14(4): 438-451.

Kosch, I.M. 2006. *Topics in Morphology in the African Language Context*. Unisa Press, Pretoria, South Africa.

Krüger, C.J.H. 2006. *Introduction to the Morphology of Setswana*. Lincom Europe, München, Germany.

Megerdoomian, K. 2003. Text mining, corpus building and testing. *In Handbook for Language Engineers*, Farghaly, A. (Ed.). CSLI Publications, California, USA.

Mikheev, A. 2003. Text segmentation. In *The Oxford Handbook of Computational Linguistics*, Mitkov, R. (Ed.) Oxford University Press, Oxford, UK.

Otlogetswe, T.J. 2007. *Corpus Design for Setswana Lexicography*. PhD thesis. University of Pretoria, Pretoria, South Africa.

Palmer, D.D. 2000. Tokenisation and sentence segmentation. *In Handbook of natural Language Processing*, Dale, R., Moisl, H. And Somers, H. (Eds.). Marcel Dekker, Inc., New York, USA.

Pretorius, R.S. 1997. *Auxiliary Verbs as a Subcategory of the Verb in Tswana*. PhD thesis. PU for CHE, Potchefstroom, South Africa.

Pretorius, L and Bosch, S.E. 2003. Computational aids for Zulu natural language processing. *South African Linguistics and Applied Language Studies*, 21(4):267-281.

Pretorius, R., Viljoen, B. and Pretorius, L. 2005. A finite-state morphological analysis of Setswana nouns. *South African Journal of African Languages*, 25(1):48-58.

Pretorius, L., Viljoen, B., Pretorius, R. and Berg, A. 2008. Towards a computational morphological analysis of Setswana compounds. *Literator*, 29(1):1-20.

Schiller, A. 1996. Multilingual finite-state noun-phrase extraction. In *Proceedings of the ECAI 96 Workshop on Extended Finite State Models of Lang*uage, Kornai, A. (Ed.).

Taljard, E. 2006 Corpus based linguistic investigation for the South African Bantu languages: a Northern Sotho case study. *South African journal of African languages*, 26(4):165-183.

Taljard, E. and Bosch, S.E. 2006. A Comparison of Approaches towards Word Class Tagging: Disjunctively versus Conjunctively Written Bantu Languages. *Nordic Journal of African Studies*, 15(4): 428-442.

Van Wyk, E.B. 1958. *Woordverdeling in Noord-Sotho en Zoeloe. 'n Bydrae tot die Vraagstuk van Woordidentifikasie in die Bantoetale.* University of Pretoria, Pretoria, South Africa.

Van Wyk, E.B. 1967. The word classes of Northern Sotho. *Lingua*, 17(2):230-261.

Interlinear glossing and its role in theoretical and descriptive studies of African and other lesser-documented languages

Dorothee Beermann
Norwegian University of Science and Technology
Trondheim, Norway
dorothee.beermann@hf.ntnu.no

Pavel Mihaylov
Ontotext,
Sofia, Bulgaria
pavel@ontotext.com

Abstract

In a manuscript William Labov (1987) states that although linguistics is a field with a long historical tradition and with a high degree of consensus on basic categories, it experiences a fundamental devision concerning the role that quantitative methods should play as part of the research progress. Linguists differ in the role they assign to the use of natural language examples in linguistic research and in the publication of its results. In this paper we suggest that the general availability of richly annotated, multi-lingual data directly suited for scientific publications could have a positive impact on the way we think about language, and how we approach linguistics.We encourage the systematic generation of linguistic data beyond what emerges from fieldwork and other descriptive studies and introduce an online glossing tool for textual data annotation. We argue that the availability of such an online tool will facilitate the generation of in-depth annotated linguistic examples as part of linguistic research. This in turn will allow the build-up of linguistic resources which can be used independent of the research focus and of the theoretical framework applied. The tool we would like to present is a non-expert-user system designed in particular for the work with lesser documented languages. It has been used for the documentation of several African languages, and has served for two projects involving universities in Africa.

1 Introduction

The role that digital tools play in all fields of modern linguistics can not be underestimated. This is partially due to the success of computational linguistics and its involvement in fields such as lexicography, corpus linguistics and syntactic parsing, to just name some. Most crucially however this development is due to the success of IT in general and in particular to the World Wide Web which has created new standards also for linguistic research. Through the internet our perception of 'data' and publication of linguistic results has changed drastically only in a matter of a few years. Although the development of language resources and language technology for African languages is increasing steadily, the digital revolution and the resources and possibilities it offers to linguistics are mostly seized by researchers in the First World connected to work centering around the key languages. For this paper we would like to conceive of this situation in terms of lost opportunities: At present formal linguistics and linguistic research conducted on Third World languages are mostly undertaken with very little knowledge of each other and hardly any exchange of research results. Likewise, language documentation, which has roots in language typology and computational linguistics, only partially coincides with work in African linguistics. Yet, it is evident that the general availability of linguistic material from a bigger sample of languages will eventually not only affect the way in which we think about language, but also might have an impact on linguistic methodology and on the way we go about linguistic research. If you are only a few mouse clicks away from showing that a certain generalization only holds for a limited set of languages, but truly fails to describe a given phenomenon for a wider sample, statements claiming linguistic generality have to be phrased much more carefully. Our perception of the nature of language could truly benefit from general access to representative multilingual data. It therefore would seem a linguistic goal in itself to (a) work towards a more general

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 74–80,
Athens, Greece, 31 March 2009.

and more straightforward access to linguistic resources, (b) encourage the systematic generation of linguistic data beyond what emerges from fieldwork and other descriptive studies and (c) advocate the generation of a multi-lingual data pool for linguistic research.

2 Annotation tools in linguistic research

It is well known that the generation of natural language examples enriched by linguistic information in the form of symbols is a time consuming enterprise quite independent of the form that the raw material has and the tools that were chosen. Equally well known are problems connected to the generation and storage of linguistic data in the form of standard document files or spread sheets (Bird and Simons 2003). Although it is generally agreed on that linguistic resources must be kept in a sustainable and portable format, it remains less clear, how a tool should look that would help the linguist to accomplish these goals. For the individual researcher it is not easy to decide which of the available tools serve his purpose best. To start with it is often not clear which direction research will take, which categories of data are needed and in which form the material should be organized and stored. But perhaps even more importantly most tools turn out to be so complex that the goal of mastering them becomes an issue in its own right. Researchers that work together with communities that speak an endangered or lesser documented language experience that digital tools used for language documentation can be technically too demanding.Training periods for annotators become necessary together with technical help and maintenance by experts which not necessarily are linguists themselves. In this way tool management develops into an issue in itself taking away resources from the original task at hand - the linguistic analysis.Linguists too often experience that some unlucky decision concerning technical tools gets data locked in systems which cannot be accessed anymore after a project, and the technical support coming along with it,has run out of funding.

2.1 TypeCraft an overview

In the following we would like to introduce a linguistic tool for text annotation called TypeCraft, which we have created through combining several well-understood tools of knowledge management. Needless to say, TypeCraft will not solve all the problems mentioned above, yet it has some new features that make data annotation an easier task while adding discernibility and general efficiency. That one can import example sentences directly into research papers is one of these features. In addition TypeCraft is a collaboration and knowledge sharing tool, and, combined with database functionality, it offers some of the most important functions we expect to see in a digital language documentation tool.

In the following we will address glossing and illustrate present day glossing standards with examples from Akan, a Kwa language spoken in Ghana, to then turn to a more detailed description of TypeCraft. However, a brief overview over the main features of TypeCraft seems in order at this point.

TypeCraft is a relational database for natural language text combined with a tabular text editor for interlinearized glossing, wrapped into a wiki which is used as a collaborative tool and for online publication. The system, which has at present 50 users and a repository of approximately 4000 annotated phrases, is still young. Table 1 gives a first overview of TypeCraft's main functionalities.

3 Glossing

The use of glosses in the representation of primary data became a standard for linguistic publications as late as in the 1980s (Lehmann, 2004) where interlinear glosses for sample sentences started to be required for all language examples except those coming from English. However, the use of glossed examples in written research was, and still is, not accompanied by a common understanding of its function, neither concerning its role in research papers nor its role in research itself. It seems that glosses, when occurring in publications, are mostly seen as a convenience to the reader. Quite commonly information essential to the understanding of examples is given in surrounding prose, and often without any appropriate reflection in the glosses themselves.

Let us look at a couple of examples with interlinear glosses taken at random from the list of texts containing Akan examples. These examples are taken from the online database Odin at Fresno State University. The Odin database (http:/www.csufresno.edu/odin/) is a repository of interlinear glossed texts which have been extracted mainly from linguistic papers. The database it-

Annotation	Collaboration	Data Migration
tabular interface for word level glossing - automatic sentence break-up	individual work spaces for users that would like to keep data private	manual import of text and individual sentence
drop down reference list of linguistic symbols	data sharing for predefined groups such a research collaborations	export of annotated sentence tokens (individual tokens or sets) to Microsoft Word, Open Office and LaTEX
word and morpheme deletion and insertion	data export from the TypeCraft database to the TypeCraft wiki	export of XML (embedded DTD) for further processing of data
lazy annotation mode (sentence parsing)	access to tag sets and help pages from the TypeCraft wiki	
customized sets of sentence level tags for the annotation of construction level properties	access to information laid out by other annotators or projects.	

Table 1: Overview over TypeCraft Functionalities

self consists of a list of URLs ordered by language leading the user to the texts of interest.

3.1 The glossing of Akan - an example

Akan is one of the Kwa languages spoken in Ghana. The first example from the Odin database, here given as (1), comes from a paper by (Haspelmath, 2001)

(1) Ámá màà mè sìká.
Ama give 1SG money
'Ama gave me money. '

The second example is extracted from a paper by (Ameka, 2001):

(2) Ámá dè sìká nó máá mè.
Ama take money the give 1SG
'Ama gave me the money'

(Lit: 'Ame took money gave me')

The third example is quoted in a manuscript by (Wunderlich, 2003):

(3) ɔ-fɛmm me ne pɔflnkono.
3sg-lent 1sg 3sgP horse that
'He lent me a horse'

and the forth one comes from a manuscript by (Drubig, 2000) who writes about focus constructions:

(4) Hena na Ama rehwehwɛ?
who FOC Ama is-looking-for?
'Who is it that Ama is looking for?'

Except for Ameka, the authors quote Akan examples which are excerpted from the linguistic literature. Often examples coming from African languages have a long citation history and their validation is in most cases nearly impossible. When we compare (1) – (4) we notice a certain inconsistency for the annotation of *nó* which is glossed as 'the' (1), 'that' (3) and as DEF (2) respectively. This difference could indicate that Akan does not make a lexical distinction between definiteness and deixis, most likely however we simply observe a 'glossing figment'. The general lack of part of speech information in all examples easily leads us astray; should we for example assume that na in example (4) is a relative pronoun? The general lack of proper word level glossing makes the data for other linguists quite useless, in particular if they are not themselves native speakers or experts in exactly this language.*Màà* is a past form, but that tense marking is derived by suffixation is only indicated in (2) via a hyphen between the translational gloss and the PAST tag. Likewise *rehwehwɛ*in(4)is a progressive form, yet the lack of morpheme boundaries, and consistent annotation prevents that these and similarly glossed serve as a general linguistic resource. Purely translational glosses might be adequate for text strings which serve as mere illustrations; however, for linguistic data,that is those examples that are (a) either crucial for the evaluation of the theoretical developxment reported on, or (b) portray linguistic pattern of general interest, to provide morpho-

syntactic and morpho-functional as well as part of speech information would seem best practice.

It seems that linguists underestimate the role that glossing, if done properly,could play as part of linguist research. Symbolic rewriting and formal-grammar development are two distinct modes of linguistic research. Yet there is no verdict that forces us to express descriptive generalizations exclusively by evoking a formal apparatus of considerable depth. Instead given simplicity and parsimony of expression it might well be that symbolic rewriting serves better for some research purposes than theoretical modeling. One can not replace one by the other. Yet which form of linguistic rendering is the best in a given situation should be a matter of methodological choice. Essential is that we realize that we have a choice. Sizing the opportunity that lies in the application of symbolic rewriting, of which interlinear glossing is one form, could make us realize that the generation of true linguistic resources is not exclusively a matter best left to computational linguists.

4 A short description of TypeCraft

Typecraft is an interlinear 'glosser' designed for the annotation of natural language phrases and small corpora. The TypeCraft wiki serves as an access point to the TypeCraft database. We use standard wiki functionality to direct the TypeCraft user from the index page of the TypeCraft wiki to the TC interface of the database, called My Texts. My Texts is illustrated in Figure 1.The interface is taken from a user that not only possesses private data (Own texts), but who also shares data with other users (Shared Texts). At present sharing of text is a feature set by the database administrator, but in the near future the user will be able to choose from the TypeCraft user list the people with whom he wants to share his data. Note that data is stored as texts which consist of annotated tokens, standardly sentences. 'Text' in TypeCraft does not necessarily entail coherent text, but may also refer to any collection of individual tokens that the user has grouped together. A TypeCraft user can publish his data online; yet his own texts are by default 'private', that is, only he as the owner of the material can see the data and change it. To share data within the system or online is a function that can be selected by the user.

Different from Toolbox, which is a linguistic data management system known to many Africanists, TypeCraft is a relational database and therefore by nature has many advantages over file based systems like Toolbox. This concerns both, data integrity and data migration. In addition databases in general offer a greater flexibility for data search.For example, it is not only possible to extract all serial verb constructions for all (or some) languages known to TypeCraft, it is also possible to use the gloss index to find all serial verb constructions where a verb receives a marking specific to the second verb in an SVC. The other mayor difference between Toolbox and TypeCraft is that TypeCraft is an online system which brings many advantages, but also some disadvantages. An online database is a multi-user system, that is, many people can access the same data at the same time independent of were they physically are. Distributive applications are efficient tools for international research collaboration. TypeCraft is designed to allow data sharing and collaboration during the process of annotation. Yet although there are many advantages to an online tool, to be only online is at the same time a major disadvantage. Not all linguists work with a stable internet connection, and in particular for work in the field TypeCraft is not suitable.

TypeCraft uses Unicode, so that every script that the user can produce on his or her PC can be entered into the browser,[1] which for TypeCraft must be Mozilla Firefox. Different from Toolbox TypeCraft insists on a set of linguistic glosses, reflecting standards advocated for example by the Leipzig Convention distributed by the Max Planck Institute for Evaluationary Anthropology or an initiative such a GOLD (Farrar and Lewis, 2005).Yet, TypeCraft still allows a user-driven flexibility when it comes to the extension of the tag-set, as explained in the next section.

5 Glossing with TypeCraft

TypeCraft supports word-to-word glossing on eight tiers as shown in Figure 2. After having imported a text and run it through the sentence splitter, a process that we will not describe here, the user can select via mouse click one of the phrases and enter the annotation mode. The system prompts the user for the Lazy Annotation Mode (in Toolbox called sentence parsing) which will automatically insert (on a first choice ba-

[1] Note however that self-defined characters or characters that are not Unicode will also cause problems in TypeCraft

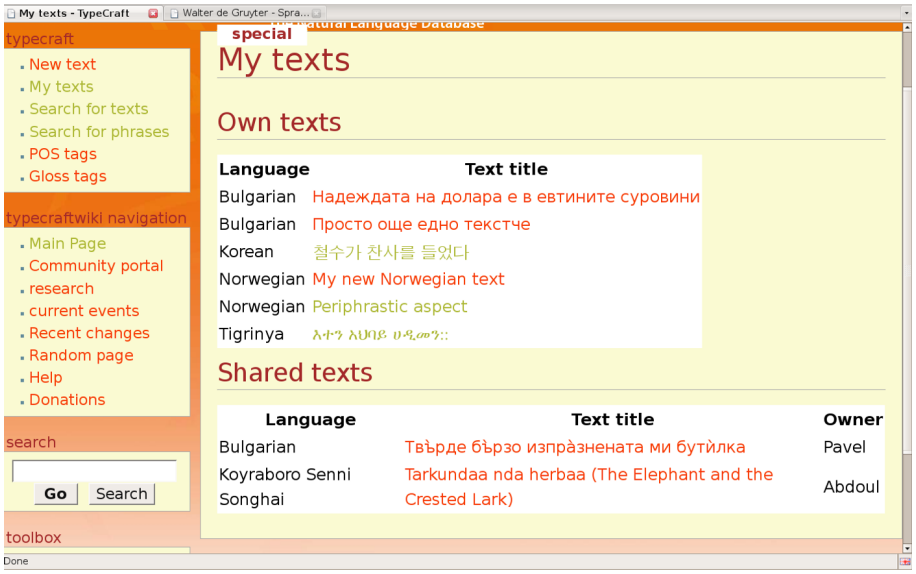


Figure 1: My texts in TypeCraft

sis) the annotation of already known words into the annotation table. TypeCraft distinguishes between translational, functional and part-of-speech glosses. They are visible to the annotator as distinct tiers called Meaning, Gloss and POS. Every TypeCraft phrase, which can be either a linguistic phrase or a sentence, is accompanied by a free translation. In addition the specification of construction parameters is possible. Although the user is restricted to a set of pre-defined tags, the TypeCraft glossery is negogiable. User discussion on the TCwiki, for example in the context of project work, or by individual users, has led to an extension of the TypeCraft tag set. Although TypeCraft endorses standardization, the system is user-driven. Glosses are often rooted in traditional grammatical terminology, which we would like to set in relation to modern linguistic terminology. The TCwiki is an adaquate forum to discuss these traditions and to arrive at a annotation standard which is supported by the users of the system. Under annotation the user has access a dropdown menu, showing standard annotation symbols. These symbols together with short explanations can also be accessed from the TypeCraft wiki so that they can be kept open in tabs during annotation. In Figure 2 we also see the effect of 'mousing over' symbols, which displays their 'long-names'. Some symbols have been ordered in classes. In Figure 2 we see for example that the feature past is a subtype of the feature Tense. This classification will in the future also inform search. Further features of the annotation interface that we cannot describe here are the easy representation of non-Latin scripts, deletion and insertion of words and morphemes during annotation, the accessibility of several phrases under annotation and the grouping of tokens into texts.

6 Data Migration

Export of data to the main text editors is one of the central functions of TypeCraft. TC tokens can be exported to Microsoft Word, OpenOffice.org Writer and LaTeX. This way the user can store his data in a database, and when the need arises, he can integrate it into his research papers. Although annotating in TypeCraft is time consuming, even in Lazy Annotation Mode, the resusablity of data stored in TypeCraft will on the long run pay off. Export can be selected from the text editing window or from the SEARCH interface.After import the examples can still be edited in case small ajustments are necessery.Example (5) is an example exported from TypeCraft.

(5)

Omu nju hakataahamu abagyenyi			
òmù	njù	hàkàtààhàmù	àbàgyéngyì
Omu	n ju	ha ka taah a mu	a ba gyenyi
in	CL9 *house*	CL16 PST *enter* IND LOC	IV CL2 *visitor*
PREP	N	V	N

'In the house entered visitors'

(5) illustrates locative inversion in Runyakitara, a Bantu language spoken in Uganda. The translational and functional glosses, which belong to two distinct tiers in the TypeCraft annotation interface, appear as one line when imported to one of the word processing programs supported by Type-

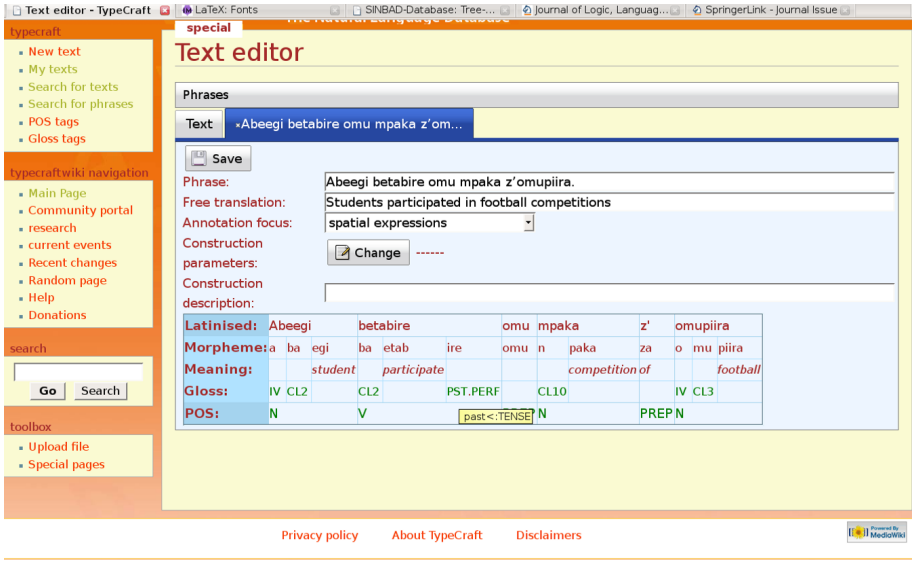


Figure 2: Glossing in TypeCrat

Craft. Although glossing on several tiers is conceptually more appropriate, linguistic publications require a more condensed format. As for now we have decided on an export which displays 6 tiers. Next to export to the main editors, TypeCraft allows XML export which allows the exchange of data with other applications. Figure 3 gives an overview over the top 15 languages in TypeCraft. In January 2009 Lule Sami with 2497 phrases and Runyakitara (Runyankore Rukiga)with 439 phrases were the top two languages. At present the database contains approximately 4000 from 30 languages.Most of the smaller language (with 300 t0 40 sentences) are African languages.

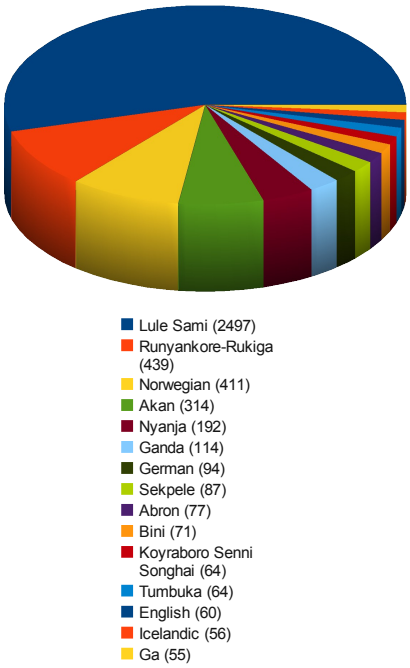


Figure 3: Top 15 TypeCraft languages by number of phrases

7 Conclusion

In this paper we suggest that the general availability of richly annotated, multi-lingual data directly suited for scientific publication could have a positive impact on the way we think about language, and how we approach linguistics. We stress the opportunity that lies in the application of symbolic rewriting, of which interlinear glossing is one form, and encourage the systematic generation of linguistic data beyond what emerges from fieldwork and other descriptive studies. With TypeCraft we introduce an online glossing tool for textual data which has two main goals (a) to allow linguists to gloss their data without having to learn how to install software and without having to undergo a long training period before the can use the tool and (b) to make linguistically annotated data available to a bigger research community. We hope that the use of this tool will add to the standardization of language annotation. We further hope that TypeCraft will be used as a forum for linguistic projects that draw attention to the lesser-studied languages of the World.

References

Felix K. Ameka. 2001. Multiverb constructions in a west african areal typological perspective. In Dorothee Beermann and Lars Hellan, editors, *Online Proceedings of TROSS – Trondheim Summer School 2001*.

Hans Bernhard Drubig. 2000. Towards a typology of focus and focus constructions. In *Manuscript, University of Tübingen, Germany*.

Scott Farrar and William D. Lewis. 2005. The gold community of practice: An infrastructure for linguistic data on the web. In *In Proceedings of the EMELD 2005 Workshop on Digital Language Documentation: Linguistic Ontologies and Data Categories for Language Resources*.

Martin Haspelmath. 2001. Explaining the ditransitive person-role constraint: A usage-based approach. In *Manuscript Max-Planck-Institut für evolutionäre Anthropologie*.

William Labov. 1987. Some observations on the foundation of linguistics. In *Unpublished manuscript, University of Pennsylvania, USA*.

Christian Lehmann, 2004. *Morphologie: Ein Internationales Handbuch zur Flexion und Wortbildung*, chapter Interlinear morphological glossing. DeGryter Berlin-New York.

Dieter Wunderlich. 2003. Was geschieht mit dem dritten argument? In *Manuscript University of Düsseldorf, Germany*.

Towards an electronic dictionary of Tamajaq language in Niger

Chantal Enguehard
LINA - UMR CNRS 6241
2, rue de la Houssinière
BP 92208
44322 Nantes Cedex 03
France
chantal.enguehard@univ-nantes.fr

Issouf Modi
Ministère de l'Education Nationale
Direction des Enseignements du Cycle de Base1
Section Tamajaq.
Republique du Niger
modyissouf@yahoo.fr

Abstract

We present the Tamajaq language and the dictionary we used as main linguistic resource in the two first parts. The third part details the complex morphology of this language. In the part 4 we describe the conversion of the dictionary into electronic form, the inflectional rules we wrote and their implementation in the Nooj software. Finally we present a plan for our future work.

1. The Tamajaq language

1.1 Socio-linguistic situation

In Niger, the official language is French and there are eleven national languages. Five are taught in a experimental schools: Fulfulde, Hausa, Kanuri, Tamajaq and Soŋay-Zarma.
According to the last census in 1998, the Tamajaq language is spoken by 8,4% of the 13.5 million people who live in Niger. This language is also spoken in Mali, Burkina-Faso, Algeria and Libya. It is estimated there are around 5 millions Tamajaq-speakers around the world.
The Tamacheq language belongs to the group of Berber languages.

1.2 Tamajaq alphabet

The Tamajaq alphabet used in Niger (Republic of Niger, 1999) uses 41 characters, 14 with diacritical marks that all figure in the Unicode standard (See appendix A). There are 12 vowels: a, â, ă, ə, e, ê, i, î, o, ô, u, û.

1.3 Articulary phonetics

Consonants		Voiceless	Voiced
Bilabial	Plosive		b
	Nasal		m
	Trill		r
	Semivowel		w
Labiodental	Fricative	f	
Dental	Plosive	t	d
	Fricative	s	z
	Nasal		n
	Lateral		l
Pharyngeal	Plosive	ṭ	ḍ
	Fricative	ṣ	ẓ
	Lateral		ḷ
Palatal	Plosive	c	ǰ
	Fricative	š	j
	Semivowel		y
Velar	Plosive	k	g, ǧ
	Fricative	ɣ	x
	Nasal		ŋ
Glottal	Plosive	q	
	Fricative	h	

Table 1a: Articulary phonetics of Tamajaq consonants

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 81–88,
Athens, Greece, 31 March 2009.

Vowels	Close	Close-mid	Open-mid	Open
Palatal	i	e		
Central		ə	ă	a
Labial	u	o		

Table 1b: Articulary phonetics of Tamajaq vowels

1.4 Tools on computers

There are no specific TALN tools for the Tamajaq language.
However characters can be easily typed on French keyboards thanks to the AFRO keyboard layout (Enguehard and al. 2008).

2 Lexicographic resources

We use the school editorial dictionary "dictionnaire Tamajaq-français destiné à l'enseignement du cycle de base 1". It was written by the SOUTEBA[1] project of the DED[2] organisation in 2006. Because it targets children, this dictionary consists only of 5,390 entries. Words have been chosen by compiling school books.

2.1 Structure of an entry

Each entry generally details :
- lemma,
- lexical category,
- translation in French,
- an example,
- gender (for nouns),
- plural form (for nouns).

Examples:

« ăbada$_1$: sn. bas ventre. Daw tədist. Bărar wa yəllûẓăn ad t-yəltəɣ ăbada-net. təmust.: yy. igət: ibadan. »

« ăbada2: sn. flanc. Tasăga meɣ daw ădăg əyyăn. Iməwwəẓla əklăn dăɣ ăbada n əkašwar. Anammelu.: azador. təmust.: yy. ∃səfsəs.: ă. Igət: ibadan. »

Homonyms are described in different entries and followed by a number, as in the above example.

2.2 Lexical categories

The linguistic terms used in the dictionary are written in the Tamajaq language using the abbreviations presented in table 2. In addition, this table gives information about the number of entries of each lexical category.

Lexical category		**Abbreviation**	**Number of entries**
Tamajaq	**English**		
əḍəkuḍ	number	ḍkḍ.	3
ənalkam	deteminant	nlkm.	1
anamal	verb	nml.	1450
samal	adjective	sml.	48
əsəmmadăɣ ən təla	possessive pronoun	smmdɣtl.	5
isən	noun	sn.	3648
isən n ənamal	Verbal noun	snnml.	33
isən an təɣərit	name of shout	sntɣrt.	2
isən xalalan	proper noun	snxln.	29
isən iẓẓəwen	complex noun	snẓwn.	137
əstakar	adverb	stkr.	8
əsatkar n ădag	adverb of location	stkrdg.	10
əṣatkar n igət	Adverb of quantity	stkrgt.	1
təɣərit	ono-matopoeia	tɣrt.	8
tənalkamt	particle	tnlkmt.	2

Table 2: Tamajaq lexical categories

3 Morphology

The Tamajaq language presents a rich morphology (Aghali-Zakara, 1996).

3.1 Verbal morphology

Verbs are classified according to the number of consonants of their lexical root and then in different types. There are monoliteral, biliteral triliteral, quadriliteral verbs...
Three moods are distinguished: imperative, simple injunctive and intense injunctive.
Three aspects present different possible values:
- accomplished: intense or negative;

[1]Soutien à l'éducation de base.
[2]DED: Deutscher Entwicklungsdienst.

- non accomplished: simple, intense or negative;
- aorist future: simple or negative.

Examples :

əktəb (to write): triliteral verb, type 1.

əṣṣən (to know): triliteral verb, type 2 (ṣṣn).

əməl (to say): biliteral verb, type 1

akər (to steal): biliteral verb, type 2

awəy (to carry): biliteral verb, type 3

ăšwu (to drink): biliteral verb, type 4

ăru (to love): monoliteral verb, type 2

ăru (to open): monoliteral verb, type 3

Each class of verb has its own rules of conjugation.

3.2 Nominal morphology

a. Simple nouns
Nouns present three characteristics:
- gender: masculine or feminine;
- number: singular or plural;
- annexation state is marked by the change of the first vowel.

Terminology		Abbreviation
təmust	gender	tmt.
yey	masculine	yy.
tənte	feminine	tnt.
awdəkki	singular	wdk.
iget	plural	gt.
əsəfsəs	annexation state	sfss.

Table 3: Tamajaq terminology for nouns

Example :

« aṭrəkka: sn. morceau de sucre. Akku: ablәy n°2. təmust.: yy. Ǝsəfsəs.: ə. Igət: əṭrəkkatăn. »

"aṭrəkka" is a masculine noun. Its plural is "əṭrəkkatăn". It becomes "əṭrəkka" when annexation state is expressed.

The plural form of nouns is not regular and has to be specifically listed.

b. Complex nouns
Complex nouns are composed by several lexical units connected together by hyphens. It could include nouns, determiners or prepositions as well as verbs.

Examples:

Noun +determiner + noun

"ejăḍ-n-əjḍăn", literally means "donkey of birds" (this is the name of a bird).

Verb + noun

"awəy-əhuḍ" literally means "it follows harmattan" (kite).

"gaẓẓăy-təfuk" literally means "it looks at sun" (sunflower).

Preposition + noun

"In-tamaṭ" means "the one of the tree acacia" (of acacia).

Verb + verb

"azəl-azəl" means "run run" (return).

We counted 238 complex nouns in the studied dictionary.

4 Natural Language Processing of Tamajaq

4.1 Nooj software (Silberztein, 2007)

« Nooj is a linguistic development environment that includes tools to create and maintain large-coverage lexical resources, as well as morphological and syntactic grammars. » This software is specifically designed for linguists who can use it to test hypothesis on real corpus. « Dictionaries and grammars are applied to texts in order to locate morphological, lexical and syntactic patterns and tag simple and compound words. » Nooj put all possible tags for each token or group of tokens but does not disambiguate between the multiple possibilities. However, the user can build his own grammar to choose between the multiple possible tags. The analysis can be displayed as a syntactic tree.
This software is supported by Windows.
We chose to construct resources for this software because it is fully compatible with Unicode.

4.2 Construction of the dictionary

We convert the edited dictionary for the Nooj software.
3,463 simple nouns, 128 complex nouns, 46 adjectives and 33 verbo-nouns are given with their plural form. Annexation state is indicated for 987

nouns, 23 complex nouns, 2 adjectives and 7 verbo-nouns.

We created morphological rules that we expressed as Perl regular expressions and also in the Nooj format (with the associated tag).

a. Annexation state rules

Thirteen morphological rules calculate the annexation state.

Examples:

The 'A1ă' rule replaces the first letter of the word by 'ă'.

'A1ă' rule	
Nooj	<LW><S>ă/sfss
Perl	^.(.*)$ → ă$1

Table 4: Rule 'A1ă'

The 'A2ə' rule replaces the second letter of the word by 'ə'.

'A2ə' rule	
Nooj	A2ə=<LW><R><S>ə/sfss
Perl	^(.).(.*)$ → $1ə$2

Table 5: Rule 'A2ə'

b. Plural form rules

We searched formal rules to unify the calculation of plural forms. We found 126 rules that fit from 2 up to 446 words. 2932 words could be associated with, at least, one flexional rule.

Examples:

'I4' rule deletes the last letter, adds "-ăn" at the end and "i-" at the beginning.	
Nooj	I4=ăn<LW><S>i/Iget
Perl	^(.*).$ → i$1ăn
#	446 words

Table 6: Rule 'I4'

'I2' rule deletes the last and the second letters and includes "-en" at the end and "-i-" in the second position.	
Nooj	I2=<B>en<LW><R><S>i/Iget
Perl	^(.).(.*).$ → $1i$2en
#	144 words

Table 7: Rule 'I2'

'I45' rule deletes the final letter and include "-en" at the end.	
Nooj	I45=<B>en/Iget
Perl	^(.*).$ → <B>en/Iget
#	78 words

Table 8: Rule 'I45'

'I102' rule deletes the two last letters and the second one and includes a final "-a" and a "-i-" in the second position.	
Nooj	I102=<B2>a<LW><R><S>i/Iget
Perl	^(.).(.*)..$ → $1i$2a
#	6 words

Table 9: Rule 'I102'

c. Combined rules

When it was necessary, the above rules have been combined to calculate singular and plural forms with or without annexation state.
We thus finally obtained 319 rules.

Example:
I2RA2ă =
:Rwdk + :I2 + :Rwdk :A2ă + :I2 :A2ă

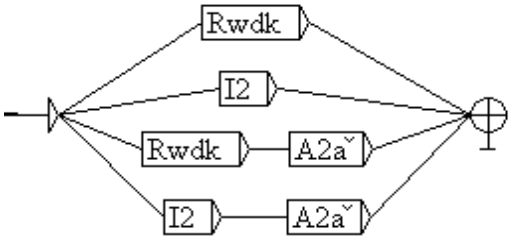


Fig. 1: Rule I2RA2ă

This rule recognizes the singular form (:Rwdk), the plural form (:I2), the singular form with the annexation state (:Rwdk :A2ă) and the plural form with the annexation state (:I2 :A2ă).

25 words meet this rule.

For instance, "taḍləmt" (accusation, provocation), is inflected in:

- taḍləmt,taḍləmt,SN+tnt+wdk
- tiḍləmen,taḍləmt,SN+tnt+Iget
- tăḍləmen,taḍləmt,SN+tnt+Iget+sfss

- tăḍləmt,taḍləmt,SN+tnt+wdk+sfss

d. Conjugaison rules

Verb classes are not indicated in the dictionary. We only describe a few conjugaison rules, just to check the expressivity of the Nooj software

Here is the rule of the verb "əṣṣən" (to know), intense accomplished aspect, represented as a transducer.

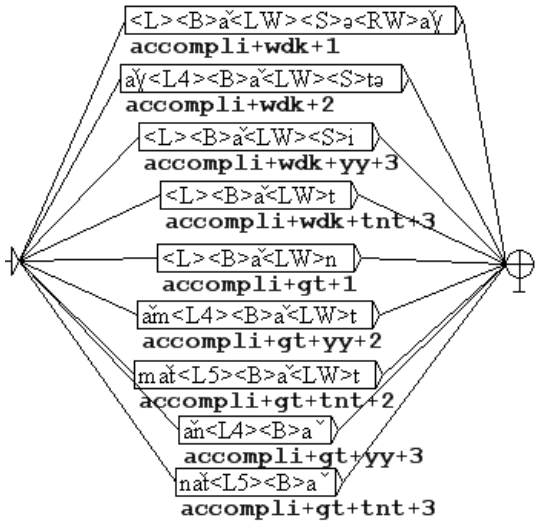


Fig. 2: Verb "əṣṣən", intense accomplished aspect

We obtain, in the inflected dictionary, the correct conjugated forms.

əṣṣănăɣ+əṣṣən,V+accompli+wdk+1

təṣṣănăɣ+əṣṣən,V+accompli+wdk+2

iṣṣăn+əṣṣən,V+accompli+wdk+yy+3

təṣṣăn+əṣṣən,V+accompli+wdk+tnt+3

nəṣṣăn+əṣṣən,V+accompli+gt+1

təṣṣănăm+əṣṣən,V+accompli+gt+yy+2

təṣṣănmăt+əṣṣən,V+accompli+gt+tnt+2

əṣṣănăn+əṣṣən,V+accompli+gt+yy+3

əṣṣănnăt+əṣṣən,V+accompli+gt+tnt+3

e. Irregular words

Finally, the singular and plural forms of 2,457 words were explicitly written in the Nooj dictionary because they do not follow any regular rule.

Examples:

Singular	**Plural**	**Translation**
ag-awnaf	kel-awnaf	tourist
amanẓo	imenẓa	young animal
ănaffareššĭ	inəffərəšša	someboby with bad mood
ănesbehu	inəsbuha	liar
efange	ifangăyan	bank
efajanfăj	ifajanfăɣăn	sling
emagărmăz	imagămăzăn	plant
emazzăle	imazzaletăn	singer
taḍaggalt	tiḍulen	daugther-in-law
tejăṭ	tizḍen	goal (football)

Table 10: Examples of irregular plural forms

f. Result

There are 6,378 entries in the Nooj dictionary. The inflected dictionary, calculated from the above dictionary and with the inflectional and conjugation rules, encounters 11,223 entries.

Nooj is able to use the electronic dictionary we've created to automatically tag a text (see an example in appendix B).

4.3 Future work

a Conversion into XML format

We will convert the inflectional dictionary into the international standard Lexical Markup Framework format (Francopoulo and al., 2006) in order to make it easily usable by other TALN application,.

b Automatic search of rules

Due to the high morphological complexity of the Tamajaq language, we plan to develop a Perl program that would automatically determine the derivational and conjugation rules.

c Completion and correction of the resource

The linguistic resource will be completed during the next months in order to add the class of verbs that are absent for the moment, and also to correct the errors that we noticed during this study.

d Enrichment of the resource

We plan to construct a corpus of school texts to evaluate the out-of-vocabulary rate of this dictionary. This corpus could then be used to enrich the dictionary. The information given by Nooj would be useful to choose the words to add.

Acknowledgement

Special thanks to John Johnson, reviewer of this text.

References

Aghali-Zakara M. 1996. *Éléments de morphosyntaxe touarègue*. Paris : CRB-GETIC, 112 p.

Enguehard C. and Naroua H. 2008. *Evaluation of Virtual Keyboards for West-African Languages*. Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), Marrakech, Morocco.

Francopoulo G., George M., Calzolari N., Monachini M., Bel N., Pet M., Soria C. 2006 *Lexical Markup Framework (LMF)*. LREC, Genoa, Italy.

République of Niger. 19 octobre 1999. *Arrêté 214-99* de la République du Niger.

Max Silberztein. 2007. *An Alternative Approach to Tagging*. NLDB 2007: 1-11

APPENDIX A : Tamajaq official alphabet (République of Niger, 1999)

Character	Code	Character	Code
a	U+0061	A	U+0041
â	U+00E1	Â	U+00C2
ă	U+0103	Ă	U+0102
ə	U+01DD	Ǝ	U+018E
b	U+0062	B	U+0042
c	U+0063	C	U+0043
d	U+0064	D	U+0044
ḍ	U+1E0D	Ḍ	U+1E0C
e	U+0065	E	U+0045
ê	U+00EA	Ê	U+00CA
f	U+0066	F	U+0046
g	U+0067	G	U+0047
ǧ	U+01E7	Ǧ	U+01E6
h	U+0068	H	U+0048
i	U+0069	I	U+0049
î	U+00EE	Î	U+00CE
j	U+006A	J	U+004A
ǰ	U+01F0	J̌	U+004AU+030C
ɣ	U+0263	Ɣ	U+0194
k	U+006B	K	U+004B
l	U+006C	L	U+004C
ḷ	U+1E37	Ḷ	U+1E36
m	U+006D	M	U+004D
n	U+006E	N	U+004E
ŋ	U+014B	Ŋ	U+014A
o	U+006F	O	U+004F
ô	U+00F4	Ô	U+00D4
q	U+0071	Q	U+0051
r	U+0072	R	U+0052
s	U+0073	S	U+0053
ṣ	U+1E63	Ṣ	U+1E62
š	U+0161	Š	U+0160
t	U+0074	T	U+0054
ṭ	U+1E6D	Ṭ	U+1E6C
u	U+0075	U	U+0055
û	U+00FB	Û	U+00DB
w	U+0077	W	U+0057
x	U+0078	X	U+0058
y	U+0079	Y	U+0059
z	U+007A	Z	U+005A
ẓ	U+1E93	Ẓ	U+1E92

APPENDIX B : Nooj tagging Tamajaq text

Nooj perfectly recognizes the four forms of the word "awăqqas" (big cat) in the text:

"awăqqas, iwaɣsan, awaɣsan"

These forms are listed in the inflectional dictionary as:

awăqqas,awăqqas,SN+yy+wdk

awăqqas,awăqqas,SN+yy+wdk+FLX=A1a+sfss

iwaɣsan,awăqqas,SN+yy+iget

awaɣsan,awăqqas,SN+yy+iget+FLX=A1a+sfss

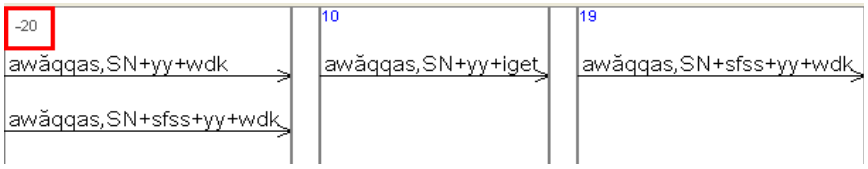


Fig.3: Tags on the text "awăqqas, iwaɣsan, awaɣsan"

On the figure 3, we can see that the first token "awăqqas" gets two tags:
- "awăqqas,SN+yy+wdk" (singular)
- "awăqqas,SN+yy+wdk+sfss" (singular and annexation state).

The second and third tokens get a unique tag because there is no ambiguity.

A repository of free lexical resources for African languages: the project and the method

Piotr Bański
Institute of English Studies
University of Warsaw
Warsaw, Poland
bansp@o2.pl

Beata Wójtowicz
Institute of Oriental Studies
University of Warsaw
Warsaw, Poland
b.wojtowicz@uw.edu.pl

Abstract

We report on a project which we believe to have the potential to become home to, among others, bilingual dictionaries for African languages. Kept in a well-structured XML format with several possible degrees of conformance, the dictionaries will be able to get usable even in their early versions, which will be then subject to supervised improvement as user feedback accumulates. The project is FreeDict, part of SourceForge, a well-known Internet repository of open source content.

We demonstrate a possible process of dictionary development on the example of one of FreeDict dictionaries, a Swahili-English dictionary that we maintain and have been developing through subsequent stages of increasing complexity and machine-processability. The aim of the paper is to show that even a small bilingual lexical resource can be submitted to this project and gradually developed into a machine-processable form that can then interact with other FreeDict resources. We also present the immediate benefits of locating bilingual African dictionaries in this project.

We have found FreeDict to be a very promising project with a lot of potential, and the present paper is meant to spread the news about it, in the hope to create an active community of linguists and lexicographers of various backgrounds, where common research subprojects can be fruitfully carried out.

1 Introduction

The FreeDict project was started by Horst Eyermann in 2000 and initially hosted bilingual dictionaries produced by concatenating (crossing) the contents of the dictionaries in the Ergane project (http://download.travlang.com/Ergane/), with Esperanto as the interlanguage. At first, the data was kept in the DICT format (Faith and Martin, 1997).

DICT (Dictionary Server Protocol) is by now a well-established TCP-based query/response protocol that allows a client to access definitions from a set of various dictionary databases. It provides data in textual form, but it also has the potential of providing MIME-encoded content. The clients can be free-standing desktop applications or they can be integrated into editors or web browsers. DICT web gateways also exist (see e.g. http://dict.org/).

The DICT format is a plain text format with an accompanying index file. The FreeDict-DICT interface initially used so-called "c5 files".[1] A c5 example of an entry from version 0.0.1 of Swahili-English dictionary is presented below.

```
abiria
     passenger(s)
```

Later on, the project adopted the TEI P4 standard (Sperberg-McQueen and Burnard, 2002) as its primary format, and with the help of its second administrator, Michael Bunk, created tools for conversion from the TEI into a variety of other dictionary platforms. A simple dictionary editor was also created. The change of the primary format was a very fortunate move, thanks to which we can today recommend the project as the possible home for free African language dictionaries big and small.

We are going to base our discussion on the Swahili-English dictionary, the first FreeDict dictionary encoded according to the guidelines of TEI P5 XML standard (TEI Consortium, 2007). The dictionary in its current form is an offshoot of a different project of ours that we decided to make available on a free license and in version 0.4 contains over 2600 headwords. We use it to demonstrate a possible process of dictionary de-

[1] C5 is the format used, among others, by the CIA World Factbook, where the heading is at the left edge and the contents are indented by 5 spaces.

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 89–95,
Athens, Greece, 31 March 2009.

velopment, from the simplest to the advanced, machine-processable form.

2 From glossaries to rich lexical databases: the possible shapes of FreeDict dictionaries

A dictionary can begin its life at FreeDict as a simple glossary, with the simplest format possible, as shown in the made-up entry below:

```
<entry>
   <form><orth>alasiri</orth></form>
   <def>afternoon</def>
</entry>
```

The next example entry comes from Swahili-English xFried Freedict Dictionary, version 0.0.2, compiled by Beata Wójtowicz. That dictionary contained around 1500 entries of varied quality. It was based on a dictionary extracted from Morris P. Fried's *Swahili-Kiswahili to English Translation Program*, to which selected entries from the first FreeDict Swahili-English dictionary (compiled by Horst Eyermann) were added. That version also introduced information on parts of speech.

```
<entry>
   <form><orth>alasiri</orth></form>
   <def>   afternoon</def>
   <gramGrp> <pos>n</pos> </gramGrp>
</entry>
```

On the way to version 0.3, the entry looked as follows:

```
<entry xml:id="alasiri">
  <form><orth>alasiri</orth></form>
  <gramGrp><pos>n</pos></gramGrp>
  <sense>
    <def>afternoon (period between 3
     p.m. and 5 p.m.)</def>
  </sense>
</entry>
```

All the bracketed information was then turned into separate `<note/>` elements, in order to make the translation equivalents easily processable (see Prinsloo and de Schryver, 2002, for remarks on processability of translation equivalents). The change was performed by regex search-and-replace, roughly from `\((.+)\)</def>` into `</def><note type="hint">$1</note>`[2], with a subsequent review of all the new `<note/>` elements extracted by an XPath query.

Depending on the regularity of expressions in brackets, some additional words would be inserted into the search string, to be converted into `<note/>` elements of the appropriate type. Initially, only `@type="hint"` was used, as the most generic. At the moment, there are several more specialized type values, including `@type="editor"`, which contains editorial remarks that will not be shown to the user but will remain in the source. `@type`-less `<note/>` elements are used for quick localized communication between editors and are discarded by XSLT scripts when preparing the source version for release (they are also clearly marked by the CSS stylesheet that accompanies the dictionary, so that the editors can easily spot each note when reviewing the dictionary in a browser). FreeDict advocates the use of some other types of notes: recording the last editor of the entry, the date of the latest modification, and the degree of certainty, valuable in this kind of projects (where, e.g., some automated changes would set the certainty level to "low" and as such requiring editorial approval).

In the current version, 0.4, the `<sense/>` element looks as follows.

```
<sense>
   <def>afternoon</def>
   <note type="def">period between 3
      p.m. and 5 p.m.</note>
</sense>
```

This is what we decided to keep in version 0.4, exactly for the purpose of illustrating the possible development stages of dictionaries. In the next version, the `<sense/>` element will eventually attain the form currently (i.e., after September 2007) recommended by the TEI Guidelines for translation equivalents:

```
<sense>
   <cit type="trans">
       <quote>afternoon</quote>
       <def>period between 3 p.m. and
       5 p.m.</def>
   </cit>
</sense>
```

The `<quote/>` element holds the translation equivalent that can be an anchor for dictionary

[2] This is in fact a slight simplification of what has been done, made for the purpose of clarity. Naturally, the regexes have to be adopted to the circumstances (regularity of markup, regularity of expressions in brackets, the number of such expressions per single element content, etc.). Sometimes, an XSL transformation may turn out to do a better job, thanks to the many string-handling functions of XPath 2.0.

reversal or concatenation. The <def/> element above is not abused anymore and holds a real definition, i.e., an "explanatory equivalent", which may become a sense-discriminating note in the reversed dictionary.

We stress that each of the XML structures presented above (some of them admittedly bordering on tag abuse) conforms to the general P5 format and can be easily processed and published. In other words, dictionary editors are not forced to conform to the final format in order to see their work being used and commented on.

The next section presents another aspect of dictionary creation, where what matters is the ease of data manipulation and filling in the predictable information for the developer.

3 Plural forms: an illustration of automated creation of entries

At the stage of development at which brackets have been eliminated from translation equivalents, a relatively simple entry might look as shown below. This is an entry as created by a developer, to be further processed by XSLT.[3]

```
<entry>
   <form>
    <orth>adui</orth>
    <ref target="#maadui"/>
   </form>
   <gramGrp><pos>n</pos></gramGrp>
   <sense><def>enemy</def></sense>
   <sense>
    <def>opponent</def>
    <note type="hint">in games or
    sports</note>
   </sense>
</entry>
```

This entry is then processed and turned into the form presented below.

```
<entry xml:id="adui">
   <form>
    <orth>adui</orth>
   </form>
   <xr type="plural-form"><ref tar-
   get="#maadui">maadui</ref></xr>
   <gramGrp>
    <pos>n</pos>
   </gramGrp>
   <sense xml:id="adui.1" n="1">
    <def>enemy</def>
   </sense>
   <sense xml:id="adui.2" n="2">
    <def>opponent</def>
    <note type="hint">in games or
    sports</note>
   </sense>
</entry>
```

If the plural form does not exist in the dictionary, the script creates an entry for it:

```
<entry xml:id="maadui">
   <form>
    <orth>maadui</orth>
   </form>
   <gramGrp>
    <pos>n</pos>
   </gramGrp>
   <sense>
    <xr type="plural-sense">Plural of
    <ref target="#adui">adui</ref>
    </xr>
    <def>enemy</def>
    <def>opponent <note type="hint">in
    games or sports</note></def>
   </sense>
</entry>
```

The equivalents in this kind of automatically created entries for plurals will remain within <def/> elements even after we adopt the cit/quote system mentioned above in the discussion of *alasiri*. This is because <def/> elements are not anchors for dictionary reversal, and plural entries will be skipped by the reversal tools, unless the plural/collective form has its own unique meaning, as is the case with e.g. *majani*, which is morphologically the plural form of *jani* "leaf; blade of grass", but apart from that, it should also be glossed as "grass", and it is the latter form that should become a headword in the reversed, English-Swahili dictionary.

Another area where the XML format and tools give excellent results is text normalization. An earlier example shows unnecessary spaces in version 0.0.2: <def> afternoon</def>. Handling these required only the use of an XPath function `normalize-space()`, which strips all the unwanted whitespace characters.[4]

All the indexing is also done automatically. The indexing system in this particular dictionary is based on the shape of the headword, which it is easy to convert into form acceptable by the XML ID attributes (the XPath `translate()` and `replace()` functions are handy here). All

[3] The verbosity of XML markup can be overwhelming, but many XML editors feature content completion and can save the developer a lot of typing, the more so that TEI schemas are often part of the editor package.

[4] That version contained more traps for machine-processing, such as bracketed parts of words — sometimes this was done in a nontrivial manner, as in the entry for *adui*: <def> enemy(-ies)</def>. See Prinsloo and de Schryver (2002) for remarks on the non-friendliness of such space-saving devices.

the entries are first reordered alphabetically, then the script checks for homographs and assigns them appropriate indexes (e.g. `chapa-1` for the noun meaning "brand", `chapa-2` for the verb meaning "beat") and appropriate attributes used later in the creation of superscripts (suppressed in the plain-text DICT format). Multiple senses are also treated similarly — each receives its own `@xml:id` attribute and numbering (see the example of *adui* above).

We emphasize the fact that the encoding format makes it possible to reduce the developer's workload, with each stage of the dictionary enhancement being publishable. This allows one to work on the dictionary on and off, in their spare time.

4 Visualization of underlying structure

Some Swahili words are best lemmatized as stems. Version 0.4 of our dictionary does not yet display the differences between bound stems and free forms, but we will transfer this functionality (which we use in another project) to one of the future versions. This will make it possible for us to, e.g., add hyphens to bound forms and prepend "-a " to adjectives introduced by the "-*a* of relationship", adding extra structure visible to the end user, but ignored in sorting or queries.

Disjunctively written languages can be handled similarly. Kiango (2000:36) lists the following examples from Haya, discussing the problems surrounding alphabetization of nouns in print dictionaries:

```
a ka yaga      'air'
e m pambo      'seed'
o mu twe       'head'
```

The vocalic pre-prefixes should not be used for the purpose of arranging headwords, because if they are, all nouns end up under one of the three letters. Instead, class prefixes (*ka*, *m*, *mu*) should form the basis for alphabetization. In the XML/TEI format adopted by FreeDict, such problems are easily solved, either by using a separate element to hold the pre-prefix, or by the use of an appropriate attribute. The former solution is illustrated below.

```
<entry>
   <form>
    <orth extent="ppref">a</orth>
    <orth>ka yaga</orth>
   </form>
   <def>air</def>
   </entry>
```

The default value for the `@extent` attribute is "full", so it only needs to be mentioned where the value is different.

An AfLaT reviewer rightly points out that in an electronic dictionary, alphabetization is irrelevant. Indeed, the DICT format features separate "dictionary" and "index" files, and searching is done on the index file, which addresses the relevant portions of the dictionary file. The issue of alphabetization arises, however, in two cases: when preparing a print version of the dictionary, or when using the dictionary outside of the DICT system (this is a "working view" for the maintainers that can also be used as an out-of-the-box view for users).

In connection with the first case — preparing print/PDF versions of dictionaries — it is worth pointing out that conversion from TEI XML into various publication formats is made easy thanks to the open-source XSLT conversion suite maintained by Sebastian Rahtz (http://www.tei-c.org/Tools/Stylesheets/).

As for the "out-of-the-box preview", FreeDict dictionaries, by virtue of being marked up in XML, can be equipped with CSS stylesheets that make it possible to display the XML source in the browser, as if it were an HTML page. Here, because the user can search the page for the given form, alphabetization is not so relevant, but it can be handy, if only for aesthetic reasons. Below is a screenshot of a fragment of the CSS view of the file swa-eng.tei from version 0.4 distribution package, opened directly in the Firefox browser.

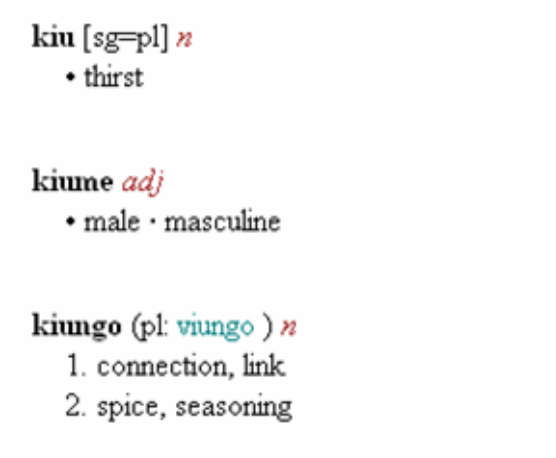


Figure 1: A CSS preview of the source XML

The CSS adds some text (e.g. "[sg=pl]", "(pl:", or sense numbering) and imposes visual structure onto the source XML. As can be seen in the entry for *kiungo*, we give precedence to formal

properties of headwords over the semantic distinctions, but other macrostructural decisions are obviously possible as well. The figure below shows one more CSS view, demonstrating subcategorization, treatment of notes (all the bracketed strings are contents of separate XML elements, with parentheses supplied by the CSS), as well as the treatment of homographs.

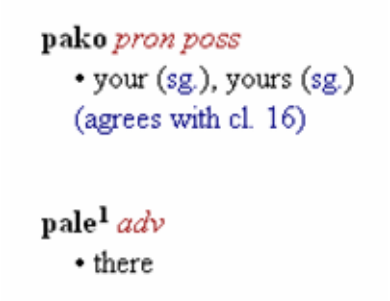


Figure 2: Subcategorization and grammar notes (another CSS view of the source XML)

Our use of colours (here: shades of grey) is also a function of the CSS, introduced mainly with the developer in mind, as a kind of an error-checking device.

5 Other possible enhancements of the microstructure

In section 3, we have illustrated a possible method of refining dictionary structure in an automatic fashion. Thanks to the many possible variations of the format, other features may be introduced stage by stage. They include, e.g., adding the corresponding plurals (illustrated above) or marking forms where the plural is the same as the singular, as done below by the use of the `@type` attribute. This example also illustrates the addition of cross-references to synonyms, where *eropleni* is linked to the second, inanimate, sense of *ndege* 'bird; airplane'.

```
<entry xml:id="eropleni">
   <form type="N">
    <orth>eropleni</orth>
   </form>
   <gramGrp>
    <pos>n</pos>
   </gramGrp>
   <sense>
    <def>airplane</def>
    <xr type="syn">(synonym: <ref tar
    get="#ndege.2">ndege</ref>)</xr>
   </sense>
</entry>
```

The `<form/>` element below illustrates the handling of alternative spellings of the noun *afisa*/*ofisa* "officer". Both headwords are used in retrieval.

```
<entry xml:id="afisa">
  <form>
    <orth n="1">afisa</orth>
    <orth type="variant"
          n="2">ofisa</orth>
    <ref target="#maafisa" n="1"/>
    <ref target="#maofisa" n="2"/>
  </form>
```

The above is the source as prepared by a developer. This is then processed, the plural forms are created if they do not exist in the dictionary, and the result is as in figure 3 below.

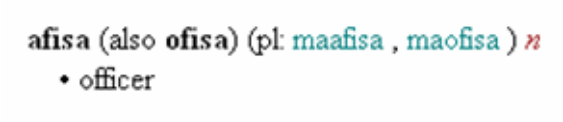


Figure 3: Illustration of alternative spellings/plurals (CSS view)

Other examples of what can be gradually introduced into the dictionary include addition of subcategorization information (illustrated by *pako* in Figure 2, where "pron" is the POS and "poss" is the content of the `<subc/>` element), addition of explicit noun-class- and agreement-marking, introduction of irregularly inflected forms, tables with inflection (linked to the appropriate stems), nested entries, and, obviously, the continuous improvement of lexicographic information (the arrangement and selection of senses, selection of headwords, an appropriate POS system).

Crucially, this system allows a developer to "publish early, publish often", and few of the enhancements mentioned here depend on others — developers are free to choose and to extend the dictionary at their own pace.

In our case, this is a gradual move towards structures of finer granularity, suitable for reversal (into English-Swahili) and concatenation with other dictionaries (we are going to use English as the bridge language for pairing the Swahili-English dictionary with English-* dictionaries).

6 Potential for the future

We wish to stress the potential that the encoding format and the entire project have for producing lexical resources for "non-commercial" languages, where funding and the time that the developers may spend on the dictionary are not always guaranteed. FreeDict dictionary development can proceed in stages, one can start with a

simple format and get the dictionary published on-line practically within days. The project has all the SourceForge publishing facilities at its disposal, together with bug/patch/etc. trackers and community forums. It also has a mailing list and a wiki that can serve to document some possibly difficult aspects of dictionary creation.

Thanks to the robust build system of FreeDict, creating a tarball containing a DICT-formatted dictionary and index is only a matter of issuing the "make" command with appropriate arguments, and submitting the resulting archive to the SourceForge file release system.[5]

FreeDict is the nexus for the following:

- XML, with its potential for creating well-structured documents,
- TEI P5, a *de-facto* standard taking advantage of this potential,
- the SourceForge repository as well as distribution and content-management network,
- the DICT distribution network: apart from being able to query DICT servers straight from the desktop, Firefox users can also take advantage of an add-on client that returns definitions for highlighted words on a web page,
- FreeDict tools (still under development for TEI P5) as means to manipulate dictionaries and to create, among others, the DICT format (usable directly from DICT servers and by other dictionary-providing projects, e.g., StarDict or Open Dict).[6]

Additionally, lexical resources submitted to FreeDict may undergo further transformations: reversal or concatenation, which means that work put into developing a single resource may well be re-used in developing others. Considering the possible re-use of lexical resources, they are expected to be prepared with a view towards clean exposure of translation equivalents (in the cit/quote system or at least by judicious use of separators and brackets).

The project has its own distribution system, in the form of GNU/Linux packages — for example, Kęstutis Biliūnas is the packager for Debian Linux and maintains a page tracking the usage of Debian-FreeDict packages;

Apart from the above, the content published by FreeDict is guaranteed to be free.

7 The costs of developing for FreeDict

An AfLaT reviewer suggested that we provide a measure of the effort required to develop a resource for FreeDict. We hope to show here that this is very much dependent on the quality and form of the resource and on how much time the dictionary creator is willing to invest into it. Crucially, given the open-source nature of the project, even a simple, small list of near-binary pairings of equivalents can be a) quickly made useful to e.g. the readers of web pages written in the given language, and b) extended by others into a more satisfying resource.

It may be that the effort needed to create language resources in e.g. Lexique Pro, an excellent free tool by SIL International (http://www.lexiquepro.com/) is smaller, but there are differences in that Lexique Pro is a Windows-only closed-source program whose native MDF (Multi-Dictionary-Formatter) format is not as flexible as XML and therefore cannot be processed by the many tools that handle XML and TEI in particular. Consequently, the perspectives for re-use of Lexique Pro dictionaries in computational linguistic applications are much smaller. To our knowledge, Lexique Pro does not make it possible for users to query words straight off web pages, which can be done thanks to dict, a Firefox add-on (http://dict.mozdev.org/). It admittedly has other advantages that make it a serious alternative.[7]

The ideal solution would be to have an editing front-end such as Lexique Pro coupled with the openness and modifiability of the data offered by FreeDict. Indeed, there are plans for creating a converter from the new LIFT interchange standard (http://code.google.com/p/lift-standard/) that the beta versions of Lexique Pro can read

[5] This is something that a dictionary creator need not bother about — submitting a TEI source of the dictionary to the mailing list is enough.

[6] The FreeDict build process provides targets for platforms other than DICT, e.g. the Evolutionary Dictionary (http://www.mrhoney.de/y/1/html/evolutio.htm) or zbedic (http://bedic.sourceforge.net/).

[7] We do not discuss professional commercial dictionary writing systems such as TshwaneLex (http://tshwanedje.com/tshwanelex/) because, despite the academic discounts, they may be out of range for the average developer. It is worth mentioning that the discounted versions of TshwaneLex come with the understandable "no-commercial-use" restriction, which is in conflict with either the GNU Public License or the nearly equivalent Creative Commons BY-SA license that all SourceForge resources must be under (cf. http://www.gnu.org/licenses/gpl-faq.html).

and write, to the version of the TEI format used by FreeDict. That would undoubtedly enhance the attractiveness of the project.

To sum up, developing for FreeDict minimally requires some basic knowledge of XML. Free XML editors exist (e.g. XML Copy Editor, http://sourceforge.net/projects/xml-copy-editor/) that can make editing easier by autocompleting the elements (inserting closing tags, suggesting elements and attributes that are allowed at the given place in the structure) and signalling encoding mistakes.

8 African Languages and FreeDict

A reviewer remarked that the link to African language technology in this paper appears only to be present in the examples. Indeed, FreeDict is not a project that focuses on African-languages — it is a project where African language resources can be hosted and quickly become useful to users. Given an opportunity, we will encourage researches dealing with other languages to join the project — which will hopefully result in the creation of more cross-language resources, especially given that the encoding format is not tied to any particular language and is able to easily accommodate features characteristic of practically any language.

During the session on "African Languages in Advance" at the 2008 Poznań Linguistic Meeting, where we presented our Swahili-Polish project and also mentioned FreeDict as the place where we wanted to donate parts of our test Swahili-English dictionary that would otherwise remain on our disks, we talked to an organizer of that session, Karien Brits, about the advantages that this project can have for some of her colleagues working on South-African languages. This is what encouraged us to move on with the FreeDict Swahili-English dictionary and we hope that others will also find this project, and the possibilities that it offers, attractive.

The FreeDict project has recently awoken after a period of lower activity, and at the moment, every week brings something new. Currently, as far as African languages are concerned, apart from Swahili↔English dictionaries, the project hosts very basic Afrikaans↔English dictionaries, and an Afrikaans-German dictionary (all of them in need of a maintainer).[8] We hope that FreeDict will become home to many African language resources, and, thanks to the possibility of dictionary concatenation, facilitate also the creation of many African↔European dictionary pairs as well as all-African bilingual dictionaries.

[8] Dictionary sources can be accessed from the "download" link on the project page: http://sourceforge.net/projects/freedict/. They can be queried online at e.g. http://dict.org/, by setting the appropriate language pair in the database.

Acknowledgements

We are grateful to three anonymous AfLaT-2009 reviewers for their helpful comments on the previous version of this paper, and to Michael Bunk for confirming that our version of the history of the project is close to reality. We also wish to thank Janusz S. Bień for turning our attention to the FreeDict project a few years ago.

References

DICT: http://www.dict.org/

Faith, Rik and Martin, Brett. 1997. A Dictionary Server Protocol. Request for Comments: 2229 (RFC #2229). Network Working Group. Available from ftp://ftp.isi.edu/in-notes/rfc2229.txt (last accessed on January 19, 2009).

FreeDict: http://www.freedict.org/, http://freedict.sourceforge.net/

Kiango, John G. 2000. *Bantu lexicography: a critical survey of the principles and process of constructing dictionary entries*. Tokyo: Institute for the Study of Languages and Cultures of Asia and Africa, Tokyo University of Foreign Studies.

Prinsloo, Danie J. and Gilles-Maurice de Schryver. 2002. Reversing an African-language lexicon: the *Northern Sotho Terminology and Orthography No. 4* as a case in point. *South African Journal of African Languages*, 22/2: 161–185

Sperberg-McQueen, Michael and Lou Burnard (eds). 2002. *The Text Encoding Initiative Guidelines (P4)*. Text Encoding Initiative, Oxford.

TEI Consortium, eds. 2007. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Version 1.2.0. Last updated on October 31st 2008. TEI Consortium. Available from http://www.tei-c.org/Guidelines/P5/ (last accessed on January 19, 2009).

Exploiting Cross-linguistic Similarities in Zulu and Xhosa Computational Morphology

Laurette Pretorius
School of Computing
University of South Africa &
Meraka Institute, CSIR
Pretoria, South Africa
pretol@unisa.ac.za

Sonja Bosch
Department of African Languages
University of South Africa
Pretoria, South Africa
boschse@unisa.ac.za

Abstract

This paper investigates the possibilities that cross-linguistic similarities and dissimilarities between related languages offer in terms of bootstrapping a morphological analyser. In this case an existing Zulu morphological analyser prototype (ZulMorph) serves as basis for a Xhosa analyser. The investigation is structured around the morphotactics and the morphophonological alternations of the languages involved. Special attention is given to the so-called "open" class, which represents the word root lexicons for specifically nouns and verbs. The acquisition and coverage of these lexicons prove to be crucial for the success of the analysers under development. The bootstrapped morphological analyser is applied to parallel test corpora and the results are discussed. A variety of cross-linguistic effects is illustrated with examples from the corpora. It is found that bootstrapping morphological analysers for languages that exhibit significant structural and lexical similarities may be fruitfully exploited for developing analysers for lesser-resourced languages.

1 Introduction

Zulu and Xhosa belong to the Nguni languages, a group of languages from the South-eastern Bantu zone and, as two of the eleven official languages of South Africa, are spoken by approximately 9 and 8 million mother-tongue speakers, respectively. In terms of natural language processing, particularly computational morphology, the Bantu languages including Zulu and Xhosa certainly belong to the lesser-studied languages of the world.

One of the few Bantu languages for which computational morphological analysers have been fully developed so far is Swahili (Hurskainen, 1992; De Pauw and De Schryver, 2008). A computational morphological analyser prototype for Zulu (ZulMorph) is in an advanced stage of development, the results of which have already been used in other applications. Preliminary experiments and results towards obtaining morphological analysers for Xhosa, Swati and Ndebele by bootstrapping ZulMorph were particularly encouraging (Bosch et al., 2008). This bootstrapping process may be briefly summarised as a sequence of steps in which the baseline analyser, ZulMorph, is applied to the new language (in this case Xhosa) and then systematically extended to include the morphology of the other language. The extensions concern the word root lexicon, followed by the grammatical morpheme lexicons and finally by the appropriate morphophonological rules. The guiding principle in this process is as follows: Use the Zulu morphological structure wherever applicable and only extend the analyser to accommodate differences between the source language (Zulu) and the target language (in this case Xhosa). So far the question as to whether the bootstrapped analyser, extended to include Xhosa morphology, could also improve the coverage of the Zulu analyser was not specifically addressed in Bosch et al. (2008).

Cross-linguistic similarity and its exploitation is a rather wide concept. In its broadest sense it aims at investigating and developing resources and technologies that can be compared and linked, used and analysed with common approaches, and that contain linguistic information for the same or comparable phenomena. In this paper the focus is on the morphological similarities and dissimilarities between Zulu and Xhosa and how these cross-linguistic similarities and dissimilarities inform the bootstrapping of a morphological analyser for Zulu and Xhosa. In particular, issues such as open versus closed classes, and language specific morphotactics and alternation rules are discussed. Special attention

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 96–103,
Athens, Greece, 31 March 2009.

is given to the word root lexicons. In addition, the procedure for bootstrapping is broadened to include a guesser variant of the morphological analyser.

The structure of the paper is as follows: Section 2 gives a general overview of the morphological structure of the languages concerned. The modelling and implementation approach is also discussed. This is followed in sections 3 and 4 by a systematic exposition of the cross-linguistic dissimilarities pertaining to morphotactics and morphophonological alternations. Section 5 focuses on the so-called "open" class, which represents the word root lexicons for specifically nouns and verbs. The acquisition and coverage of these lexicons prove to be crucial for the success of the analysers under development. Section 6 addresses the use of the guesser variant of the morphological analyser as well as the application of the bootstrapped morphological analyser to parallel test corpora. A variety of cross-linguistic effects is illustrated with examples from the corpora. This provides novel insights into the investigation and exploitation of cross-linguistic similarities and their significance for bootstrapping purposes. Section 7 concerns future work and a conclusion.

2 General overview

2.1 Morphological structure

Bantu languages are characterised by a rich agglutinating morphological structure, based on two principles, namely the nominal classification system, and the concordial agreement system. According to the nominal classification system, nouns are categorised by prefixal morphemes. These noun prefixes have, for ease of analysis, been assigned numbers by scholars who have worked within the field of Bantu linguistics, In Zulu a noun such as *umuntu* 'person' for instance, consists of a noun prefix *umu-* followed by the noun stem *-ntu* and is classified as a class 1 noun, while the noun *isitha* 'rival' consists of a noun prefix *isi-* and the noun stem *-tha* and is classified as a class 7 noun. Noun prefixes generally indicate number, with the uneven class numbers designating singular and the corresponding even class numbers designating plural. The plural forms of the above examples would therefore respectively be the class 2 noun *abantu* 'persons' and the class 8 noun *izitha* 'rivals'. We follow Meinhof's (1932:48) numbering system which distinguishes between 23 noun prefixes altogether in the various Bantu languages.

The concordial agreement system is significant in the Bantu languages because it forms the backbone of the whole sentence structure. Concordial agreement is brought about by the various noun classes in the sense that their prefixes link the noun to other words in the sentence. This linking is manifested by a concordial morpheme that is derived from the noun prefix, and usually bears a close resemblance to the noun prefix, as illustrated in the following example:

Izi*tsha le****zi ezi****ne* ***z****ephukile*
'These four plates are broken'

This concordial agreement system governs grammatical correlation in verbs, adjectives, possessives, pronouns, and so forth. Bantu languages are predominantly agglutinating and polymorphematic in nature, with affixes attached to the root or core of the word.

The morphological make-up of the verb is considerably more complex than that of the noun. A number of slots, both preceding and following the verb root may contain numerous morphemes with functions such as derivations, inflection for tense-aspect and marking of nominal arguments. Examples are cross-reference of the subject and object by means of class- (or person-/number-)specific object markers, locative affixes, morphemes distinguishing verb forms in clause-final and non-final position, negation etc.

Despite the complexities of these domains, they are comparable across language boundaries, specifically Nguni language boundaries, with a degree of formal similarity that lends itself to exploitation for bootstrapping purposes.

2.2 Modelling and Implementation

In the modelling and implementation of the morphological structure a finite-state approach is followed. The suitability of finite-state approaches to computational morphology is well known and has resulted in numerous software toolkits and development environments for this purpose (cf. Koskenniemi, 1997 and Karttunen, 2001). Yli-Jyrä (2005) discusses the importance of a finite-state morphology toolkit for lesser-studies languages. He maintains that "[a]lthough some lexicons and morphological grammars can be learned automatically from texts ... fully automatic or unsupervised methods are not sufficient. This is due to two reasons. First, the amount of freely available corpora is limited for many of the less studied languages. Second, many of the less studied languages have rich morphologies that are difficult to learn accurately with unsupervised methods".

The Xerox finite-state tools (Beesley and Karttunen, 2003) as one of the preferred toolkits for modelling and implementing natural language morphology, is used in this work.

The morphological challenges in computational morphological analysis comprise the modelling of two general linguistic components, namely morphotactics (word formation rules) as well as morphophonological alternations.

Ideally, the morphotactics component should include all and only word roots in the language, all and only the affixes for all parts-of-speech (word categories) as well as a complete description of the valid combinations and orders of these morphemes for forming all and only the words of the language concerned. Moreover, the morphophonological alternations rules should constitute all known sound changes that occur at morpheme boundaries. The combination of these two components constitutes an accurate model of the morphology of the language(s) under consideration.

The Xerox lexicon compiler, **lexc**, is well-suited to capturing the morphotactics of Zulu. A **lexc** script, consisting of cascades of so-called continuation classes (of morpheme lexicons) representing the (concatenative) morpheme sequencing, is compiled into a finite-state network. The Xerox regular expression language, **xfst**, provides an extended regular expression calculus with sophisticated Replace Rules for describing the morphophonological alternations rules of Zulu. The **xfst** script is also compiled into a finite-state network. These networks are finally combined by means of the operation of composition into a so-called Lexical Transducer that constitutes the morphological analyser and contains all the morphological information of Zulu, including derivation, inflection, alternation and compounding. Pretorius and Bosch (2002) address the suitability of this approach to Zulu morphology and illustrate it by means of examples of **lexc** and **xfst** scripts for modelling the Zulu noun.

A detailed exposition of the design and implementation of ZulMorph may be found in Pretorius and Bosch (2003). In addition to considering both the accurate modelling of the morphotactics and the morphophonological alternation rules, they also address implementation and other issues that need to be resolved in order to produce a useful software artefact for automated morphological analysis. Issues of implementation include a justification for the finite-state approach followed, designing for accuracy and correctness and decisions regarding the analyser's interface with its environment and its usage.

Particular attention is paid to the handling of exceptions; the modelling of separated dependencies by means of so-called flag-diacritics; the specification of lexical forms (analyses) in terms of morphological granularity and feature information; the choice of an associated and appropriate morphological tag set and also the positioning of these tags in relation to the morphemes they are associated with in the morphological analyses (lexical forms) that are rendered.

The components of ZulMorph, including its scope in terms of word categories and their morphological structure, are summarised in Table 1 while its lexical coverage as reflected by the number of different noun stems, verb roots etc. is discussed in section 5.

The bootstrapping of ZulMorph to provide for Xhosa as well requires a careful investigation of the cross-linguistic similarities and dissimilarities and how they are best modelled and implemented. This aspect will be discussed in more detail in the following section.

Morphotactics (**lexc**)	Affixes for all parts-of-speech (e.g. subject & object concords, noun class prefixes, verb extensions etc.)	Word roots (e.g. nouns, verbs, relatives, ideophones)	Rules for legal combinations and orders of morphemes (e.g. *u-ya-ngi-thand-a* and not **ya-u-a-thand-ngi*)
Morphophonological alternations (**xfst**)	Rules that determine the form of each morpheme (e.g. *ku-lob-w-a* > *ku-lotsh-w-a*, *u-mu-lomo* > *u-m-lomo*)		

Table 1: Zulu Morphological Analyser Components

3 Morphotactics

In word formation we distinguish between so-called closed and open classes. The open class accepts the addition of new items by means of processes such as borrowing, coining, compounding and derivation. In the context of this paper, the open class represents word roots including verb roots and noun stems. The closed class represents affixes that model the fixed morphological structure of words, as well as items such as conjunctions, pronouns etc. Typically no new items can be added to the closed class (Fromkin et al., 2003:74).

Since our point of departure is ZulMorph, we focus on Xhosa affixes that differ from their Zulu

counterparts. A few examples are given in Table 2.

Certain areas in the Xhosa grammar need to be modelled independently and then built into the

Morpheme	Zulu	Xhosa
Noun Class Prefixes		
Class 1 and 3 *um(u)-*	full form *umu-* with monosyllabic noun stems, shortened form with polysyllabic noun stems: *umu-ntu, um-fana*	um- with all noun stems: *um-ntu, um-fana*
Class 2a	*o-: o-baba*	*oo-: oo-bawo*
Class 9	*in-* with all noun stems: *in-nyama*	*i-* with noun stems beginning with *h, i, m, n, ny*: *i-hambo*
Class 10	*izin-* with monosyllabic and polysyllabic stems. *izin-ja; izin-dlebe*	*iin-* with polysyllabic stems: *iin-dlebe*
Contracted subject concords (future tense). Examples:		
1ps 2ps, Class 1 & 3 Class 4 & 9	*ngo-* *wo-* *yo-*	*ndo-* *uyo-* *iyo-*
Object concords		
1ps	*ngi-*	*ndi-*
Absolute pronouns		
1ps Class 15	*mina* *khona*	*mna* *kona*
Demonstrative Pronouns: Three positional types of the demonstrative pronouns are listed separately for each language. Examples:		
Class 1 Class 5	Pos. 1 *lo*; Pos. 2 *lowo*; Pos. 3 *lowaya* Pos. 1 *leli*; Pos. 2 *lelo*; Pos. 3 *leliya*	Pos. 1 *lo*; Pos. 2 *lowo/loo*; Pos. 3 *lowa* Pos. 1 *eli*; Pos. 2 *elo*; Pos. 3 *eliya*
Adjective basic prefixes		
1ps 2ps Class1& 3 Class 8	*ngim(u-)* *umu-* *mu-* *zin-*	*nim-* *um-* *m-* *zi-*
Locative demonstrative copulatives : Three positional types of the so-called locative demonstrative copulatives differ considerably for Zulu and Xhosa and are therefore listed separately for each language. Examples:		
Class 1 Class 5	Pos. 1 *nangu*; Pos. 2 *nango*; Pos. 3 *nanguya* Pos. 1 *nanti*; Pos. 2 *nanto*; Pos. 3 *nantiya*	Pos. 1 *nanku*; Pos. 2 *nanko*; Pos. 3 *nankuya* Pos. 1 *nali*; Pos. 2 *nalo*; Pos. 3 *naliya*
Copulatives : Formation of copulatives derived from Xhosa nouns differs considerably from Zulu. This construction is class dependent in Xhosa and is modelled differently to its Zulu counterpart. Examples:		
	yi- combines with noun prefixes i-: *yi-indoda > yindoda* *ngu-* combines with noun prefixes *u-, o-, a*: *ngu-umuntu > ngumuntu* *ngu-obaba > ngobaba* *ngu-amakati > ngamakati* *wu* combines with noun prefixes *u-, o-:* *wu-muntu > wumuntu,* *wu-obaba > wobaba*	*ngu-* combines with classes 1, 1a, 2, 2a, 3 & 6, e.g. *ngu-umntu > ngumntu* *yi-* combines with classes 4 *imi-* and 9 *in-*, e.g. *yi-imithi > yimithi* *li-* combines with class 5 *i(li)-:* *li-ihashe > lihashe* *si-* combines with class 7 *isi-:* *si-isitya > sisitya* etc.

Table 2. Examples of variations in Zulu and Xhosa 'closed' morpheme information

analyser, for instance the formation of the so-called temporal form that does not occur in Zulu. The temporal form is an indication of when an action takes place or when a process is carried out, and has a present or past tense form (Louw, et al., 1984:163). The simple form consists of a subject concord plus *-a-* followed by the verb stem in the infinitive, the preprefix of which has been elided, for example *si-a-uku-buya > sa-kubuya* 'when we return'. In terms of the word formation rules this means that an additional Xhosa specific morpheme lexicon (continuation class) needs to be included. To facilitate accurate modelling appropriate constraints also need to be formulated.

The bootstrapping process is iterative and new information regarding dissimilar morphological constructions is incorporated systematically in

the morphotactics component. Similarly, rules are adapted in a systematic manner. The process also inherently relies on similarities between the languages, and therefore the challenge is to model the dissimilarities accurately. The carefully conceptualised and appropriately structured (**lexc**) continuation classes embodying the Zulu morphotactics provide a suitable framework for including all the closed class dissimilarities discussed above.

4 Morphophonological alternations

Differences in morphophonological alternations between Zulu and Xhosa are exemplified in Table 3. Some occur in noun class prefixes of class 10 and associated constructions, such as prefixing of adverbial morphemes (*na-*, *nga-*, etc.). Others are found in instances of palatalisation, "a sound change whereby a bilabial sound in passive formation, locativisation and diminutive formation is replaced by a palatal sound" (Poulos and Msimang, 1998:531).

Zulu	Xhosa
Class 10 class prefix *izin-* occurs before monosyllabic as well as polysyllabic stems, e.g. *izinja*, ***izindlebe*** Adverb prefix *na + i > ne*, e.g. ***nezindlebe*** (*na-izin-ndlebe*)	Class 10 class prefix *izin-* changes to *iin-* before polysyllabic stems, e.g. *izinja, iindlebe* Adverb prefix *na + ii > nee*; e.g. ***neendlebe*** (*na-iin-ndlebe*)
Palatalisation with passive, diminutive & locative formation: ***b > tsh*** *-hlab-w-a > -hlatsh-w-a, intaba-ana > intatsh-ana, indaba > entdatsheni* ***ph > sh*** *-boph-w-a > -bosh-w-a, iphaphu-ana > iphash-ana iphaphu > ephasheni*	Palatalisation with passive, diminutive & locative formation: ***b > ty*** *-hlab-w-a > -hlaty-w-a, intaba-ana > intaty-a na ihlobo > ehlotyeni* ***ph > tsh*** *–boph-w-a > -botsh-w-a, iphaphu-ana > iphatsh-ana, usapho > elusatsheni*

Table 3. Examples of variations in Zulu and Xhosa morphophonology

As before, the Zulu alternations are assumed to apply to Xhosa unless otherwise modelled. Regarding language-specific alternations special care is taken to ensure that the rules fire only in the desired contexts and order. For example, Xhosa-specific sound changes should not fire between Zulu-specific morphemes, and vice versa. This applies, for instance, to the vowel combination *ii*, which does not occur in Zulu. While the general rule *ii > i* holds for Zulu, the vowel combination *ii* needs to be preserved in Xhosa.

5 The word root lexicons

Compiling sufficiently extensive and complete word root lexicons (i.e. populating the "open" word classes) is a major challenge, particularly for lesser-resourced languages (Yli-Jyrä, 2005:2). A pragmatic approach of harvesting roots from all readily available sources is followed. The Zulu lexicon is based on an extensive word list dating back to the mid 1950s (cf. Doke and Vilakazi, 1964), but significant improvements and additions are regularly made. At present the Zulu word roots include noun stems with class information (15 759), verb roots (7 567), relative stems (406), adjective stems (48), ideophones (1 360), conjunctions (176). Noun stems with class information (4 959) and verb roots (5 984) for the Xhosa lexicon were extracted from various recent prototype paper dictionaries whereas relative stems (27), adjective stems (17), ideophones (30) and conjunctions (28) were only included as representative samples at this stage.

The most obvious difference between the two word root lexicons is the sparse coverage of nouns for Xhosa. A typical shortcoming in the current Xhosa lexicon is limited class information for noun stems.

Observations are firstly occurrences of shared noun stems (mainly loan words) but different class information, typically class 5/6 for Zulu versus class 9/10 for Xhosa, for example

'box' *-bhokisi* (Xhosa 9/10; Zulu 5/6)
'duster' *-dasta* (Xhosa 9/10; Zulu 5/6)
'pinafore' *-fasikoti* (Xhosa 9/10; Zulu 5/6).

It should be noted that although a Xhosa noun stem may be identical to its Zulu counterpart, analysis is not possible if the class prefix differs from the specified Zulu class prefix + noun stem combination in the morphotactics component of the analyser.

A second observation is identical noun stems with correct class information, valid for both languages, but so far only appearing in the Xhosa lexicon, for example

'number' *-namba* (Xhosa and Zulu 9/10)
'dice'*-dayisi* (Xhosa and Zulu 5/6).

This phenomenon occurs mainly with borrowed nouns that are more prevalent in the Xhosa lexicon than in the more outdated Zulu lexicon.

A closer look at the contents of the lexicons reveals that the two languages have the following in common: 1027 noun stems with corresponding class information, 1722 verb roots, 20 relative stems, 11 adjective stems, 10 ideophones and 9 conjunctions.

6 A computational approach to cross-linguistic similarity

This section discusses the extension of the bootstrapping procedure of the morphological analyser to include the use of the guesser variant of the morphological analyser. In addition the application of the bootstrapped morphological analyser to parallel test corpora is addressed. A variety of cross-linguistic effects is illustrated with examples from the corpora.

Even in languages where extensive word root lexicons are available, new word roots may occur from time to time. The Xerox toolkit makes provision for a **guesser variant** of the morphological analyser that uses typical word root patterns for identifying potential new word roots (Beesley and Karttunen, 2003:444). By exploiting the morphotactics and morphophonological alternations of the analyser prototype, the guesser is able to analyse morphologically valid words of which the roots match the specified pattern. Therefore, in cases where both the Zulu and Xhosa word root lexicons do not contain a root, the guesser may facilitate the bootstrapping process.

The extended **bootstrapping procedure** is schematically represented in Figure 1.

Since the available Zulu word list represents a rather outdated vocabulary, it is to be expected that the coverage of word roots/stems from a recent corpus of running Zulu text may be unsatisfactory, due to the dynamic nature of language. For example the word list contains no entry of the loan word *utoliki* 'interpreter' since 'interpreter' is rendered only as *i(li)humusha* 'translator', the traditional term derived from the verb stem *–humusha* 'to translate, interpret'. Provision therefore needs to be made for the constant inclusion of new roots/stems, be they newly coined, compounds or as yet unlisted foreign roots/stems.

Updating and refining the lexicon requires the availability of current and contemporary

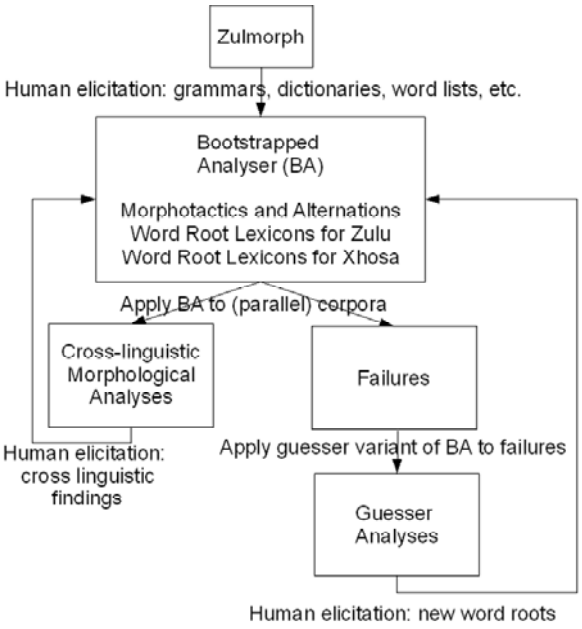


Figure 1. Bootstrapping procedure

language resources in the form of **text corpora** as well as human intervention in the form of expert lexicographers or linguists to determine the eligibility of such words.

The language resources chosen to illustrate this point are parallel corpora in the form of the South African Constitution (The Constitution, (sa). The reason for the choice of these corpora is that they are easily accessible on-line, and it is assumed that the nature of the contents ensures accurate translations.

The **results** of the application of the bootstrapped morphological analyser to this corpus are as follows:

Zulu Statistics

Corpus size: 7057 types
Analysed: 5748 types (81.45 %)
Failures: 1309 types (18.55%)
Failures analysed by guesser: 1239 types
Failures not analysed by guesser: 70 types

Xhosa Statistics

Corpus size: 7423 types.
Analysed: 5380 types (72.48 %)
Failures: 2043 types (27.52%)
Failures analysed by guesser: 1772 types
Failures not analysed by guesser: 271 types

The output of the combined morphological analyser enables a detailed investigation into cross-linguistic features pertaining to the morphology of Zulu and Xhosa. The outcome of this investigation is illustrated by means of typical examples from the corpora. This provides novel nsights into the investigation and exploitation of

cross-linguistic similarities and their significance for bootstrapping purposes, as shown in Figure 1.

Notational conventions include `[Xh]` for Xhosa specific morphemes, numbers indicate noun class information, e.g. `[NPrePre9]` tags the noun preprefix of a class 9 noun while `[RelConc8]` tags the relative concord of a class 8 noun.

Examples from the Zulu corpus:

The analysis of the Zulu word *ifomu* 'form' uses the Xhosa noun stem *–fomu* (9/10) in the Xhosa lexicon in the absence of the Zulu stem:

```
ifomu  i[NPrePre9]fomu[Xh][NStem]
```

The analysis of the Zulu word *ukutolikwa* 'to interpret' uses the Xhosa verb root *-tolik-* in the Xhosa lexicon:

```
ukutolikwa
u[NPrePre15]ku[BPre15]
tolik[Xh][VRoot]w[PassExt]a[VerbTerm]
```

Examples from the Xhosa corpus:

The analysis of the Xhosa words *bephondo* 'of the province' and *esikhundleni* 'in the office' use the Zulu noun stems *–phondo* (5/6) and *-khundleni* (7/8) respectively in the Zulu lexicon:

```
bephondo
ba[PossConc14]i[NPrePre5]li[BPre5]
phondo[NStem]

bephondo
ba[PossConc2]i[NPrePre5]li[BPre5]
phondo[NStem]

esikhundleni
e[LocPre]i[NPrePre7]si[BPre7]
khundla[NStem]ini[LocSuf]
```

The analysis of the Xhosa words *ekukhethweni* 'in the election' and *esihlonyelweyo* 'amended' use the Zulu verb roots *–kheth-* and *–hlom-* respectively in the Xhosa lexicon:

```
ekukhethweni
e[LocPre]u[NPrePre15]ku[BPre15]
kheth[VRoot]w[PassExt]a[VerbTerm]
ini[LocSuf]

esihlonyelweyo
esi[RelConc7]hlom[VRoot]el[ApplExt]
w[PassExt]e[VerbTermPerf]yo[RelSuf]
```

Ideophones used from the Zulu lexicon are:

```
ga[Ideoph]    qho[Ideoph]
sa[Ideoph]    tu[Ideoph]
ya[Ideoph]
```

Relative stems used from the Zulu lexicon are:

```
mandla[RelStem]
mdaka[RelStem]
njalo[RelStem]
mcimbi[RelStem]
```

Conjunctions used from the Zulu lexicon are:

```
futhi[Conj]
ukuthi[Conj]
```

Examples of the guesser output from the Zulu corpus:

The compound noun *-shayamthetho* (7/8) 'legislature' is not listed in the Zulu lexicon, but was guessed correctly:

```
isishayamthetho
i[NPrePre7]si[BPre7]
shayamthetho-Guess[NStem]
```

The following are two examples of borrowed nouns (*amabhajethi* 'budgets' and *amakhemikali* 'chemicals') not in the Zulu lexicon, but guessed correctly:

```
amabhajethi
a[NPrePre6]ma[BPre6]
bhajethi-Guess[NStem]

amakhemikali
a[NPrePre6]ma[BPre6]
khemikali-Guess[NStem]
```

The borrowed verb root *-rejest-* 'register' is not listed in the Zulu lexicon, but was guessed correctly:

```
ezirejestiwe
ezi[RelConc8]rejest-Guess[VRoot]
iw[PassExt]e[VerbTermPerf]

ezi[RelConc10]rejest-Guess[VRoot]
iw[PassExt]e[VerbTermPerf]
```

The relatively small number of failures that are not analysed by the guesser and for which no guessed verb roots or noun stems are offered, simply do not match the word root patterns as specified for Zulu and Xhosa in the analyser prototype, namely

```
[C (C (C)) V]+ C (C (C))
```

for verb roots and

```
[C (C (C)) V]+ C (C (C)) V
```

for noun stems. The majority of such failures is caused by spelling errors and foreign words in the test corpus.

7 Conclusion and Future Work

In this paper we focused on two aspects of cross-linguistic similarity between Zulu and Xhosa, namely the morphological structure (morphotactics and alternation rules) and the word root lexicons.

Regarding the morphological structure only differences between Zulu and Xhosa were added.

Therefore, Zulu informed Xhosa in the sense that the systematically developed grammar for ZulMorph was directly available for the Xhosa analyser development, which significantly reduced the development time for the Xhosa prototype compared to that for ZulMorph.

Special attention was also given to the so-called "open" class, which represents the word root lexicons for specifically nouns and verbs. The acquisition and coverage of these lexicons proved to be crucial for the success of the analysers under development. Since we were fortunate in having access to word root lexicons for both Zulu and Xhosa we included what was available in such a way that word roots could be shared between the languages. Here, although to a lesser extent, Xhosa also informed Zulu by providing a current (more up to date) Xhosa lexicon. In addition, the guesser variant was employed in identifying possible new roots in the test corpora, both for Zulu and for Xhosa.

In general it is concluded that bootstrapping morphological analysers for languages that exhibit significant structural and lexical similarities may be fruitfully exploited for developing analysers for lesser-resourced languages.

Future work includes the application of the approach followed in this work to the other Nguni languages, namely Swati and Ndebele (Southern and Zimbabwe); the application to larger corpora, and the subsequent construction of stand-alone versions. Finally, the combined analyser could also be used for (corpus-based) quantitative studies in cross-linguistic similarity.

Acknowledgements

This material is based upon work supported by the South African National Research Foundation 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.

References

Beesley, K.R. and Karttunen, L. 2003. *Finite State Morphology*. CSLI Publications, Stanford, CA.

Bosch, S., Pretorius, L., Podile, K. and Fleisch, A. 2008. Experimental fast-tracking of morphological analysers for Nguni languages. *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco. ISBN 2-9517408-4-0.

De Pauw, G. and de Schryver, G-M. 2008. Improving the computational morphological analysis of a Swahili corpus for lexicographic purposes. *Lexikos 18* (AFRILEX-reeks/series 18: 2008): 303–318.

Doke, C.M. and Vilakazi, B.W. 1964. *Zulu–English Dictionary*. Witwatersrand University Press, Johannesburg.

Fromkin, V., Rodman, R. and Hyams, N. 2007. *An Introduction to Language*. Thomson Heinle, Massachusetts, USA.

Hurskainen, A. 1992. A two-level formalism for the analysis of Bantu morphology: an application to Swahili. *Nordic Journal of African Studies*, 1(1):87-122.

Koskenniemi, K. 1997. Representations and finite-state components in natural language, in *Finite-State Language Processing* E. Roche and Y. Schabes (eds.), pp. 99–116. MIT Press, Boston.

Karttunen, L. 2001. Applications of finite-state trransducers in natural language processing, in *Implementation and application of automata*, S. Yu and A. Paun (eds.). Lecture Notes in Computer Science, 2088:34-46. Springer, Heidelberg.

Louw, J.A., Finlayson, R. and Satyo, S.C. 1984. *Xhosa Guide 3 for XHA100-F*. University of South Africa, Pretoria.

Meinhof, C. 1932. *Introduction to the phonology of the Bantu languages.* Dietrich Reimer/Ernst Vohsen, Berlin.

Poulos, G. and Msimang, C.T. 1998. *A linguistic analysis of Zulu*. Via Afrika, Pretoria.

Pretorius, L. and Bosch, S.E. 2002. Finite state computational morphology: Treatment of the Zulu noun. *South African Computer Journal*, 28:30-38.

Pretorius, L. and Bosch, S.E. 2003. Finite state computational morphology: An analyzer prototype for Zulu. *Machine Translation – Special issue on finite-state language resources and language processing*, 18:195-216.

The Constitution. (sa). [O]. Available: http://www.concourt.gov.za/site/theconstitution/thetext.htm. Accessed on 31 January 2008.

Yli-Jyrä, A. 2005. Toward a widely usable finite-state morphology workbench for less studied languages — Part I: Desiderata. *Nordic Journal of African Studies*, 14(4): 479 – 491.

Methods for Amharic Part-of-Speech Tagging

Björn Gambäck[†‡] **Fredrik Olsson**[†] **Atelach Alemu Argaw**[⋆] **Lars Asker**[⋆]
[†]Userware Laboratory
Swedish Institute of Computer Science
Kista, Sweden
{gamback,fredriko}@sics.se
[‡]Dpt. of Computer & Information Science
Norwegian University of Science & Technology
Trondheim, Norway
gamback@idi.ntnu.no
[⋆]Dpt. of Computer & System Sciences
Stockholm University
Kista, Sweden
{atelach,asker}@dsv.su.se

Abstract

The paper describes a set of experiments involving the application of three state-of-the-art part-of-speech taggers to Ethiopian Amharic, using three different tagsets. The taggers showed worse performance than previously reported results for English, in particular having problems with unknown words. The best results were obtained using a Maximum Entropy approach, while HMM-based and SVM-based taggers got comparable results.

1 Introduction

Many languages, especially on the African continent, are under-resourced in that they have very few computational linguistic tools or corpora (such as lexica, taggers, parsers or tree-banks) available. Here, we will concentrate on the task of developing part-of-speech taggers for Amharic, the official working language of the government of the Federal Democratic Republic of Ethiopia: Ethiopia is divided into nine regions, each with its own nationality language; however, Amharic is the language for country-wide communication.

Amharic is spoken by about 30 million people as a first or second language, making it the second most spoken Semitic language in the world (after Arabic), probably the second largest language in Ethiopia (after Oromo), and possibly one of the five largest languages on the African continent. The actual size of the Amharic speaking population must be based on estimates: Hudson (1999) analysed the Ethiopian census from 1994 and indicated that more than 40% of the population then understood Amharic, while the current size of the Ethiopian population is about 80 million.[1]

[1]82.5 million according to CIA (2009); 76.9 according to Ethiopian parliament projections in December 2008 based on the preliminary reports from the census of May 2007.

In spite of the relatively large number of speakers, Amharic is still a language for which very few computational linguistic resources have been developed, and previous efforts to create language processing tools for Amharic—e.g., Alemayehu and Willett (2002) and Fissaha (2005)—have been severely hampered by the lack of large-scale linguistic resources for the language. In contrast, the work detailed in the present paper has been able to utilize the first publicly available medium-sized tagged Amharic corpus, described in Section 5.

However, first the Amharic language as such is introduced (in Section 2), and then the task of part-of-speech tagging and some previous work in the field is described (Section 3). Section 4 details the tagging strategies used in the experiments, the results of which can be found in Section 6 together with a short discussion. Finally, Section 7 sums up the paper and points to ways in which we believe that the results can be improved in the future.

2 Amharic

Written Amharic (and Tigrinya) uses a unique script originating from the Ge'ez alphabet (the liturgical language of the Ethiopian Orthodox Church). Written Ge'ez can be traced back to at least the 4th century A.D., with the first versions including consonants only, while the characters in later versions represent consonant-vowel (CV) pairs. In modern Ethiopic script each syllograph (syllable pattern) comes in seven different forms (called orders), reflecting the seven vowel sounds. The first order is the basic form; the others are derived from it by modifications indicating vowels. There are 33 basic forms, giving 7*33 syllographs, or *fidels* ('*fidel*', lit. 'alphabet' in Amharic, refers both to the characters and the entire script). Unlike Arabic and Hebrew, Amharic is written from left to right. There is no agreed upon spelling standard for compound words and the writing system uses several ways to denote compounds

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 104–111,
Athens, Greece, 31 March 2009.

	form	pattern
root	*sbr*	CCC
perfect	*säbbär*	CVCCVC
imperfect	*säbr*	CVCC
gerund	*säbr*	CVCC
imperative	*sbär*	CCVC
causative	*assäbbär*	as-CVCCVC
passive	*täsäbbär*	täs-CVCCVC

Table 1: Some forms of the verb *sbr* ('break')

2.1 Amharic morphology

A significantly large part of the vocabulary consists of verbs, and like many other Semitic languages, Amharic has a rich verbal morphology based on triconsonantal roots with vowel variants describing modifications to, or supplementary detail and variants of the root form. For example, the root *sbr*, meaning 'to break' can have (among others!) the forms shown in Table 1. Subject, gender, number, etc., are also indicated as bound morphemes on the verb, as well as objects and possession markers, mood and tense, beneficative, malfactive, transitive, dative, negative, etc.

Amharic nouns (and adjectives) can be inflected for gender, number, definiteness, and case, although gender is usually neutral. The definite article attaches to the end of a noun, as do conjunctions, while prepositions are mostly prefixed.

2.2 Processing Amharic morphology

The first effort on Amharic morphological processing was a rule-based system for verbs (and nouns derived from verbs) which used root patterns and affixes to determine lexical and inflectional categories (Bayou, 2000), while Bayu (2002) used an unsupervised learning approach based on probabilistic models to extract stems, prefixes, and suffixes for building a morphological dictionary. The system was able to successfully analyse 87% of a small testdata set of 500 words.

The first larger-scale morphological analyser for Amharic verbs used XFST, the Xerox Finite State Tools (Fissaha and Haller, 2003). This was later extended to include all word categories (Amsalu and Gibbon, 2005). Testing with 1620 words text from an Amharic bible, 88–94% recall and 54–94% precision (depending on the word-class) were reported. The lowest precision (54%) was obtained for verbs; Amsalu and Demeke (2006) thus describe ways to extend the finite-state system to handle 6400 simple verbal stems generated from 1300 root forms.

Alemayehu and Willett (2002) report on a stemmer for Information Retrieval for Amharic, and testing on a 1221 random word sample stated "Manual assessment of the resulting stems showed that 95.5 percent of them were linguistically meaningful," but gave no evaluation of the correctness of the segmentations. Argaw and Asker (2007) created a rule-based stemmer for a similar task, and using 65 rules and machine readable dictionaries obtained 60.0% accuracy on fictional text (testing on 300 unique words) and 76.9% on news articles (on 1503 words, of which 1000 unique).[2]

3 Part-of-Speech Tagging

Part-of-speech (POS) tagging is normally treated as a classification task with the goal to assign lexical categories (word classes) to the words in a text. Most work on tagging has concentrated on English and on using supervised methods, in the sense that the taggers have been trained on an available, tagged corpus. Both rule-based and statistical / machine-learning based approaches have been thoroughly investigated. The Brill Tagger (Brill, 1995) was fundamental in using a combined rule- and learning-based strategy to achieve 96.6% accuracy on tagging the Penn Treebank version of the Wall Street Journal corpus. That is, to a level which is just about what humans normally achieve when hand-tagging a corpus, in terms of interannotator agreement—even though Voutilainen (1999) has shown that humans can get close to the 100% agreement mark if the annotators are allowed to discuss the problematic cases.

Later taggers have managed to improve Brill's figures a little bit, to just above 97% on the Wall Street Journal corpus using Hidden Markov Models, HMM and Conditional Random Fields, CRF; e.g., Collins (2002) and Toutanova et al. (2003). However, most recent work has concentrated on applying tagging strategies to other languages than English, on combining taggers, and/or on using unsupervised methods. In this section we will look at these issues in more detail, in particular with the relation to languages similar to Amharic.

3.1 Tagging Semitic languages

Diab et al. (2004) used a Support Vector Machine, SVM-based tagger, trained on the Arabic Penn

[2]Other knowledge sources for processing Amharic include, e.g., Gasser's verb stem finder (available from `nlp.amharic.org`) and wordlists as those collected by Gebremichael (`www.cs.ru.nl/~biniam/geez`).

Treebank 1 to tokenize, POS tag, and annotate Arabic base phrases. With an accuracy of 95.5% over a set of 24 tags, the data-driven tagger performed on par with state-of-the-art results for English when trained on similar-sized data (168k tokens). Bar-Haim et al. (2008) developed a lexicon-based HMM tagger for Hebrew. They report 89.6% accuracy using 21 tags and training on 36k tokens of news text. Mansour (2008) ported this tagger into Arabic by replacing the morphological analyzer, achieving an accuracy of 96.3% over 26 tags on a 89k token corpus. His approach modifies the analyses of sentences receiving a low probability by adding synthetically constructed analyses proposed by a model using character information.

A first prototype POS tagger for Amharic used a stochastic HMM to model contextual dependencies (Getachew, 2001), but was trained and tested on only one page of text. Getachew suggested a tagset for Amharic consisting of 25 tags. More recently, CRFs have been applied to segment and tag Amharic words (Fissaha, 2005), giving an accuracy of 84% for word segmentation, using character, morphological and lexical features. The best result for POS-tagging was 74.8%, when adding a dictionary and bigrams to lexical and morphological features, and 70.0% without dictionary and bigrams. The data used in the experiments was also quite small and consisted of 5 annotated news articles (1000 words). The tagset was a reduced version (10 tags) of the one used by Getachew (2001), and will be further discussed in Section 5.2.

3.2 Unsupervised tagging

The desire to use unsupervised machine learning approaches to tagging essentially originates from the wish to exploit the vast amounts of unlabelled data available when constructing taggers. The area is particularly vivid when it comes to the treatment of languages for which there exist few, if any, computational resources, and for the case of adapting an existing tagger to a new language domain.

Banko and Moore (2004) compared unsupervised HMM and transformation-based taggers trained on the same portions of the Penn Treebank, and showed that the quality of the lexicon used for training had a high impact on the tagging results. Duh and Kirchhoff (2005) presented a minimally-supervised approach to tagging for dialectal Arabic (Colloquial Egyptian), based on a morphological analyzer for Modern Standard Arabic and unlabeled texts in a number of dialects. Using a trigram HMM tagger, they first produced a baseline system and then gradually improved on that in an unsupervised manner by adding features so as to facilitate the analysis of unknown words, and by constraining and refining the lexicon.

Unsupervised learning is often casted as the problem of finding (hidden) structure in unlabeled data. Goldwater and Griffiths (2007) noted that most recent approaches to this problem aim to identify the set of attributes that maximizes some target function (Maximum Likelihood Estimation), and then to select the values of these attributes based on the representation of the model. They proposed a different approach, based on Bayesian principles, which tries to directly maximize the probability of the attributes based on observation in the data. This Bayesian approach outperformed Maximum Likelihood Estimation when training a trigram HMM tagger for English. Toutanova and Johnson (2007) report state-of-the-art results by extending the work on Bayesian modelling for unsupervised learning of taggers both in the way that prior knowledge can be incorporated into the model, and in the way that possible tags for a given word is explicitly modeled.

3.3 Combining taggers

A possible way to improve on POS tagging results is to combine the output of several different taggers into a committee, forming joint decisions regarding the labeling of the input. Roughly, there are three obvious ways of combining multiple predicted tags for a word: random decision, voting, and stacking (Dietterich, 1997), with the first way suited only for forming a baseline. *Voting* can be divided into two subclasses: unweighted votes, and weighted votes. The weights of the votes, if any, are usually calculated based on the classifiers' performance on some initial dataset. *Stacking*, finally, is a way of combining the decisions made by individual taggers in which the predicted tags for a given word are used as input to a subsequent tagger which outputs a final label for the word.

Committee-based approaches to POS tagging have been in focus the last decade: Brill and Wu (1998) combined four different taggers for English using unweighted voting and by exploring contextual cues (essentially a variant of stacking). Aires et al. (2000) experimented with 12 different ways of combining the output from taggers for Brazilian

Portuguese, and concluded that some, but not all, combinations yielded better accuracy than the best individual tagger. Shacham and Wintner (2007) contrasted what they refer to as being a naïve way of combining taggers with a more elaborate, hierarchical one for Hebrew. In the end, the elaborated method yielded results inferior to the naïve approach. De Pauw et al. (2006) came to similar conclusions when using five different ways of combining four data-driven taggers for Swahili. The taggers were based on HMM, Memory-based learning, SVM, and Maximum Entropy, with the latter proving most accurate. Only in three of five cases did a combination of classifiers perform better than the Maximum Entropy-based tagger, and simpler combination methods mostly outperformed more elaborate ones.

Spoustová et al. (2007) report on work on combining a hand-written rule-based tagger with three statistically induced taggers for Czech. As an effect of Czech being highly inflectional, the tagsets are large: 1000–2000 unique tags. Thus the approach to combining taggers first aims at reducing the number of plausible tags for a word by using the rule-based tagger to discard impossible tags. Precision is then increased by invoking one or all of the data-driven taggers. Three different ways of combining the taggers were explored: serial combination, involving one of the statistical taggers; so called SUBPOS pre-processing, involving two instances of statistical taggers (possibly the same tagger); and, parallel combination, in which an arbitrary number of statistical taggers is used. The combined tagger yielded the best results for Czech POS tagging reported to date, and as a side-effect also the best accuracy for English: 97.43%.[3]

4 The Taggers

This section describes the three taggers used in the experiments (which are reported on in Section 6).

4.1 Hidden Markov Models: TnT

TnT, "Trigrams'n'Tags" (Brants, 2000) is a very fast and easy-to-use HMM-based tagger which painlessly can be trained on different languages and tagsets, given a tagged corpus.[4] A Markov-based tagger aims to find a tag sequence which maximizes $P(word_n|tag_n) * P(tag_n|tag_{1...n-1})$, where the first factor is the emit (or lexical) probability, the likelihood of a word given certain tag, and the second factor is the state transition (or contextual) probability, the likelihood of a tag given a sequence of preceding tags. TnT uses the Viterbi algorithm for finding the optimal tag sequence. Smoothing is implemented by linear interpolation, the respective weights are determined by deleted interpolation. Unknown words are handled by a suffix trie and successive abstraction.

Applying TnT to the Wall Street Journal corpus, Brants (2000) reports 96.7% overall accuracy, with 97.0% on known and 85.5% on unknown words (with 2.9% of the words being unknown).

4.2 Support Vector Machines: SVMTool

Support Vector Machines (SVM) is a linear learning system which builds two class classifiers. It is a supervised learning method whereby the input data are represented as vectors in a high-dimensional space and SVM finds a hyperplane (a decision boundary) separating the input space into two by maximizing the margin between positive and negative data points.

SVMTool is an open source tagger based on SVMs.[5] Comparing the accuracy of SVMTool with TnT on the Wall Street Journal corpus, Giménez and Màrquez (2004) report a better performance by SVMTool: 96.9%, with 97.2% on known words and 83.5% on unknown.

4.3 Maximum Entropy: MALLET

Maximum Entropy is a linear classification method. In its basic incarnation, linear classification combines, by addition, the pre-determined weights used for representing the importance of each feature to a given class. Training a Maximum Entropy classifier involves fitting the weights of each feature value for a particular class to the available training data. A good fit of the weights to the data is obtained by selecting weights to maximize the log-likelihood of the learned classification model. Using an Maximum Entropy approach to POS tagging, Ratnaparkhi (1996) reports a tagging accuracy of 96.6% on the Wall Street Journal.

The software of choice for the experiments reported here is MALLET (McCallum, 2002), a freely available Java implementation of a range of machine learning methods, such as Naïve Bayes, decision trees, CRF, and Maximum Entropy.[6]

[3] As reported on `ufal.mff.cuni.cz/compost/en`

[4] `www.coli.uni-saarland.de/~thorsten/tnt`

[5] `www.lsi.upc.edu/~nlp/SVMTool`

[6] `mallet.cs.umass.edu`

5 The Dataset

The experiments of this paper utilize the first medium-sized corpus for Amharic (available at `http://nlp.amharic.org`). The corpus consists of all 1065 news texts (210,000 words) from the Ethiopian year 1994 (parts of the Gregorian years 2001–2002) from the Walta Information Center, a private news service based in Addis Ababa. It has been morphologically analysed and manually part-of-speech tagged by staff at ELRC, the Ethiopian Languages Research Center at Addis Ababa University (Demeke and Getachew, 2006).

The corpus is available both in *fidel* and transcribed into a romanized version known as SERA, System for Ethiopic Representation in ASCII (Yacob, 1997). We worked with the transliterated form (202,671 words), to be compatible with the machine learning tools used in the experiments.

5.1 "Cleaning" the corpus

Unfortunately, the corpus available on the net contains quite a few errors and tagging inconsistencies: nine persons participated in the manual tagging, writing the tags with pen on hard copies, which were given to typists for insertion into the electronic version of the corpus—a procedure obviously introducing several possible error sources.

Before running the experiments the corpus had to be "cleaned": many non-tagged items have been tagged (the human taggers have, e.g., often tagged the headlines of the news texts as one item, end-of-sentence punctuation), while some double tags have been removed. Reflecting the segmentation of the original Amharic text, all whitespaces were removed, merging multiword units with a single tag into one-word units. Items like '`"`' and '`/`' have been treated consistently as punctuation, and consistent tagging has been added to word-initial and word-final hyphens. Also, some direct tagging errors and misspellings have been corrected.

Time expressions and numbers have not been consistently tagged at all, but those had to be left as they were. Finally, many words have been transcribed into SERA in several versions, with only the cases differing. However, this is also difficult to account for (and in the experiments below we used the case sensitive version of SERA), since the SERA notation in general lets upper and lower cases of the English alphabet represent different symbols in *fidel* (the Amharic script).

5.2 Tagsets

For the experiments, three different tagsets were used. Firstly, the full, original 30-tag set developed at the Ethiopian Languages Research Center and described by Demeke and Getachew (2006). This version of the corpus will be referred to as 'ELRC'. It contains $200,863$ words and differs from the published corpus in way of the corrections described in the previous section.

Secondly, the corpus was mapped to 11 basic tags. This set consists of ten word classes: Noun, Pronoun, Verb, Adjective, Preposition, Conjunction, Adverb, Numeral, Interjection, and Punctuation, plus one tag for problematic words (unclear: `<UNC>`). The main differences between the two tagsets pertain to the treatment of prepositions and conjunctions: in 'ELRC' there are specific classes for, e.g., pronouns attached with preposition, conjunction, and both preposition and conjunction (similar classes occur for nouns, verbs, adjectives, and numerals). In addition, numerals are divided into cardinals and ordinals, verbal nouns are separated from other nouns, while auxiliaries and relative verbs are distinguished from other verbs. The full tagset is made up of thirty subclasses of the basic classes, based on type of word only: the tags contain no information on grammatical categories (such as number, gender, tense, and aspect).

Thirdly, for comparison reasons, the full tagset was mapped to the 10 tags used by Fissaha (2005). These classes include one for Residual (`R`) which was assumed to be equivalent to `<UNC>`. In addition, `<CONJ>` and `<PREP>` were mapped to Adposition (`AP`), and both `<N>` and `<PRON>` to `N`. The other mappings were straight-forward, except that the 'BASIC' tagset groups all verbs together, while Fissaha kept Auxiliary (`AUX`) as its own class. This tagset will be referred to as 'SISAY'.

5.3 Folds

For evaluation of the taggers, the corpus was split into 10 folds. These folds were created by chopping the corpus into 100 pieces, each of about 2000 words in sequence, while making sure that each piece contained full sentences (rather than cutting off the text in the middle of a sentence), and then merging sets of 10 pieces into a fold. Thus the folds represent even splits over the corpus, to avoid tagging inconsistencies, but the sequences are still large enough to potentially make knowledge sources such as n-grams useful.

Fold	TOTAL	KNOWN	UNKNOWN
fold00	20,027	17,720	2,307
fold01	20,123	17,750	2,373
fold02	20,054	17,645	2,409
fold03	20,169	17,805	2,364
fold04	20,051	17,524	2,527
fold05	20,058	17,882	2,176
fold06	20,111	17,707	2,404
fold07	20,112	17,746	2,366
fold08	20,015	17,765	2,250
fold09	20,143	17,727	2,416
Average	20,086	17,727	2,359
Percent	—	88.26	11.74

Table 2: Statistics for the 10 folds

Table 2 shows the data for each of the folds, in terms of total number of tokens, as well as split into known and unknown tokens, where the term UNKNOWN refers to tokens that are not in any of *the other* nine folds. The figures at the bottom of the table show the average numbers of known and unknown words, over all folds. Notably, the average number of unknown words is about four times higher than in the Wall Street Journal corpus (which, however, is about six times larger).

6 Results

The results obtained by applying the three different tagging strategies to the three tagsets are shown in Table 3, in terms of average accuracies after 10-fold cross validation, over all the tokens (with standard deviation),[7] as well as accuracy divided between the known and unknown words. Additionally, SVMTool and MALLET include support for automatically running 10-fold cross validation on their own folds. Figures for those runs are also given. The last line of the table shows the baselines for the tagsets, given as the number of tokens tagged as regular nouns divided by the total number of words after correction.

6.1 TnT

As the bold face figures indicate, TnT achieves the best scores of all three taggers, on all three tagsets, on *known* words. However, it has problems with the unknown words—and since these are so frequent in the corpus, TnT overall performs worse than the other taggers. The problems with the unknown words increase as the number of possible tags increase, and thus TnT does badly on the original tagging scheme ('ELRC'), where it only gets

[7]The standard deviation is given by $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2}$ where $\overline{x}$ is the arithmetic mean ($\frac{1}{n}\sum_{i=1}^{n} x_i$).

	ELRC	BASIC	SISAY
TnT	85.56	92.55	92.60
STD DEV	0.42	0.31	0.32
KNOWN	**90.00**	**93.95**	**93.99**
UNKNOWN	52.13	82.06	82.20
SVM	**88.30**	**92.77**	**92.80**
STD DEV	0.41	0.31	0.37
KNOWN	89.58	93.37	93.34
UNKNOWN	**78.68**	**88.23**	**88.74**
Own folds	88.69	92.97	92.99
STD DEV	0.33	0.17	0.26
MaxEnt	87.87	92.56	92.60
STD DEV	0.49	0.38	0.43
KNOWN	89.44	93.26	93.27
UNKNOWN	76.05	87.29	87.61
Own folds	**90.83**	**94.64**	**94.52**
STD DEV	1.37	1.11	0.69
BASELINE	35.50	58.26	59.61

Table 3: Tagging results

a bit over 50% on the unknown words (and 85.6% overall). For the two reduced tagsets TnT does better: overall performance goes up to a bit over 92%, with 82% on unknown words.

Table 3 shows the results on the default configuration of TnT, i.e., using 3-grams and interpolated smoothing. Changing these settings give no substantial improvement overall: what is gained at one end (e.g., on unknown words or a particular tagset) is lost at the other end (on known words or other tagsets). However, per default TnT uses a suffix trie of length 10 to handle unknown words. Extending the suffix to 20 (the maximum value in TnT) gave a slight performance increase on 'ELCR' (0.13% on unknown words, 0.01% overall), while having no effect on the smaller tagsets.

6.2 SVM

The SVM-tagger outperforms TnT on unknown words, but is a bit worse on known words. Overall, SVM is slightly better than TnT on the two smaller tagsets and clearly better on the large tagset, and somewhat better than MaxEnt on all three tagsets.

These results are based on SVMTool's default parameters: a one-pass, left-to-right, greedy tagging scheme with a window size of 5. Previous experiments with parameter tuning and multiple pass tagging have indicated that there is room for performance improvements by $\approx 2\%$.

6.3 Maximum Entropy

The MaxEnt tagger gets results comparable to the other taggers on the predefined folds. Its overall

Word$_n$; Tag of *Word*$_n$
Prefixes of *Word*$_n$, length 1-5 characters
Postfixes of *Word*$_n$, length 1-5 characters
Is *Word*$_n$ capitalized?
Is *Word*$_n$ all digits?
Does *Word*$_n$ contain digits?
Does *Word*$_n$ contain a hyphen?
Word$_{n-1}$; Tag of *Word*$_{n-1}$
Word$_{n-2}$; Tag of *Word*$_{n-2}$
Word$_{n+1}$
Word$_{n+2}$

Table 4: Features used in the MaxEnt tagger

performance is equivalent to TnT's on the smaller tagsets, but significantly better on 'ELRC'.

As can be seen in Table 3, the MaxEnt tagger clearly outperforms the other taggers on all tagsets, when MALLET is allowed to create its own folds: all tagsets achieved classification accuracies higher than 90%, with the two smaller tagsets over 94.5%. The dramatic increase in the tagger's performance on these folds is surprising, but a clear indication of one of the problems with n-fold cross validation: even though the results represent averages after n runs, the choice of the original folds to suit a particular tagging strategy is of utmost importance for the final result.

Table 4 shows the 22 features used to represent an instance (*Word*$_n$) in the Maximum Entropy tagger. The features are calculated per token within sentences: the starting token of a sentence is not affected by the characteristics of the tokens ending the previous sentence, nor the other way around. Thus not all features are calculated for all tokens.

6.4 Discussion

In terms of accuracy, the MaxEnt tagger is by far the best of the three taggers, and on all three tagsets, when allowed to select its own folds. Still, as Table 3 shows, the variation of the results for each individual fold was then substantially larger.

It should also be noted that TnT is by far the fastest of the three taggers, in all respects: in terms of time to set up and learn to use the tagger, in terms of tagging speed, and in particular in terms of training time. Training TnT is a matter of seconds, but a matter of hours for MALLET/MaxEnt and SVMTool. On the practical side, it is worth adding that TnT is robust, well-documented, and easy to use, while MALLET and SVMTool are substantially more demanding in terms of user effort and also appear to be more sensitive to the quality and format of the input data.

7 Conclusions and Future Work

The paper has described experiments with applying three state-of-the-art part-of-speech taggers to Amharic, using three different tagsets. All taggers showed worse performance than previously reported results for English. The best accuracy was obtained using a Maximum Entropy approach when allowed to create its own folds: 90.1% on a 30 tag tagset, and 94.6 resp. 94.5% on two reduced sets (11 resp. 10 tags), outperforming an HMM-based (TnT) and an SVM-based (SVMTool) tagger. On predefined folds all taggers got comparable results (92.5-92.8% on the reduced sets and 4-7% lower on the full tagset). The SVM-tagger performs slightly better than the others overall, since it has the best performance on unknown words, which are four times as frequent in the 200K words Amharic corpus used than in the (six times larger) English Wall Street Journal corpus. TnT gave the best results for known words, but had the worst performance on unknown words.

In order to improve tagging accuracy, we will investigate including explicit morphological processing to treat unknown words, and combining taggers. Judging from previous efforts on combining taggers (Section 3.3), it is far from certain that the combination of taggers actually ends up producing better results than the best individual tagger. A pre-requisite for successful combination is that the taggers are sufficiently dissimilar; they must draw on different characteristics of the training data and make different types of mistakes.

The taggers described in this paper use no other knowledge source than a tagged training corpus. In addition to incorporating (partial) morphological processing, performance could be increased by including knowledge sources such as machine readable dictionaries or lists of Amharic stem forms (Section 2.2). Conversely, semi-supervised or unsupervised learning for tagging clearly are interesting alternatives to manually annotate and construct corpora for training taggers. Since there are few computational resources available for Amharic, approaches as those briefly outlined in Section 3.2 deserve to be explored.

Acknowledgements

The work was partially funded by Sida, the Swedish International Development Cooperation Agency through SPIDER (the Swedish Programme for ICT in Developing Regions).

Thanks to Dr. Girma Demeke, Mesfin Getachew, and the ELRC staff for their efforts on tagging the corpus, and to Thorsten Brants for providing us with the TnT tagger.

References

Rachel V. Xavier Aires, Sandra M. Aluísio, Denise C. S. Kuhn, Marcio L. B. Andreeta, and Osvaldo N. Oliveira Jr. 2000. Combining classifiers to improve part of speech tagging: A case study for Brazilian Portuguese. In *15th Brazilian Symposium on AI*, pp. 227–236, Atibaia, Brazil.

Nega Alemayehu and Peter Willett. 2002. Stemming of Amharic words for information retrieval. *Literary and Linguistic Computing*, 17:1–17.

Saba Amsalu and Dafydd Gibbon. 2005. Finite state morphology of Amharic. In *5th Recent Advances in Natural Language Processing*, pp. 47–51, Borovets, Bulgaria.

Saba Amsalu and Girma A. Demeke. 2006. Non-concatinative finite-state morphotactics of Amharic simple verbs. *ELRC Working Papers*, 2:304-325.

Atelach Alemu Argaw and Lars Asker. 2007. An Amharic stemmer: Reducing words to their citation forms. *Computational Approaches to Semitic Languages*, pp. 104–110, Prague, Czech Rep.

Michele Banko and Robert C. Moore. 2004. Part of speech tagging in context. In *20th Int. Conf. on Computational Linguistics*, pp. 556–561, Geneva, Switzerland.

Roy Bar-Haim, Khalil Simaan, and Yoad Winter. 2008. Part-of-speech tagging of modern Hebrew text. *Natural Language Engineering*, 14:223–251.

Abiyot Bayou. 2000. Design and development of word parser for Amharic language. MSc Thesis, Addis Ababa University, Ethiopia.

Tesfaye Bayu. 2002. Automatic morphological analyser: An experiment using unsupervised and autosegmental approach. MSc Thesis, Addis Ababa University, Ethiopia.

Thorsten Brants. 2000. TnT — a statistical part-of-speech tagger. In *6th Conf. Applied Natural Language Processing*, pp. 224–231, Seattle, Wash.

Eric Brill and Jun Wu. 1998. Classifier combination for improved lexical disambiguation. In *17th Int. Conf. on Computational Linguistics*, pp. 191–195, Montreal, Canada.

Eric Brill. 1995. Transformation-based error-driven learning and Natural Language Processing: A case study in part of speech tagging. *Computational Linguistics*, 21:543–565.

CIA. 2009. *The World Factbook — Ethiopia*. The Central Intelligence Agency, Washington, DC. [Updated 22/01/09.]

Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Empirical Methods in Natural Language Processing*, pp. 1–8, Philadelphia, Penn.

Girma A. Demeke and Mesfin Getachew. 2006. Manual annotation of Amharic news items with part-of-speech tags and its challenges. *ELRC Working Papers*, 2:1–17.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic tagging of Arabic text: From raw text to base phrase chunks. In *HLT Conf. North American ACL*, pp. 149–152, Boston, Mass.

Thomas G. Dietterich. 1997. Machine-learning research: Four current directions. *AI magazine*, 18:97–136.

Kevin Duh and Katrin Kirchhoff. 2005. POS tagging of dialectal Arabic: A minimally supervised approach. *Computational Approaches to Semitic Languages*, pp. 55–62, Ann Arbor, Mich.

Sisay Fissaha and Johann Haller. 2003. Amharic verb lexicon in the context of machine translation. In *10th Traitement Automatique des Langues Naturelles*, vol. 2, pp. 183–192, Batz-sur-Mer, France.

Sisay Fissaha. 2005. Part of speech tagging for Amharic using conditional random fields. *Computational Approaches to Semitic Languages*, pp. 47–54, Ann Arbor, Mich.

Mesfin Getachew. 2001. Automatic part of speech tagging for Amharic: An experiment using stochastic hidden Markov model (HMM) approach. MSc Thesis, Addis Ababa University, Ethiopia.

Jesús Giménez and Lluís Màrquez. 2004. SVMTool: A general POS tagger generator based on support vector machines. In *4th Int. Conf. Language Resources and Evaluation*, pp. 168–176, Lisbon, Portugal.

Sharon Goldwater and Thomas L. Griffiths. 2007. A fully Bayesian approach to unsupervised part-of-speech tagging. In *45th ACL*, pp. 744–751, Prague, Czech Rep.

Grover Hudson. 1999. Linguistic analysis of the 1994 Ethiopian census. *Northeast African Studies*, 6:89–107.

Saib Mansour. 2008. Combining character and morpheme based models for part-of-speech tagging of Semitic languages. MSc Thesis, Technion, Haifa, Israel.

Andrew Kachites McCallum. 2002. MALLET: A machine learning for language toolkit. Webpage.

Guy De Pauw, Gilles-Maurice de Schryver, and Peter W. Wagacha. 2006. Data-driven part-of-speech tagging of Kiswahili. In *9th Int. Conf. Text, Speech and Dialogue*, pp. 197–204, Brno, Czech Rep.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Empirical Methods in Natural Language Processing*, pp. 133–142, Philadelphia, Penn.

Danny Shacham and Shuly Wintner. 2007. Morphological disambiguation of Hebrew: A case study in classifier combination. In *Empirical Methods in Natural Language Processing*, pp. 439–447, Prague, Czech Rep.

Drahomirá Spoustová, Jan Hajič, Jan Votrubec, Pavel Krbec, and Pavel Květoň. 2007. The best of two worlds: Cooperation of statistical and rule-based taggers for Czech. *Balto-Slavonic Natural Language Processing*, pp. 67–74. Prague, Czech Rep.

Kristina Toutanova and Mark Johnson. 2007. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *21st Int. Conf. Advances in Neural Information Processing Systems*, pp. 1521–1528, Vancover, B.C.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT Conf. North American ACL*, pp. 173–180, Edmonton, Alberta.

Atro Voutilainen. 1999. An experiment on the upper bound of interjudge agreement: The case of tagging. In *9th European ACL*, pp. 204–208, Bergen, Norway.

Daniel Yacob. 1997. The System for Ethiopic Representation in ASCII — 1997 standard. Webpage.

An Ontology for Accessing Transcription Systems (OATS)

Steven Moran
University of Washington
Seattle, WA, USA
stiv@u.washington.edu

Abstract

This paper presents the Ontology for Accessing Transcription Systems (OATS), a knowledge base that supports interoperation over disparate transcription systems and practical orthographies. The knowledge base includes an ontological description of writing systems and relations for mapping transcription system segments to an interlingua pivot, the IPA. It includes orthographic and phonemic inventories from 203 African languages. OATS is motivated by the desire to query data in the knowledge base via IPA or native orthography, and for error checking of digitized data and conversion between transcription systems. The model in this paper implements these goals.

1 Introduction

The World Wide Web has emerged as the predominate source for obtaining linguistic field data and language documentation in textual, audio and video formats. A simple keyword search on the nearly extinct language Livonian [liv][1] returns numerous results that include text, audio and video files. As data on the Web continue to increase, including material posted by native language communities, researchers are presented with an ideal medium for the automated discovery and analysis of linguistic data, e.g. (Lewis, 2006). However, resources on the Web are not always accessible to users or software agents. The data often exist in legacy or proprietary software and data formats. This makes them difficult to locate and access.

Interoperability of linguistic resources has the ability to make disparate linguistic data accessible to researchers. It is also beneficial for data aggregation. Through the use of ontologies, applications can be written to perform intelligent search (deriving implicit knowledge from explicit information). They can also interoperate between resources, thus allowing data to be shared across applications and between research communities with different terminologies, annotations, and notations for marking up data.

OATS is a knowledge base, i.e. a data source that uses an ontology to specify the structure of entities and their relations. It includes general knowledge of writing systems and transcription systems that are core to the General Ontology of Linguistic Description (GOLD)[2] (Farrar and Langendoen, 2003). Other portions of OATS, including the relationships encoded for relating segments of transcription systems, or the computational representations of these elements, extend GOLD as a Community of Practice Extension (COPE) (Farrar and Lewis, 2006). OATS provides interoperability for transcription systems and practical orthographies that map phones and phonemes in unique relationships to their graphemic representations. These systematic mappings thus provide a computationally tractable starting point for interoperating over linguistic texts. The resources that are targeted also encompass a wide array of data on lesser-studied languages of the world, as well as *low density* languages, i.e. those with few electronic resources (Baldwin et al., 2006).

This paper is structured as follows: in section 2, linguistic and technological definitions and terminology are provided. In section 3, the theoretical and technological challenges of interoperating over heterogeneous transcriptions systems are described. The technologies used in OATS and its design are presented in section 4. In section 5, OATS' implementation is illustrated with linguistic data that was mined from the Web, therefore motivating the general design objectives taken into

[1] ISO 639-3 language codes are in [].

[2] http://linguistics-ontology.org/

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 112–120,
Athens, Greece, 31 March 2009.

account in its development. Section 6 concludes with future research goals.

2 Conventions and Terminology

2.1 Conventions

Standard conventions are used for distinguishing between graphemic < >, phonemic / / and phonetic representations [ ].[3] For character data information, I follow the Unicode Standard's notational conventions (The Unicode Consortium, 2007). Character names are represented in small capital letters (e.g. LATIN SMALL LETTER SCHWA) and code points are expressed as 'U+*n*' where *n* is a four to six digit hexadecimal number (e.g. U+0256), which is rendered as <ɘ>.

2.2 Linguistic definitions

In the context of this paper, a **transcription system** is a system of symbols and rules for graphically transcribing the sounds of a language variety. A **practical orthography** is a phonemic writing system designed for practical use by speakers already competent in the language. The mapping relation between phonemes and graphemes in practical orthographies is purposely shallow, i.e. there is a faithful mapping from a unique sound to a unique symbol.[4] The IPA is often used by field linguists in the development of practical orthographies for languages without writing systems. An **orthography** specifies the symbols, punctuation, and the rules in which a language is correctly written in a standardized way. All orthographies are language specific.

Practical orthographies and transcription systems are both kinds of **writing systems**. A writing system is a symbolic system that uses visible or tactile signs to represent a language in a systematic way. Differences in the encoding of meaning and sound form a continuum for representing writing systems in a typology whose categories are commonly referred to as either logographic, syllabic, phonetic or featural. A logographic system denotes symbols that visually represent morphemes (and sometimes morphemes and syllables). A syllabic system uses symbols to denote syllables. A phonetic system represents sound segments as symbols. Featural systems are less common and encode phonological features within the shapes of the symbols represented in the script.

The term **script** refers to a collection of symbols (or distinct marks) as employed by a writing system. The term script is confused with and often used interchangeably with 'writing system'. A writing system may be written with different scripts, e.g. the alphabet writing system can be written in Roman and Cyrillic scripts (Coulmas, 1999). A **grapheme** is the unit of writing that represents a particular abstract representation of a symbol employed by a writing system. Like the phoneme is an abstract representation of a distinct sound in a language, a grapheme is a contrastive graphical unit in a writing system. A grapheme is the basic, minimally distinctive symbol of a writing system. A script may employ multiple graphemes to represent a single phoneme, e.g. the graphemes <c> and <h> when conjoined in English represent one phoneme in English, <ch> pronounced /ʧ/ (or /k/). The opposite is also found in writing systems, where a single grapheme represents two or more phonemes, e.g. <x> in English is a combination of the phonemes /ks/.

A **graph** is the smallest unit of written language (Coulmas, 1999). The electronic counterpart of the graph is the glyph. **Glyphs** represent the variation of graphemes as they appear when rendered or displayed. In typography glyphs are created using different illustration techniques. These may result in **homoglyphs**, pairs of characters with shapes that are either identical or are beyond differentiation by swift visual inspection. When rendered by hand, a writer may use different styles of handwriting to produce glyphs in standard handwriting, cursive, or calligraphy. When rendered computationally, a repertoire of glyphs makes up a **font**.

A final distinction is needed for interoperating over transcription systems. The term **scripteme** is used for the use of a grapheme within a writing system with the particular semantics (i.e., pronunciation) it is assigned within that writing system. The notion scripteme is needed because graphemes may be homoglyphic across scripts and languages, and the semantics of a grapheme is dependent on the writing system using it. For example, the grapheme <p> in Russian represents a dental or alveolar trill; /r/ in IPA. However, <p> is realized by English speakers as a voiceless bilabial stop /p/. The defining of scripteme is necessary

[3] Phonemic and phonetic representations are given in the International Phonetic Alphabet (IPA).

[4] Practical orthographies are intended to jump-start written materials development by correlating a writing system with its sound units, making it easier for speakers to master and acquire literacy.

for interoperability because it provides a level for mapping a writing system specific grapheme to the phonological level, allowing the same grapheme to represent different sounds across different transcription and writing systems.

2.3 Technological definitions

A **document** refers to an electronic document that contains language data. Each document is associated with metadata and one or more transcription systems or practical orthographies. A document's content is comprised of a set scriptemes from its transcription system. A **mapping relation** is an unordered pair of a scripteme in a transcription system and its representation in IPA.

OATS first maps scriptemes to their grapheme equivalent(s). Graphemes are then mapped to their character equivalents. A **character** in OATS is a computational representation of a grapheme. **Character encodings** represent a range of integers known as the **code space**. A **code point** is a unique integer, or point, within this code space. An abstract character is then mapped to a unique code point and rendered as an **encoded character** and typographically defined by the font used to render it. A set of encoded characters is a **character set** and different character encodings encode characters as numbers via different encoding schemes.

3 Interoperating Over Transcription Systems

Section 3.1 uses the Sisaala languages to illustrate interoperability challenges posed by linguistic data. Section 3.2 addresses technological issues including encoding and ambiguity.

3.1 Linguistic challenges

Three genetically related languages spoken in Northern Ghana, Sisaala Pasaale [sig], Sisaala Tumulung [sil] and Sisaala Western [ssl], differ slightly in their orthographies for two reasons: they have slightly divergent phonemic inventories and their orthographies may differ graphemically when representing the same phoneme. See Table 1.

Table 1: Phoneme-to-grapheme relations

	/kp/	**d**	**/ʧ/**	**/ɪ/**	**/ʊ/**	**Tone**
sig	kp	d, r	ky	ɪ	ʊ	not marked
sil	kp	d	ch	i	u	accents
ssl	-	d	ky	ɪ	ʊ	accents

The voiceless labial-velar phoneme /kp/ appears in both Sisaala Tumulung and Sisaala Pasaale, but has been lost in Sisaala Western. There is a convergence of the allophones [d] and [r] into one phoneme /d/ in Sisaala Pasaale (Toupin, 1995).[5]

These three orthographies also differ because of their authors' choices in assigning graphemes to phonemes. In Sisaala Pasaale and Sisaala Western, the phonemes /ʧ/ and /ʤ/ are written as <ky> and <gy>. In Sisaala Tumulung, however, these sounds are written <ch> and <j>. Orthography developers may have made these choices for practical reasons, such as ease of learnability or technological limitations (Bodomo, 1997). During the development of practical orthographies for Sisaala Pasaale and Sisaala Western, the digraphs <ky> and <gy> were chosen because children learn Dagaare [dga] in schools, so they are already familiar with their sounds in the Dagaare orthography (Mcgill et al., 1999) (Moran, 2008).

Another difference lies in the representation of vowels. Both Sisaala Pasaale and Sisaala Western represent their full sets of vowels orthographically. These orthographies were developed relatively recently, when computers, character encodings, and font support, have become less problematic. In Sisaala Tumulung, however, the phonemes /i/ and /ɪ/ are collapsed to <i>, and /u/ and /ʊ/ to <u> (Blass, 1975). Sisaala Tumulung's orthography was developed in the 1970s and technological limitations may have led its developers to collapse these phonemes in the writing system. For example, the Ghana Alphabet Committee's 1990 Report lacks an individual grapheme <ŋ> for the phoneme /ŋ/ for Dagaare. This difficulty of rendering unconventional symbols on typewriters once posed a challenge for orthography development (Bodomo, 1997).

Tone is both lexically and grammatically contrastive in Sisaala languages. In Sisaala Pasaale's official orthography tone is not marked and is not used in native speaker materials. On the other hand, in linguistic descriptions that use this orthography, tone is marked to disambiguate tonal

[5]The phoneme /d/ has morphologically conditioned allographs <d> (word initial) or <r> (elsewhere) (McGill, 2004).

minimal pairs in lexical items and grammatical constructions (McGill, 2004). In the Sisaala (Tumulung)-English dictionary, tone is marked only to disambiguate lexical items (Blass, 1975). In linguistic descriptions of Sisaala Western, non-contrastive tone is marked. When tone is marked, it appears as acute (high tone) and grave (low tone) accents over vowels or nasals.

Language researchers would quickly pick up on these minute differences in orthographies. However, what first seem to be trivial differences, illustrate one issue of resource discovery on the Web – without methods for interoperability, even slightly divergent resources are more difficult to discover, query and compare. How would someone researching a comparative analysis of /ʧ/ sounds of languages in Northern Ghana discover that it is represented as <ky> and <ch> without first locating the extremely sparse grammatical information available on these languages? Furthermore, automatic phonetic research is possible on languages with shallow orthographies (Zuraw, 2006), but crosslinguistic versions of such work require interoperation over writing systems.

3.2 Technological challenges

The main technological challenges in interoperating over textual electronic resources are: encoding multilingual language text in an interoperable format and resolving ambiguity between mapping relations. These are addressed below.

Hundreds of character encoding sets for writing systems have been developed, e.g. ASCII, GB 18030[6] and Unicode. Historically, different standards were formalized differently and for different purposes by different standards committees. A lack of interoperability between character encodings ensued. Linguists, restricted to standard character sets that lacked IPA support and other language-specific graphemes that they needed, made their own solutions (Bird and Simons, 2003). Some chose to represent unavailable graphemes with substitutes, e.g. the combination of <ng> to represent <ŋ>. Others redefined selected characters from a character encoding to map their own fonts to. One linguist's redefined character set, however, would not render properly on another linguist's computer if they did not share the same font. If two character encodings defined two character sets differently, then data could not be reliably and correctly displayed.

To circumvent these problems, OATS uses the Unicode Standard[7] for multilingual character encoding of electronic textual data. Unicode encodes 76 scripts and includes the IPA.[8] In principle this allows OATS to interoperate over IPA and all scripts currently encoded in Unicode. However, writing systems, scripts and transcriptions are often themselves encoded ambiguously.

Unicode encodes characters, not glyphs, in scripts and sometimes unifies duplicate characters across scripts. For example, IPA characters of Greek and Latin origin, such as <β> and <k> are not given a distinct position within Unicode's IPA character block. The Unicode code space is subdivided into character blocks, which generally encode characters from a single script, but as is illustrated by the IPA, characters may be dispersed across several different character blocks. This poses a challenge for interoperation, particularly with regard to homographs. Why shouldn't a speaker of Russian use the <а> CYRILLIC SMALL LETTER A at code point U+0430 for IPA transcription, instead of <a> LATIN SMALL LETTER A at code point U+0061, when visually they are indistinguishable?

Homoglyphs come in two flavors: linguistic and non-linguistic. Linguists are unlikely to distinguish between the <ə> LATIN SMALL LETTER SCHWA at code point U+0259 and <ə> LATIN SMALL LETTER TURNED E at U+01DD. And non-linguists are unlikely to differentiate any semantic difference between an open back unrounded vowel <ɑ>, the LATIN SMALL LETTER ALPHA at U+0251, and the open front unrounded vowel <a>, LATIN SMALL LETTER A at U+0061.

Another challenge is how to handle ambiguity in transcription systems and orthographies. In Serbo-Croatian, for example, the digraphs <lj>, <nj> and <dz> represent distinct phonemes and each are comprised of two graphemes, which themselves represent distinct phonemes. Words like <nadzivjeti> 'to outlive' are composed of the morphemes <nad>, a prefix, and the verb <zivjeti>. In this instance the combination of <d> and <z> does not represent a single digraph <dz>; they represent two neighboring phonemes across a morpheme boundary. Likewise in En-

[6] Guójiā Biāozhǔn, the national standard character set for the People's Republic of China

[7] ISO/IEC 1064

[8] http://www.unicode.org/Public/UNIDATA/Scripts.txt

glish, the grapheme sequence <sh> can be both a digraph as well as a sequence of graphemes, as in <mishmash> and <mishap>. When parsing words like <mishit> and <mishear> both disambiguations are theoretically available. Another example is illustrated by <h>, <t>, and <th>. How should <t> be interpreted before <h> when English gives us both /tɔməs/ 'Thomas' and /θioudor/ 'Theodore'? The Sisaala Western word <niikyuru> 'waterfall' could be parsed as /niik.yuru/ instead of /nii.ʧuru/ to speakers unfamiliar with the <ky> digraph of orthographies of Northwestern Ghana.

These ambiguities are due to mapping relations between phonemes and graphemes. Transcription systems and orthographies often have complex grapheme-to-phoneme relationships and they vary in levels of phonological abstraction. The transparency of the relation between spelling and phonology differ between languages like English and French, and say Serbo-Croatian. The former represent deep orthographic systems where the same grapheme can represent different phonemes in different contexts. The latter, a shallow orthography, is less polyvalent in its grapheme-to-phoneme relations. Challenges of ambiguity resolution are particularly apparent in data conversion.

4 Ontological Structure and Design

4.1 Technologies

In Philosophy, Ontology is the study of existence and the meaning of being. In the Computer and Information Sciences, *ontology* has been co-opted to represent a data model that represents concepts within a certain domain and the relationships between those concepts. At a low level an ontology is a taxonomy and a set of inference rules. At a higher-level, ontologies are collections of information that have formalized relationships that hold between entities in a given domain. This provides the basis for automated reasoning by computer software, where content is given meaning in the sense of interpreting data and disambiguating entities. This is the vision of the Semantic Web,[9] a common framework for integrating and correlating linked data from disparate resources for interoperability (Beckett, 2004). The General Ontology for Linguistic Description (GOLD) is grounded in the Semantic Web and provides a foundation for the interoperability of linguistic annotation to enable intelligent search across linguistic resources (Farrar and Langendoen, 2003). Several technologies are integral to the architecture of the Semantic Web, including Unicode, XML,[10] and the Resource Description Framework (RDF).[11] OATS has been developed with these technologies and uses SPARQL[12] to query the knowledge base of linked data.

The Unicode Standard is the standard text encoding for the Web, the recommended best-practice for encoding linguistic resources, and the underlying encoding for OATS. XML is a general purpose specification for markup languages and provides a structured language for data exchange (Yergeau, 2006). It is the most widely used implementation for descriptive markup, and is in fact so extensible that its structure does not provide functionality for encoding explicit relationships across documents. Therefore RDF is needed as the syntax for representing information about resources on the Web and it is itself written in XML and is serializable. RDF describes resources in the form *subject-predicate-object* (or *entity-relationship-entity*) and identifies unique resources through Uniform Resource Identifiers (URIs). In this manner, RDF encodes meaning in sets of triples that resemble subject-verb-object constructions. These triples form a graph data structure of nodes and arcs that are non-hierarchical and can be complexly connected. Numerous algorithms have been written to access and manipulate graph structures. Since all URIs are unique, each subject, object and predicate are uniquely defined resources that can be referred to and reused by anyone. URIs give users flexibility in giving concepts a semantic representation. However, if two individuals are using different URIs for the same concept, then a procedure is needed to know that these two objects are indeed equivalent. A common example in linguistic annotation is the synonymous use of genitive and possessive. By incorporating domain specific knowledge into an ontology in RDF, disambiguation and interoperation over data becomes possible. GOLD addresses the challenge of interoperability of disparate linguistic annotation and termsets in morphosyntax by functioning as an interlingua between them. In OATS, the interlingua

[9] http://www.w3.org/2001/sw/

[10] http://www.w3.org/XML/

[11] http://www.w3.org/RDF/

[12] http://www.w3.org/TR/rdf-sparql-query/

between systems of transcription is the IPA.

4.2 IPA as interlingua

OATS uses the IPA as an interlingua (or pivot) to which elements of systems of transcription are mapped. The IPA was chosen for its broad coverage of the sounds of the world's languages, its mainstream adoption as a system for transcription by linguists, and because it is encoded (at least mostly) in Unicode. The pivot component resides at the Character ID entity, which is in a one-to-one relationship with a Unicode Character. The Character ID entity is provided for mapping characters to multiple character encodings. This is useful for mapping IPA characters to legacy character encoding sets like IPA Kiel and SIL IPA93, allowing for data conversion between character encodings. The IPA also encodes phonetic segments as small feature bundles. Phonological theories extend the idea and interpretation of proposed feature sets, an area of debate within Linguistics. These issues should be taken into consideration when encoding interoperability via an interlingua, and should be leveraged to expand current theoretical questions that can be asked of the knowledge base. Character semantics also require consideration (Gibbon et al., 2005). Glyph semantics provide implicit information such as a resource's language, its language family assignment, its use by a specific social or scientific group, or corporate identity (Trippel et al., 2007). Documents with IPA characters or in legacy IPA character encodings provide semantic knowledge regarding the document's content, namely, that it contains transcribed linguistic data.

4.3 Ontological design

OATS consists of the following ontological classes: Character, Grapheme, Document, Mapping, MappingSystem, WritingSystem, and Scripteme. WritingSystem is further subdivided into OrthographicSystem and TranscriptionSystem. Each Document is associated with the OLAC Metadata Set,[13] an extension of the Dublin Core Type Vocabulary[14] for linguistic resources. This includes uniquely identifying the language represented in the document with its ISO 639-3 three letter language code. Each Document is also associated with an instance of WritingSystem. Each TranscriptionSystem is a set of instances of Scripteme. Every Scripteme instance is in a Mapping relation with its IPA counterpart. The MappingSystem contains a list of TranscriptionSystem instances that have Scripteme instances mapped to IPA. The Grapheme class provides the mapping between Scripteme and Character. The Character class is the set of Unicode characters and contains the Unicode version number, character name, HTML entity and code point.

5 Implementation

5.1 Data

The African language data used in OATS were mined from Systèmes alphabétiques des langues africanies,[15] an online database of *Alphabets des langues africaines* (Hartell, 1993). Additional languages were added by hand. Currently, OATS includes 203 languages from 23 language families. Each language contains its phonemic and orthographic inventories.

5.2 Query

Linguists gain unprecedented access to linguistic resources when they are able to query across disparate data in standardized notations regardless of how the data in those resources is encoded. Currently OATS contains two phonetic notations for querying: IPA and X-SAMPA. To illustrate the querying functionality currently in place, the IPA is used to query the knowledge base of African language data[16] for the occurrence of two segments. The first is the voiced palatal nasal /ɲ/. The results are captured in table 2.

Table 2: Occurrences of voiced palatal nasal /ɲ/

Grapheme	Languages	% of Data
<ny>	114	84%
<ñ>	11	8%
<ɲ>	8	6%
<ŋ>	2	1%
<ni>	1	.05%

The voiced palatal nasal /ɲ/ is accounted for in 136 languages, or roughly 67% of the 203 languages queried. Orthographically the voiced palatal nasal /ɲ/ is represented as <ny>, <ñ>,

[13] http://www.language-archives.org/OLAC/metadata.html
[14] http://dublincore.org/usage/terms/dcmitype/
[15] http://sumale.vjf.cnrs.fr/phono/
[16] For a list of these languages, see http://phoible.org

<ɲ>, <ni>, and interestingly as <ŋ>. The two languages containing <ŋ>, Koonzime [ozm] and Akoose [bss] of Cameroon, both lack a phonemic /ŋ/. In these languages' orthographies, both <ny> and <ŋ> are used to represent the phoneme /ɲ/. With further investigation, one can determine if they are contextually determined allographs like the <d> and <r> in Sisaala Pasaale.

The second simple query retrieves the occurrence of the voiced alveo-palatal affricate /ʤ/. Table 3 displays the results from the same sample of languages.

Table 3: Occurrences of voiced alveo-palatal affricate /ʤ/

Grapheme	Languages	% of Data
<j>	84	92%
<dz>	2	2%
<gy>	2	2%
<dj>	1	1%
<ʤ>	1	1%
<ǧ>	1	1%

The voiced alveo-palatal affricate /ʤ/ is accounted for in 92 languages, or 45%, of the 203 languages sampled. The majority, over 92%, use the same grapheme <j> to represent /ʤ/. Other graphemes found in the language sample include <dz>, <gy>, <dj>, <ʤ>, and <ǧ>. The <ǧ> stands out in this data sample. Interestingly, it comes from Sudanese Arabic, which uses Latin-based characters in its orthography. It contains the phonemes /g/, /ɣ/, and /ʤ/, which are gramphemically represented as <g>, <gh> and <ǧ>.

These are rather simplistic examples, but the graph data structure of RDF, and the power of SPARQL provides an increasingly complex system for querying any data stored in the knowledge base and relationships as encoded by its ontological structure. For example, by combining queries such as 'which languages have the phoneme /gb/' and 'of those languages which lack its voiceless counterpart /kp/', 11 results are found from this sample of African languages, as outlined in Table 4.

5.3 Querying for phonetic data via orthography

The ability to query the knowledge base via a language-specific orthography is ultimately the same task as querying the knowledge base via the pivot. In this case, however, a mapping relation from the language-specific grapheme to IPA is first established. Since all transcription systems' graphemes must have an IPA counterpart, this relationship is always available. A query is then made across all relevant mapping relations from IPA to languages within the knowledge base.

Table 4: Occurrence of /gb/ and lack of /kp/

Code	Language Name	Genetic Affiliation
emk	Maninkakan	Mande
kza	Karaboro	Gur
lia	Limba	Atlantic
mif	Mofu-Gudur	Chadic
sld	Sissala	Gur
ssl	Sisaala	Gur
sus	Susu	Mande
ted	Krumen	Kru
tem	Themne	Atlantic
tsp	Toussian	Gur

For example, a user familiar with the Sisaala Western orthography queries the knowledge base for languages with <ky>. Initially, the OATS system establishes the relationship between <ky> and its IPA counterpart. In this case, <ky> represents the voiceless alveo-palatal affricate /ʧ/. Having retrieved the IPA counterpart, the query next retrieves all languages that have /ʧ/ in their phonemic inventories. In the present data sample, this query retrieves 99 languages with the phonemic voiceless alveo-palatal affricate. If the user then wishes to compare the graphemic distributions of /ʧ/ and /ʤ/, which was predominately <j>, these results are easily provided. They are displayed in Table 5.

The 97 occurrences of /ʧ/ account for five more than the 92 languages sampled in section 5.2 that had its voiced alveo-palatal affricate counterpart. Such information provides statistics for phoneme distribution across languages in the knowledge base. OATS is a powerful tool for gathering such knowledge about the world's languages.

5.4 Code

There were two main steps in the implementation of OATS. The first was the design and creation of the OATS RDF model. This task was undertaken

Table 5: Occurrences of voiceless alveo-palatal affricate /ʧ/

Grapheme	Languages	% of Data
<c>	60	62%
<ch>	28	29%
<ts>	3	3%
<ky>	2	2%
<ʧ>	1	1%
<tʃ >	1	1%
<j>	1	1%
<č>	1	1%

using Protege,[17] an open source ontology editor developed by Stanford Center for Biomedical Informatics Research. The use of Protege was primarily to jump start the design and implementation of the ontology. The software provides a user interface for ontology modeling and development, and exports the results into RDF. After the architecture was in place, the second step was the development of a code base in Python[18] for gathering data and working with RDF. This code base includes two major pieces. The first was the development of a scraper, which was used to gather phonemic inventories off of the Web by downloading Web pages and scraping them for relevant contents. Each language was collected with its ISO 639-3 code, and its orthographic inventory and the mapping relation between these symbols and their IPA phonemic symbols. The second chunk of the code base provides functionality for working with the RDF graph and uses RDFLib,[19] an RDF Python module. The code includes scripts that add all relevant language data that was scraped from the Web to the OATS RDF graph, it fills the graph with the Unicode database character tables, and provides SPARQL queries for querying the graph as illustrated above. There is also Python code for using OATS to convert between two character sets, and for error checking of characters within a document that are not in the target set.

6 Conclusion and Future Work

OATS is a knowledge base that supports interoperation over disparate transcription systems. By leveraging technologies for ontology description, query, and multilingual character encoding, OATS is designed to facilitate resource discovery and intelligent search over linguistic data. The current knowledge base includes an ontological description of writing systems and specifies relations for mapping segments of transcription systems to their IPA equivalents. IPA is used as the interlingua pivot that provides the ability to query across all resources in the knowledge base. OATS' data source includes 203 African languages' orthographic and phonemic inventories.

The case studies proposed and implemented in this paper present functionality to use OATS to query all data in the knowledge base via standards like the IPA. OATS also supports query via any transcription system or practical orthography in the knowledge base. Another outcome of the OATS project is the ability to check for inconsistencies in digitized lexical data. The system could also test linguist-proposed phonotactic constraints and look for exceptions in data. Data from grapheme-to-phoneme mappings, phonotactics and character encodings can provide an orthographic profile/model of a transcription or writing system. This could help to bootstrap software and resource development for low-density languages. OATS also provides prospective uses for document conversion and development of probabilistic models of orthography-to-phoneme mappings.

[17] http://protege.stanford.edu/

[18] http://python.org

[19] http://rdflib.net/

Acknowledgements

This work was supported in part by the Max-Planck-Institut für evolutionäre Anthropologie and thanks go to Bernard Comrie, Jeff Good and Michael Cysouw. For useful comments and reviews, I thank Emily Bender, Scott Farrar, Sharon Hargus, Will Lewis, Richard Wright, and three anonymous reviewers.

References

Timothy Baldwin, Steven Bird, and Baden Hughes. 2006. Collecting Low-Density Language Materials on the Web. In *Proceedings of the 12th Australasian World Wide Web Conference (AusWeb06)*.

David Beckett. 2004. RDF/XML Syntax Specification (Revised). Technical report, W3C.

Steven Bird and Gary F. Simons. 2003. Seven Dimensions of Portability for Language Documentation and Description. *Language*, 79(3):557–582.

Regina Blass. 1975. *Sisaala-English, English-Sisaala Dictionary*. Institute of Linguistics, Tamale, Ghana.

Adams Bodomo. 1997. *The Structure of Dagaare*. Stanford Monographs in African Languages. CSLI Publications.

Florian Coulmas. 1999. *The Blackwell Encyclopedia of Writing Systems*. Blackwell Publishers.

Scott Farrar and Terry Langendoen. 2003. A Linguistic Ontology for the Semantic Web. *GLOT*, 7(3):97–100.

Scott Farrar and William D. Lewis. 2006. The GOLD Community of Practice: An Infrastructure for Linguistic Data on the Web. In *Language Resources and Evaluation*.

Dafydd Gibbon, Baden Hughes, and Thorsten Trippel. 2005. Semantic Decomposition of Character Encodings for Linguistic Knowledge Discovery. In *Proceedings of Jahrestagung der Gesellschaft für Klassifikation 2005*.

Rhonda L. Hartell. 1993. *Alphabets des langues africaines*. UNESCO and Société Internationale de Linguistique.

William D. Lewis. 2006. ODIN: A Model for Adapting and Enriching Legacy Infrastructure. In *Proceedings of the e-Humanities Workshop, held in cooperation with e-Science 2006: 2nd IEEE International Conference on e-Science and Grid Computing*.

Stuart Mcgill, Samuel Fembeti, and Mike Toupin. 1999. *A Grammar of Sisaala-Pasaale*, volume 4 of *Language Monographs*. Institute of African Studies, University of Ghana, Legon, Ghana.

Stuart McGill. 2004. Focus and Activation in Paasaal: the particle rɛ. Master's thesis, University of Reading.

Steven Moran. 2008. *A Grammatical Sketch of Isaalo (Western Sisaala)*. VDM.

The Unicode Consortium. 2007. *The Unicode Standard, Version 5.0*. Boston, MA, Addison-Wesley.

Mike Toupin. 1995. The Phonology of Sisaale Pasaale. *Collected Language Notes*, 22.

Thorsten Trippel, Dafydd Gibbon, and Baden Hughes. 2007. The Computational Semantics of Characters. In *Proceedings of the Seventh International Workshop on Computational Semantics (IWCS-7)*, pages 324–329.

Francois Yergeau. 2006. Extensible Markup Language (XML) 1.0 (Fourth Edition). W3C Recommendation 16 August 2006, edited in place 29 September 2006.

Kie Zuraw. 2006. Using the Web as a Phonological Corpus: a case study from Tagalog. In *Proceedings of the 2nd International Workshop on Web as Corpus*.

Author Index