

Local Language Speech Technology Initiative (LLSTI)

Ibibio Speech Synthesis Project Progress Report December 2003 - May 2004

Dafydd Gibbon, Eno-Abasi Urua, Moses Ekpenyong

1. Overview

The origins of the LLSTI Ibibio TTS project are fourfold:

1. Work on close copy synthesis of African French by Dafydd Gibbon and Sandrine Adouakou, using the MBROLA system, in the late 1990s.
2. Cooperation between Dafydd Gibbon and Eno-Abasi Urua during her tenure of a Humboldt Fellowship in Bielefeld in 2000-2001.
3. Discussions between Dafydd Gibbon and Roger Tucker at various points in the past few years about applying speech technologies to local languages, and subsequent facilitation work by Roger Tucker.
4. A DAAD funded project on the development of a joint curriculum for language documentation between the universities of Bielefeld (Germany), Uyo (Nigeria) and Cocody (Abidjan, Côte d'Ivoire).

The Ibibio speech synthesis project has the goal of producing a basic TTS system for Ibibio during the LLSTI pilot phase year. The Ibibio language community is the fourth largest in Nigeria, with several million speakers (the larger communities are Hausa, Yoruba and Igbo). Existing language resources

The following tasks were defined, based partly on previous work:

1. *Collation of a lexical database for Ibibio.* The lexical database will initially be based on an existing dictionary which was re-typed, partly prior to the project, and will be extended using corpus material. The lexical database will serve several purposes:
 1. TTS:
 1. additional corpus material for TTS unit selection,
 2. source of tonal information for lexical tone assignment,
 3. component of spell checker for TTS input texts.
 2. Community use:
 1. print medium dictionary,
 2. hypertext dictionary (hyperlexicon) for CD-ROM and web distribution.
2. *Construction of an Ibibio speech and text corpus.* The corpus will serve the following purposes:
 1. TTS:
 1. basis for TTS unit selection,
 2. extension of lexical database.
 2. Community use:
 1. Community heritage and information archive.
 2. Source of educational materials.
3. *Development of a model for Ibibio tone assignment.* Ibibio is a sub-type of *lexical tone language* which we will refer to as a *morphemic tone language*: like many other West African languages (and indeed languages in other parts of Africa, particularly Central Africa and South Africa), Ibibio has three kinds of tonal function in addition to some intonational functions. The three tonal functions are:
 1. Lexical tone: lexical morphemes are assigned tones which have phonemic function, distinguishing words with the same segmental structure:
 2. Morphosyntactic tone: tones represent grammatical morphemes, i.e. the tones themselves are

meaningful units (e.g. falling tone for *proximal future*, rising tone for *distal future*).

3. Syntactic tone: tones mark the internal structure and boundaries of syntactic domains such as Noun Phrases.

This report provides an overview of developments in each of these three areas during the start-up phase of the project.

2. Collation of a lexical database for Ibibio

A legacy print dictionary of Ibibio is available. Scanning and OCR treatment proved ineffective, so the dictionary was re-typed by the Uyo partners in Nigeria using the following procedure:

1. Definition of a lexical microstructure for the types of lexical information in the dictionary.
2. Re-typing as a table in MS-Word (the typist was familiar with MS-Word).
3. Export as CSV file.
4. Import to OpenOfficeCalc, the OpenOffice spreadsheet.
5. Post-editing using various sortings of the table.
6. Discussion of semantic information in the table, re-categorising according to definition, synonym, name categories, etc., and re-structuring the dictionary table.

The lexical database derived from the legacy dictionary has the structure shown in the table (before re-structuring and post-editing) for the first few entries. Note that the letter A alone has 3650 entries; the enormous task of re-modelling the dictionary manually was performed in Uyo under the supervision of Moses Ekpenyong and Eno-Abasi Urua.

<i>Beginning of Ibibio lexical database (based on the Kaufman dictionary)</i>					
Orthography	Tones	Pl	Pl_Tones	POS	Definition
aa	HH			interj	indicates surprise
aa	HF			interj	indicates surprise
aaba	HLF			num	forty
aafo	LHL			n	[Eng 'Half'] half penny
aafu	LHL			n	[Eng 'Half'] half penny
aak	HF			interj	please (emphatic, persuasive, very polite)
aakp̩N̩	HLHH			n	named child (someone named after another, each is 'akp̩'N̩ to the other)
aba edem	HF LL			n	the back (anat) ('ed'em back, side)
aba(1)	LL			n	cavity, hollow, hole
aba(2)	LL			num	forty

The next step is to design a formatting filter to convert the lexicon to SIL *Shoebox* format with the goals of

1. using the corpus facility for integrating further words into the lexicon,
2. using the formatting facility for producing the Ibibio lexicon.

3. Construction of an Ibibio speech and text corpus

On the basis of discussions with KS and PP, the following procedure was followed in order to create a machine readable corpus. For the corpus processing, ASCII conversions and scripting techniques were used (partly Perl, partly standard UNIX tool scripts) in order to take advantage of the string and regular expression processing facilities in rapid prototyping.

1. Corpus collation. In selecting the corpus, the flow of the sentences and ease of reading was taken into consideration and verified, and all rules regarding the formation of phrases and sentences observed in order to ensure well-formed input in the pilot phase. A corpus of minimally 3000 phrases was agreed (Ksenia Shalnova, Dafydd Gibbon); just over 3000 phrases were collected. The sources of the corpus are:
 1. The 1995 Beijing Platform for Action translated by the Women Information Centre, Uyo.
 2. News translations from NTA Channel 12, Uyo.
 3. Texts from published materials on Ibibio (Elerius Edet John, 2003; Effiong T. Inyang, n.d.).
 4. Texts written by project members.
2. Orthographic ASCII normalisation. In order to adapt to the requirements of scripting languages a 7-bit ASCII encoding of standard orthography was introduced. Conversion was effected by a small number of manual "replace all" operations in a word processor.
3. SAMPA adaptation. For ergonomic reasons, slight modifications
4. Grapheme-to-phoneme conversion. A grapheme-to-phoneme table and a phoneme-to-grapheme table were defined. In each case, both an ASCII and a Unicode version of the tables was produced; in all cases the phonemic representation is in SAMPA (i.e. 7-bit ASCII). Four scripts were developed (Thorsten Trippel), one for each of these four cases. The remaining processing was performed on the SAMPA representation.
5. Wordlist extraction. For verification purposes (and for extending the dictionary) a wordlist was automatically extracted from the SAMPA text with standard UNIX tool scripting (Dafydd Gibbon)
6. Concordance creation. For later use in morphology and grammar development and testing a concordance was produced from the SAMPA text with standard UNIX tool scripting (Dafydd Gibbon).
7. Phoneme list extraction. The phonemes which occurred in the SAMPA text were listed and validated for membership of the Ibibio SAMPA conventions.
8. Unit list extraction. Initially only diphone units were selected and formatted according to the Hindi sample in the OTS tool distribution (Partha Pratim Talukdar). A total of 445 units were detected. This count is typical of languages with shorter syllable structures (V, CV, CVC etc.) than the Indo-European languages, which have notoriously long syllables.
9. Unit selection. After detailed advice from Partha Pratim Talukdar, an algorithm for extracting units from the SAMPA text was developed with the following properties (see Appendix):
 1. Selection units are diphones with no further weighting (unit weight).
 2. Phrases were extracted based on punctuation criteria.
 3. The format was normalised to the Hindi sample format.
 4. A *longest sentence first* heuristic was used to find a smallest sentence set. This will need to be modified at a future stage in order to use the unique unit count in a sentence rather than the sentence length.
 5. Outputs were produced in the Hindi sample format and in a readable format with normal spacing.
10. Post-editing. The output of the unit selection tool (53 phrases from a total of over 3000) was checked for correctness and readability (Eno-Abasi Urua, Moses Ekpenyong). Successive versions were preserved. Several problem features were identified:
 1. Spelling mistakes.
 2. Some phrases were incorrectly chunked by the punctuation criterion, and were adapted for reading by adding or deleting initial or final elements (Eno-Abasi Urua, Moses Ekpenyong). Checks were made

to ensure that all diphone units were still covered.

3. Phrasing in oral rendering yields shorter phrases than the punctuation provides, so how is oral phrasing to be handled? In this pilot phase, phrases were marked for reading by hand.
4. Loan words from English (or French, etc.) are very typical of post-colonial regional languages and need a principled method for handling them, both in terms of pronunciation and in terms of translation:
 1. Ibibio nouns begin with a prefix (vowel or syllabic nasal), so how does one borrow 'police', 'stephen', 'Jesus Christ', etc?
 2. years (1994, 1671, 2015)
 3. abbreviations (UN, NYSC, NNPC, NAFDAC)
 4. foreign names (Grace, Moses)
 5. titles (Reverend Father, Bishop)
11. Recording. The selected and post-edited phrases were recorded on DAT tape at 48 kHz with no compression on a Sony portable DAT recorder using the following procedure:
 1. Reader: Eno-Abasi Urua.
 2. Operator: Dafydd Gibbon.
 3. Director: Moses Ekpenyong.
 4. Equipment: DAT tape, 48 kHz, no compression, stereo with two Sennheiser condenser studio microphones.
 5. Locale: Office with close-placed microphones.
 6. Length: 27 minutes.
 7. Comments:
 1. After approximately 10 phrases a short reading pause was made and the reader took a sip of water to avoid drying out the vocal folds.
 2. Some phrases were repeated due to misperformance or extraneous noise.
12. Archiving. The DAT recording was transferred to WAV format with CoolEdit2000 and stored on CD-ROM:
 1. A single file with the complete recording.
 2. 59 files with the separate phrases.
13. Next steps:
 1. Further development of the unit selection process:
 1. Application of the OTS tool.
 2. Validation of the correctness of the UNIX script by unit analysis of the output.
 3. Validation of the correctness of the corpus-based diphone unit extraction using a diphone table based on Ibibio speech sounds and their phonotactic restrictions (with the assistance of Udeme Edet and Mfon Udoinyang, Graduate Assistants in the Department of Linguistics and Nigerian languages, University of Uyo).
 4. Extension of the UNIX script to include other sentence and unit weighting criteria.
 5. Porting of unit selection to a common interoperative multi-platform script language like Perl.
 2. The files have not yet been annotated. The annotation will be done either with Praat or with Transcriber. Perl scripts are available (Thorsten Trippel) for interconversion between Praat, Transcriber and esps/waves+ formats.
14. Problem areas for TTS:
 1. The Ibibio orthography does not contain tone marks since these are not needed by the native reader,

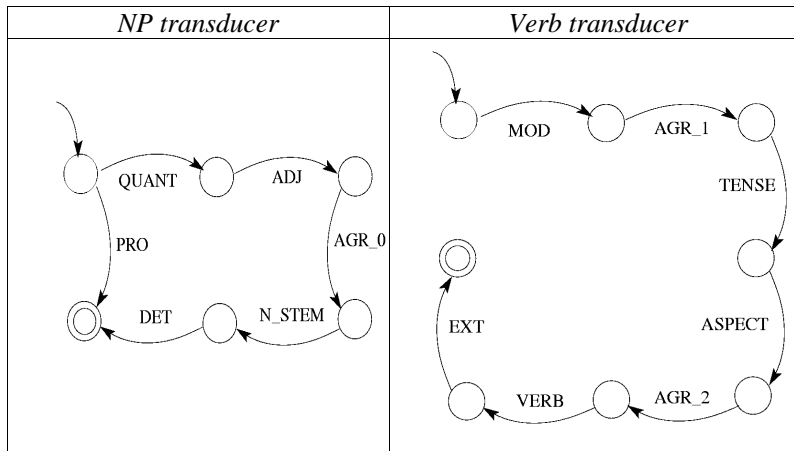
who can disambiguate words on the basis of the grammatical and co-textual context (as with any kind of lexical ambiguity). Consequently, the tone marks will need to be assigned according to the following three kinds of linguistic criterion:

1. Lexicon lookup for lexical tone.
 2. Morphosyntactic parsing for morphosyntactic tone (with finite state transducer).
 3. Syntactic criteria for intonational tone effects (not well identified at present).
2. Out of vocabulary words: A procedure needs to be defined for dealing with the unknown tones of out of vocabulary words in input texts. There are native reader intuitions on this, but there is also a strongly arbitrary element. One solution would be to use a default tone for a given word class, defined in terms of the frequency of tone patterns in the corpus or in the lexicon for this word class.

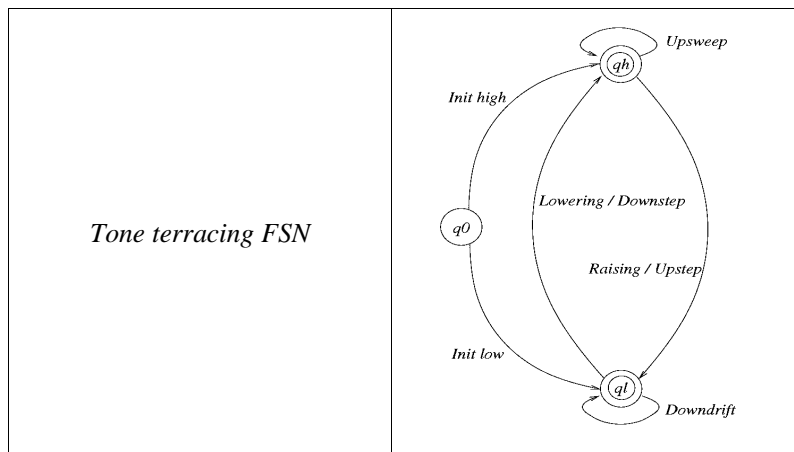
4. Morphology

Based on previous work on finite state tonology and finite state tonology, a multi-tape finite state transducer for Ibibio morphosyntax was designed (see Appendix) and implemented with standard UNIX tool scripting techniques. In detail the techniques were not standard UNIX, because the non-deterministic FST was implemented with backtracking using recursive programming in *awk*, an unusual technique for this environment.

The basis for the morphosyntax is provided by the two F'SNs (Finite State Networks) shown in the table.



The terraced tone modelling is based on the FSN shown in the table, which describes conditions which are characteristic of many two-tone West African languages.



The FSN designed for tone association concatenates the NP and V FSNs and composes the result with the tone terracing FSN. The detailed specifications will be provided in a later report. A typical output provides parallel strings of

1. lexically and morphosyntactically determined tones,
2. morphemes,

in the style of interlinear glossing, as shown in the table. The alignment is implicit: one tone per vowel.

```

TONE:  L L + H + H + H L
SEGM:  ami + n + bed + miin

TONE:  L L + H + L H + H + H L
SEGM:  ami + n + yaa + bed + miin

TONE:  L L + H + H L + H + H L
SEGM:  ami + n + yaa + bed + miin

```

TONE: L L + H + L H + H + H L
SEGM: ami + n + maa + bed + miin

TONE: L L + H + H L + H + H L
SEGM: ami + n + maa + bed + miin

TONE: L L + H + H + H L
SEGM: ami + n + bed + nnjIn

TONE: L L + H + L H + H + H L
SEGM: ami + n + yaa + bed + nnjIn

TONE: L L + H + H L + H + H L
SEGM: ami + n + yaa + bed + nnjIn

Other formats are possible; currently the implementation is not at all generic; the Ibibio facts are hard-wired into the code.

It is planned to change to a generic style of implementation. It should be possible to use the Generic Morphology Tool (no author, no date) used in the LLSTI consortium.

5. Appendix 1: Ibibio SAMPA

In the Ibibio TTS project, we have had to modify the original SAMPA to suit the ergonomic needs of readers and writers of Ibibio transcriptions, which the members of the project have christened 'Ibibio SAMPA'. The selected texts for the TTS project have to be re-wordprocessed using Ibibio SAMPA. This was initially done manually; at a later stage the Ibibio grapheme-to-phoneme tool will be used.

<i>Ibibio SAMPA table: X-SAMPA modified for ergonomic reasons (ease of reading and writing)</i>		
Vowels:	a	a
	e	e
	i	i
	o	o
	O	open o
	V	inverted v
	@	schwa
	I	barred i
	u	u
	}	barred u
Consonants:	b	b
	p	p
	t	t
	d	d
	k	k
	m	m
	n	n
	N	velar nasal
	J	palatal nasal
	h	h
	g	h
	r	r
	s	s
	j	y
	kp	kp

Thus this Ibibio sentence “atim ake bok afia efere ndien enye adika akeni ajin eba” would read thus in SAMPA: atIm akebOk afia efere ndien eJe adika akenO ajIn eba.

6.Ibibio grapheme-to-phoneme script

```
#!/usr/local/bin/perl
#-w
# praat to sampa ipa font converter
# By Thorsten Trippel
# University of Bielefeld
# ttrippel@spectrum.uni-bielefeld.de
# November 2003
#
# This program requires perl 5.8 or higher,
use warnings;
#use Data::Dumper;
$infile= $ARGV[0];
$outfile= $ARGV[1];
open IN,$infile || die("Konnte File $infile nicht oeffnen! $!");
    binmode(IN, ":utf8");
open OUT,d ">$outfile" || die("Konnte Datei $outfile nicht schreiben: $!");
    binmode(OUT, ":utf8");
open G2P, "<g_ascii2ptable.csv" ||die("No G2P table found, aborted");
    binmode(G2P, ":utf8");
#my @ibiblioG2P2 = <G2P>;
my %ibiblioG2P2;
my $grapheme;
my $phoneme;
while (<G2P>){
    chomp;
    s/\"//g;
    ($grapheme,$phoneme) = split /\t/,$_;

    $ibiblioG2P2{$grapheme} = $phoneme;
    #print STDERR %ibiblioG2P2;
}
while (<IN>){
    chomp;
    tr/[A-Z]/[a-z]/;
    s/\"//g;
    s/\'//g;
    s/\'//g;
    $var=$_;
while (($grapheme,$phoneme)= each(%ibiblioG2P2))
{
    $var=~s#s#$grapheme#$phoneme#g;
};
$var =~ s/\.\.\.\\n/g;
$var =~ s/,/,\\n/g;
print OUT "$var";
};
close G2P;
close IN;
close
```

OUT;

7. Grapheme-to-phoneme table

Note that the rendering in the table below is incorrect because the proprietary font is currently not available.

<i>Grapheme (proprietary font)</i>	<i>Phoneme (Ibibio SAMPA)</i>
Ƶ	i
Ƶ⊗	I
ƶ	e
ƶ	a
⊗	O
⊗	o
Ƶ⊗	V
ƶ⊗	@
⊗	ʃ
⊗	u
⊗	p
⊗	b
⊗	t
⊗	d
⊗	k
⊗	f
⊗	s
⊗	j
⊗	w
⊗	m
⊗	n
⊗④	N
⊗	R
⊗	R\
⊗	r
⊗	J

8. Appendix: Unit selection script

```
#!/bin/sh
# selectphrases02.sh
# D. Gibbon
# Extract phoneme set and diphone set from text

#=====

# Phonetised text format:
# >-U-n-h-O-N-n-E-#-b-a-t-A-y-A-#-k-I-#-p-U-l-I-s-#-n-E-#-w-a-j-i-r-#-b-A-g-#-s-th-I-t-#-U-n-
k-E-#-A-w-A-s-#-k-i-#-s-U-b-a-h-#-gh-E-r-A-b-a-n-d-i-k-a-r-#-l-i-#-o-r-#-U-n-h-En-#-n-a-m-A-j-#-a-t-
A-#-k-a-r-n-E-#-k-E-#-l-I-E-#-b-A-h-a-r-#-n-a-h-i-n-#-j-A-n-E-#-d-I-y-A-#-g-a-y-A-<
# >-U-n-h-O-N-n-E-#-k-a-h-A-#-E-p-i-E-c-s-i-#-n-E-#-k-I-s-i-#-h-a-R-t-A-l-#-k-A-#-A-h-w-A-n-#-n-a-h-
in-#-k-I-y-A-#-h-e-#-o-r-#-m-U-jh-E-#-s-a-m-a-jh-#-m-En-#-n-a-h-i-#-A-t-A-#-y-a-h-#-k-a-d-a-m-#-k-y-
O-En-#-U-Th-A-y-A-#-g-a-y-A-#-h-e-<

# Unit format
# #-w 758
# #-y 252
# >-A 32
# >-D 2
# ...
# >-w 86
# >-y 30
# A-# 2396
# A-< 191
# A-A 2

#=====

STATS=stats.txt

FILE=`echo $1 | sed "s/.txt$/g" | sed "s/.data$/g"`
RAWFILE=$FILE.data
INFILE=$FILE-in.txt

PHONETIZED=$FILE-LLSTI-phonetised.txt
PHONEMES=$FILE-LLSTI-phonemes.txt
UNITSWEIGHTED=$FILE-LLSTI-unitsweighted.txt
UNITS=$FILE-LLSTI-units.txt
SELECTED=$FILE-LLSTI-selected.txt
READABLE=$FILE-LLSTI-readable.txt

if [ ! -f $RAWFILE ]
then
    echo No input file.
    exit
fi

#=====

echo Creating input file with separate phrases and normalising ...

cat $RAWFILE |
tr "\015" "\012" |
tr "[-]" "\012" |
tr "[,]" "\012" |
tr "[.]" "\012" |
tr "[;]" "\012" |
tr "[:]" "\012" |
tr "[?]" "\012" |
tr "[!]" "\012" |
grep . |
sed "s/ * /g" |
sed "s/[-,.;:<>](/\"'\`'//g" |
tr "[" " " |
tr "]" " " |
sed "s/^ */g" |
sed "s/ */g" |
sed "s/ */g" |
tr " " "#" |
# sed "s/#>/>/g" |
# sed "s/<#/</g" |
grep . > $INFILE

#=====
```

```

echo Creating Parthas phonetized text format ...

cat $INFILE |
sed "s/^/>/g" |
sed "s/$/</g" |
sed "s/./&-/g" |
sed "s/^-*$///g" |
sed "s/-*$///g" > $PHONETIZED

=====

# Create list of phonemes in text for checking purposes

echo Creating list of phonemes for checking ...

cat $PHONETIZED|
tr "-" "\012" |
tee tmp.phon |
sort -u |
grep . > $PHONEMES

=====

# Create list of diphone units

# tmp.phon contains text with phoneme occurrences listed one per line

echo Creating list of diphone units...

cat tmp.phon |
tail +2 > tmp.second
paste tmp.phon tmp.second |
tr "\011" "-" |
grep -v "<->" |
grep -v "\-$" |
sort |
uniq -c |
tr "\011" " " |
sed "s/^ *///g" |
sed "s/ */ /g" |
grep . |
sort -rn |
gawk '
    {print $2, $1}
' |
sort |
tee $UNITSWEIGHTED |
gawk '
    {print $1}
' > $UNITS

=====

# Unit selection algorithm

# The idea is to select the minimum number of occurrences of units
# in the minimum number of phrases.

# The heuristic for finding the minimum number of phrases is to sort
# the corpus by sentence length on the assumption that the longer the
# phrase, the more units occur in the same phrase.

# The procedure for finding the minimum number of occurrences of units
# (i.e. as phonetically balanced as possible, assuming equal weighting) is
# repeat for each unit:
#   - find the phrase with the first occurrence of a unit,
#   - add this phrase in an initially empty set of selected phrases
#     if it is not there already,
#   - add the set of selected phrases to the head of the corpus
# emit the final set of selected phrases

# sort the corpus according to sentence length
# create a selected phrases file
# for each unit
# do
#   select the phrase in the corpus with the first occurrence
#   append this phrase to the selected phrases file
#   prepend the selected phrases file to the corpus
# done

```

```

# emit selected phrases file

#=====

cat $PHONETIZED |
gawk -F"- " '{print NF,$0}' |
sort -nru |
grep . > tmp.oldinput

#=====

rm -f $SELECTED
touch $SELECTED

for UNIT in `cat $UNITS | gawk '{print $1}'`
do
    echo So far `wc -l $SELECTED | gawk '{print $1}'` phrases selected at unit $UNIT
    grep $UNIT tmp.oldinput |
    head -1 |
    grep . >> $SELECTED
# Use numeric sort:
    sort -nru $SELECTED > tmp.selected
    mv -f tmp.selected $SELECTED
# Use this order to ensure prepending:
    cat $SELECTED tmp.oldinput > tmp.newinput
    mv -f tmp.newinput tmp.oldinput
done

#=====

echo Emit selected phrases in Parthas format and in readable format ...

rm -f $READABLE
touch $READABLE

echo Selected Ibibio sentences \(`date`\) >> $READABLE
echo `whoami`, `date`, $0 >> $READABLE
echo "" >> $READABLE

cat $SELECTED | gawk '{print $2}' |
tee tmp.selected |
sed "s/[-><]//g" |
sed "s/#/ /g" |
gawk '
{
    print NR ": " $0
    print ""
}
' >> $READABLE

mv -f tmp.selected $SELECTED

rm -f tmp.*

echo Done.

#=====

```

9. Appendix: Ibibio minimal phrase set selected for diphones

The following phrase set was generated automatically from the Ibibio corpus. This set is not post-edited.

Selected Ibibio sentences (Mon Mai 17 18:09:25 CEST 2004)
gibbon, Mon Mai 17 18:09:25 CEST 2004, ./selectphrases02.sh

1: odoho keed ese NkpO abaNu uforo ibaan je eJe eseehe NkpO abaNu usuan etop nOO afId unadod emaanam ekikere edikappa mme NkOOKiwood emi nsIn ke mme usem ukaN nJIn nO odo se afId ibaan mme ideen ke idId nJIn esIme ekood eJVN enie ifiOk ebaNa se ikikaiso ke mbono emi afId unadod ekedVkkO je mme ekikere ekereke ebaNa ibaan

s2: NJIn ibioro ibo jak afId mme Nka emi edian ubOk edukoppo daNa uforo OdVk OnO mme nsionsio idid OmmO ke se NJIn inam ke mbono beiJIn emi ediwuO usVN edieke mme ukara keedekeed ke mme idId mme idId emaedian ubOk mme NJIn eJVN eJImme edibere mme NJIn NkId mbo ke mme ediomO NJIn emi edVk ke edinam

3: Oku abasi ekoodo bisioP udeeme saimOn abaaha ke uFOk abasi ekamba mbuOtidem amaanam nO emi anwaNa ke ini eJe ekenamma edere ekOOm ekewOttO utId usOrO mme edinam usen emana OjOhO itiaba ake uFOk abasi odo mme OjOhO isua OmO keed tOOONO nte ekedod eJe ke itie ukpono nte bisioP uFOk abasi odo

4: ete udOkaN amaataN obo ke OwuO etop NkpO ake mmimO ikinamma mi akeene daNa mmimO ikisIn ifIk iben NkpOutom ake ukara sted ekenOOhO ke isua ifaN oboijoke ko inam nO OwuONO usVN ndion ke uFOk utom odo OsVk odomo ukeme OmO edikId mbo ke mme idVN mfen ake mmOON esaN ukwak mmiisImme OmmO kaNa

5: usVk ke utO mme mbono emi edVNNO mbono unen afId Owu ekeniehe ke idVN Vienna ke obio austria ke OjOhO isua tOsIn keed je ikie usVkkeed je anaaN je duop eta je ekamba mbono eke Ibat Owu je Nkod ke idVN kairo ke ijIb ke OjOhO isua tOsIn keed je ikie usVkkeed je anaaN je duop enaaN

6: mma ekpeJON amaabo ke et@k utom imO ikamake ke ubOk ekoodo Nka unwam nt@kejIn ejaediana mme Nka Nkaan nditO ufOkNwed NkOrO mme mboho mfen ke abaNu edikId mbo ke ukpeebNkpO asaNu mfOn mfOn ke ufOkNwed mbaak oto ke ndo edIppe mme idaha ufOkNwed

7: ufOkutom emINNe OkVk ke naijiria emaejaara ami ke Nwed kOmiti NkOrO afId ufOkutom odo ekenOOhO esOk adaaji umar khadi odooho ekamba ataNiko ke ekamba uFOk mbed ukara ake abuja ke abaNu uduak edijak utom umINNkpO ake ukara nnOO mbonowo

8: OkOmbVk batrik ssien ataN obo ke ete joo akenofa odooho andise NkpO mbaNa ufOkutom mObiid dian mme akama utom mfen emaebo ke edinam ami akewOd obo ke ufOkutom ami esIn ifIk adikId mbo ke Nkod mme uforo adad itie ke mme mbOhOidVN OmO

9: ubOON sONde udo odooho keed ke otu mbon ebOkkOke Nkpa ke ikaN odo ataN OnO OkOmbVk uFOk utom usuan mbVk ake ekebe ndise ke uFOk IbOk eJe anaaha ObO usObO ke idaha imO ikiwuOhO ke ubeed imO ikaa ediJaaNa mbon uFOk odo nsio ke ikaN

10: nsinia basi amaaneke ataN abaNu edinam nO mme itie ke uFOk ibOk odo amaana atara akan daNa OmmO edONNO ke idaha esVkkO enam mbeeNeidem ebaNa edikoono akpa iduOk nditO ufOkNwed ukpeebNkpO mbaNa ibOk mbakara mme nsONidem ndON

11: enam ediOONO ke ekamba unOmO NkpO isaN aketippeke ke usen OfiON edIb mme inaaN ke OfiON oko ke ikOt akpan udo ke idak ifIm ukara ake ikOnO amaawod owu inaaN ekeekoonoke ekedON ke itie ubon okpo ke uFOk ibOk ake ikOt ekpene

12: brins enaN amaataN obo ke ekemek njobio owu ete nte andibo ukpono emi ke ini owu itiaita etooho Nka use NkankVk ekondo ke dOndOn ekedieehe edise se atie iwuod okpokoro oro akenam ke abaNu edidIppe idaha ikpehe ukara odo

13: Nwed etop odo amaabo nO mme akama utom ke uFOk utom usuan mbVk ake ekebe ndise ujo ekpe mme anam utom OmmO et@k OkVk ekpad akpenaaha nte ekpe daNa ekpe ke itie usuan mbVk ujo naijiria ke NkpO nte isua keed idahami

14: dOkta Okadibo amaadOkkO adaiwuod ukara sted nJIn obo ke Nka isaN kOmiti odo ekId ebo ke mme mfIna ke NkaN usop utIn edubiotto mbeed ejaekop inemesId ebaNa uforo mme utom emem ke ikpehe ediOONOke nte naija deedta

15: ikO odo amaana asIn ebeeNe obo nO ukeed nditO anaaN se ibaNu mi ke naijiria mme ke esen idId enONNO mme NkpO unwam nO esOk nditOeka OmmO ke ika ake nnanenJIn OwOONOke OmmO mbaak oto ke ndo mbiomO OmmO afeere

16: mma ekaete OrOk amaataN obo ke baNk odo akemana ke OjOhO isua tOsIn keed je ikie usVkkeed ye ata ye efiTenaan ndion ke aketOONO ndubehe ndutVm OkVk ke OjOhO isua tOsIn keed je ikie usVkkeed je anaaN je inaaN

17: ekamba anam nduuNO mbaNa ndutVm OkVk ke mme ikpehe ukara ke akwa ibom sted odooho ete sONde ibaNu OnwONO ke ijainO mbon ufOkutom imO eesInne idem enam utom NkpO ajIn utom mbaak mbon sisVhO ekeene OmmO edem

18: anamutom odo keed akebaaha idaha ikaN odo aketippeke amaataN obo ke ikaN odo aketippe ke idaha keed ke otu mbon esidomoke enJin ikaN akejemme adisio ikaN isektrik aketooho enjin ikaN odo nsIn ke ake nepa

19: ke OjOhO OfiON usVkkeed ke OjOhO isua tOsIn keed je ikie usVkkeed je anaaN je efId tOONO eketOONO utO mbono emi ke NkpO nte isua aba mme duop oboijoke do ke idak NkOk eseehe NkpO abaNafId unadod

20: keed ke otu ami odo ete mme mma udoudom odooho ekamba akama ndutVm ke ufoK utom usuan mbVk ake ekebe ndise ujo akedaaha ke iwuod ekamba ase NkpO mbaNa ufOkutom usuan mbVk ake ekebe ndise ke ujo

21: eJe amaakOom abasi akekamake OmmO ke isua itiaba akeboijoke OJVN OnO mbon ufOk abasi OmO item obo nO OmmO esOONO eda ke mbuOtidem ake OmmO eniehe ke abasi mbaak utIp OmmO okpon ke heaven

22: aNwaan OmO mma mmedi edaija amadOkko mbon borisi mme mbon ufOk utom usuan mbVk ake ekebe ndise obo ke owo odo ekebine ebe imO abakusen odo nte eJe ekeben deta oto ufOk ibOk OsOk ebe imO

23: Nka ndadndad krOs ake naijiria enONNO mbon ikOt udo uruan mme enO edinwam idem OmmO ke ntak esId mbom OmmO eniehe ebaNa mbon ekenieehe unOmO ke anwaN idvN odo ekenieehe mme Nkanna OmmO

24: daNa ekspesIn uNwam nO uwem amaana enemme owo edidu ke ededimbod emi odo akpan nneme akedatta itie ke OjOhO isua tOsIn keed je ikie usVkkeed je anaaN je efId ke obio kopenhaken

25: eJe amaafaaNa obo ke OjOhO ifiOk kabaNa mme ikpehe mfIna ke idId nJIn naijiria NkOrO mme nti NkpOanJIn inieehe nte inie ajaanam nO ekON ekan enam utom OmmO OjOhO OjOhO

26: mbVk emi ekedikoppo edemusen udian Nwed ufa eJIN owo emi akewOd nte ke bodisi amaamVm nsinia iwuod ikId ye dOkta etIm ibOk ekpo ke ntak emi OmO ekemiaha ekepkp akai

27: etaN ebo ke eja enONNO mme isvN utom nO esaNa eseed mme nsionsio ufOkutom tOONO ke Nkanika itiaita ubahusen mbaak edioONO mme mme anamutom kOfmen enam ujo eekenOOho

28: ke edisaNa nseed NkpO ke sted ataN obo ke isaN mbon Nka usuanetop ake ataifiok jerri kana akedatta usvN emaanam ndudue ke mme ubieere OmmO ekenamma ebaNa akwa ibom

29: anamutom napep mfen aketaNna ikO OnO ufOk utom usuan mbVk ake ekebe ndise amaakwo ikwo itooro OnO ukara fedraad ke adikibeere usvN OkVv nO eJe adad aNwam idem OmO

30: mma kift sitifIns amaataN ke Nka odo ekedi ufOkutom odo edinam nO edioONO inemesId OmmO ekoppo ebaNa ufOkutom odo ke edibiommo OmmO mbiomo unana eti mmOON OmmO

31: Nka odo amaakop mbOm abaNafId daNa ekenONNO mme Nkpoon owo ke ufOk utom ufOk utom usuan mbVk ake ekebe ndise OkVv udeeme Nkpod@ho ufOk odo owo inONNOke mbon Nko

32: eJe amaawOd mme isvN utom odo mme itie ekekereke ebo ke mbon afai etooho krOs rifa sted ekebiaad ke ini mfIna akebaaha ke oku Iboku ke idak ikpehe ukara itu

33: atie iwuod okpokoro ufOkutom edioONOke nte bafiid kOnsOtium odooho njobio owo basi etIm akenOO item ami ke idaha Nka Nkparawa akwa ibom ekedieehe edise eJe

34: mma kres ekpeJON aketaN ami ke ujo ke mboho ufa isua eJe akenamma OnO Nkaan nditO ufOkNwed ibaan ake doreto ke eriam afaha OboN ke idak ikpehe ukara abak

35: ete ibaaNa amaabo nO mme owo esIn ifIk enam utom enO ukara sted sia odooho inua ikpikanna ibad ufOn ake mme anamutom ediaaha eto ukara ObOON fikto attah

36: amaana OkOom bastO maksweed OsamO odooho adausvN ke ufOkutom kOppa ke ikpehe ukara ikOt ekspene NkOrO mbon mfen ke item mme uNwam ake OmmO ekenOOho eJe

37: ukpeeb NkpO odo amaanam aNwaNa ke mme mfIna ake mme ibaan ufOkndO ajaamaana afeere ke OmmO ekpeben nditO ake miniehe ete ye eka ekamma nte nditO OmmO

38: dOkta ini udoka amaataN obo ke mbono emi odo eti mbono adinam nO mme owo edioONO ke usem aNwam eti eti ke mme ikpehe mme ikpehe eduuwem ke idvD nJIn

39: ataifiOk edwOd attah akenam edioONO ami ke ukpeeb NkpO usen ita ke abaNake udONO itiaita ekenImme enO mme Nkparawa ibaan mme ideen

40: esIn ujo ekood mme ikpO ufOk esop mi ke idId nJIn ebo nO emaana ekeduuNO mme ikpe eketaNna enIm ebo ke owo ikipheeh ke ataa unen

41: ufOkutom mObiid je enenbiisi enONNO mme nsionsio mboho mme Nka mi ke akwa ibom sted ibad OkVv akanna midiOn nnaida aba mfIn ami

42: odooho njobio owo udo sam umoataN amaawvD inemesId abaNafId adido keed ke otu mbon ekoodo ebo edisIn uNwam enO ufOkNwed odo

43: ufOk unIm OkVv ekoodo kooberatif defedopmen baNk akedvK idvN ke isua tOsIn keed je ikie usVkkeed je anaaN je duopeta

44: oJVN obo nO mbon ufOk abasi kwa ibo enekke esIn idem OmmO ke OkpOsON akam mbaak oto ke ndo abasi OjO ukuud OmmO

45: ndion IkO odo ataN obo ke owo ake ododo ediimVm daNa amaana OtOONO ntIme ke ika mme itie akeododo ke sted

46: senatO adOf Nwabara akenam ami aNwaNa ke ini eJe akekesehe mme aJam urua ke akamba urua ariaria ke aba

47: ataifiOk niji oshundare amaataN ikO abaNa edidemme ake afrika NkOrO mme ukuud NkpO mbre ukara OwOONOke

48: ataifiok akpan ekpo amaataN abaNa mme ufOn ufOk Nwed ntaifiOk ke edidIppe NkOd mme uforo ke idVn

49: enam nO ediOONO ke mma susana basiid essien otooho ata mmiJa ke idak ukara Nkpat enin osop

50: amaataN obo ke udON adikpVhO aJIN ufOk abasi odo ikitooho ndausuN ake sibiId abasi

51: jeneraad edet akpan amaadian obo ke OfOn owo adino

52: eka kristi obot

53: deKO

10. Appendix: FST all-tape generator script

```
#!/bin/sh
# ibiblio_fst_08.sh
# D. Gibbon
#=====
# History:

# ibiblio_fst_01.sh
# 2004-02-03
# 2004-02-06
# Plain FST with no cooccurrence restrictions.

# ibiblio_fst_02.sh
# 2004-02-05
# 2004-02-09
# Enhanced FST with registers for cooccurrence restrictions.

# ibiblio_fst_03.sh
# 2004-02-09
# Substitution of Tense yaa for maa corrected (Bruce Connell).
# History and description added.
# Registers given functional names: subject, object
# Proper names added at subject and object positions

# ibiblio_fst_04.sh
# 2004-02-09
# Simplified generator code by removal of general conditions

# ibiblio_fst_05.sh
# 2004-02-10
# Separate tonal tier

# ibiblio_fst_06.sh
# 2004-02-16
# Further corrections from Bruce Connell

# ibiblio_fst_07.sh
# 2004-02-19
# Further corrections from Eno Urua
# Added object agreement prefix
# NPs except pronouns commented out

# ibiblio_fst_08.sh
# 2004-02-22
# Added grammatical tone tape
# NPs except pronouns commented out
# Corrections: AGR2 2nd person object verbal prefix

# ibiblio_fst_09.sh
# 2004-02-22
# Change: distinguish between
# - AGR_1 as auxiliary prefix, AGR_2 as main prefix
# - AGR_1 only has subj pref, and only occurs if there is an aux,
# - AGR_2 has obj pref if 2nd person object
# Previously the other way round.

#=====
# Description:

# 1. Nondeterministic FST implementation in gawk.
# 2. The gawk script is enclosed in a shell script wrapper.
# 3. Non-deterministic processing is implemented by sequences of
#    function calls with the same input category, but outputs based
#    on specific subcategories (values).
# 4. Concord constraints are expressed using register variables in order to
#    avoid explosively multiplying the number of transitions:
#    subnumber, subperson, tense (later: other categories, e.g. negation)
# 5. Registers are used in order to generalise over dependencies (possibly
#    discontinuous) and avoid multiplying sub-paths in the FST.
# 6. The morphosyntactic model is non-cyclic.

#=====

# Input checks

# INPUTSTRING=$1
INPUTSTRING="subject tense verb object"
#=====
```

```

# gawk script

gawk '

# I/O

BEGIN {
    counter=0;
    q_subject("","","");
}

#####

# FST transitions

#####
# SUBJECT NODE: set SUBJECT registers

function q_subject(out, tones, gramtones, lextones, cats){

# Epsilon transition:
# Cut out in order to get pronouns only
    q_subjquant(out, tones, gramtones, lextones, cats " [SUBJECT:");
#
# Proper name:
#     subnumber="sing"; subperson="3";
#     q_concord(out " [Ime]", tones " [L H]", gramtones " [ ]", lextones " [L H]", cats "
[SUBJECT: [proname: " subnumber " " subperson " =Ime]]");

# Personal pronouns:

# First:
    subperson="1";
    subnumber="sing";
    q_concord(out " [ami]", tones " [L L]", gramtones " [L L]", lextones " [ ]", cats "
[SUBJECT: [pron: " subnumber " " subperson "=I]]");
    subnumber="plur";
    q_concord(out " [nnjIn]", tones " [L L]", gramtones " [L L]", lextones " [ ]", cats "
[SUBJECT: [pron: " subnumber " " subperson " =we]]");

# Second:
    subperson="2";
    subnumber="sing";
    q_concord(out " [afo]", tones " [L L]", gramtones " [L L]", lextones " [ ]", cats "
[SUBJECT: [pron: " subnumber " " subperson "=you]]");
    subnumber="plur";
    q_concord(out " [ndufo]", tones " [L L L]", gramtones " [L L]", lextones " [ ]", cats "
[SUBJECT: [pron: " subnumber " " subperson "=you]]");

# Third:
    subperson="3";
    subnumber="sing";
    q_concord(out " [anje]", tones " [L H]", gramtones " [L H]", lextones " [ ]", cats "
[SUBJECT: [pron: " subnumber " " subperson "=he-she-it]]");
    subnumber="plur";
    q_concord(out " [OmmO]", tones " [LH HL]", gramtones " [LH HL]", lextones " [ ]", cats "
[SUBJECT: [pron: " subnumber " " subperson "=they]]");
}

#####
# SUBJECT QUANTIFIER NODE

function q_subjquant(out, tones, gramtones, lextones, cats){
    subquant="yes";
    subperson="3";
    subnumber="sing";
    q_subjadj(out, tones, gramtones, lextones, cats);

    subnumber="plur";
    q_subjnounnum(out " [uwak]", tones " [L H]", gramtones " [L H]", lextones " [ ]", cats "
[quant: =many]");
    q_subjnounnum(out " [umiaN]", tones " [L H]", gramtones " [L H]", lextones " [ ]", cats "
[quant: =many]");
    q_subjnounnum(out " [usVk]", tones " [L H]", gramtones " [L H]", lextones " [ ]", cats "
[quant: =some]");
    q_subjnounnum(out " [ubaak]", tones " [H D]", gramtones " [H D]", lextones " [ ]", cats "
[quant: =some]");
    q_subjnounnum(out " [afIt]", tones " [L H]", gramtones " [L H]", lextones " [ ]", cats "
[quant: =all]");
}

```

```

        q_subjnounnum(out " [ukeet]", tones " [H L]", gramtones " [H L]", lextones " [ ]", cats "
[quant: =all]");
    }

#####
# SUBJECT NUMBER NODE

# Note mme -> mm /_V postlexically

function q_subjnounnum(out, tones, gramtones, lextones, cats){
    if (subnumber=="plur")
        q_subjadj(out " [mme]", tones " [L L]", gramtones " [L L]", lextones " [ ]", cats "
[agr: " subnumber " " subperson];
    else
        q_subjadj(out, tones, gramtones, lextones, cats " [agr: " subnumber " " subperson
"]");
}

#####
# SUBJECT ADJECTIVE NODE (Urua:87)

function q_subjadj(out, tones, gramtones, lextones, cats){
    q_subjnoun(out, tones, lextones, cats);
    q_subjnoun(out " [afja]", tones " [L H]", gramtones " [ ]", lextones " [L H]", cats " [adj:
=white]");
}

#####
# SUBJECT NOUN NODE

function q_subjnoun(out, tones, gramtones, lextones, cats){
    if (subquant=="yes")
        tone_1 = "H(L)";
    else
        tone_1 = "H";
# No, term of respect for older woman (Connell)
#   q_subjdet(out " [nne]", tones " [" tone_1 " L]", gramtones " [ ]", lextones " [ ]", cats "
=girl");
    q_subjdet(out " [edON]", tones " [H HL]", gramtones " [ ]", lextones " [H HL]", "[noun: "
cats " =sheep]");
}

#####
# SUBJECT DETERMINER NODE

function q_subjdet(out, tones, gramtones, lextones, cats){
#   if (subquant=="yes"){
#       q_concord(out, tones, gramtones, lextones, cats "]);
#   }
#   else {
#       q_concord(out " [emi]", tones " [L L]", gramtones " [L L]", lextones " [ ]", cats "
[det: prox =this]");
#       q_concord(out " [odo]", tones " [L L]", gramtones " [L L]", lextones " [ ]", cats "
[det: dist =that]");
#   }
}

#####
# AGR_2 OBJECT NODE

# Underlyingly SOV?

function q_concord(out, tones, gramtones, lextones, cats){
    obperson="1";
    obnumber="sing";
    q_subjectagr(out, tones, gramtones, lextones, cats);
    obnumber="plur";
    q_subjectagr(out, tones, gramtones, lextones, cats);

    obperson="2";
    obnumber="sing";
    q_subjectagr(out, tones, gramtones, lextones, cats);
    obnumber="plur";
    q_subjectagr(out, tones, gramtones, lextones, cats);

    obperson="3";
    obnumber="sing";
    q_subjectagr(out, tones, gramtones, lextones, cats);
    obnumber="plur";
    q_subjectagr(out, tones, gramtones, lextones, cats);
}

```

```

    }

#####
# AGR SWITCH NODE FOR TENSE - OBJECT CONCORD

function q_subjectagr(out, tones, gramtones, lextones, cats){

    tensecat="present";

    if (obperson=="2")
        q_verbprefix(out, tones, gramtones, lextones, cats);
    else q_agr(out, tones, gramtones, lextones, cats);

    tensecat="nonpresent";
    q_agr(out, tones, gramtones, lextones, cats);

}

#####
# AGR_1 SUBJECT CONCORD

function q_agr(out, tones, gramtones, lextones, cats){

    cats=" [verb: ";
    lextones=" [ ]";

    if (subperson=="1"){
        if (subnumber=="sing"){
            out=out " [n-]"; tones=tones " [H]"; gramtones "[H]";
            q_tense(out, tones, gramtones, lextones, cats);
        }
        else {
            out=out " [i-]"; tones=tones " [L]"; gramtones "[L]";
            q_tense(out, tones, gramtones, lextones, cats);
        }
    }
    if (subperson=="2"){
        if (subnumber=="sing"){
            out=out " [a-]"; tones=tones " [L]"; gramtones "[L]";
            q_tense(out, tones, gramtones, lextones, cats);
        }
        else {
            out=out " [e-]"; tones=tones " [L]"; gramtones "[L]";
            q_tense(out, tones, gramtones, lextones, cats);
        }
    }
    if (subperson=="3"){
        if (subnumber=="sing"){
            out=out " [a-]"; tones=tones " [H]"; gramtones "[H]";
            q_tense(out, tones, gramtones, lextones, cats);
        }
        else {
            out=out " [e-]"; tones=tones " [H]"; gramtones "[H]";
            q_tense(out, tones, gramtones, lextones, cats);
        }
    }
}

#####
# TENSE VERB PREFIX NODE: set TENSE inflection register

# proximal vs. distal tense (Connell)

function q_tense(out, tones, gramtones, lextones, cats){

    if (tensecat=="present"){
        q_verbprefix(out, tones, gramtones, lextones, cats);
    }

    else{
        tense=" [future]"; tensemorph=" [yaa-]";
        scale=" [prox]"; scalemorph=" [L H]"
        q_verbprefix(out tensemorph, tones scalemorph, gramtones scalemorph, lextones " [ ]",
cats tense scale);
        scale=" [dist]"; scalemorph=" [H L]"
        q_verbprefix(out tensemorph, tones scalemorph, gramtones scalemorph, lextones " [ ]",
cats tense scale);
    }
}

```

```

        tense=" [past]"; tensemorph=" [maa-]";
        scale=" [prox]"; scalemorph=" [L H]"
        q_verbprefix(out tensemorph, tones scalemorph, gramtones scalemorph, lextones " [ ]",
cats tense scale);
        scale=" [dist]"; scalemorph=" [H L]"
        q_verbprefix(out tensemorph, tones scalemorph, gramtones scalemorph, lextones " [ ]",
cats tense scale);
    }
}

#####
# AGR_2 VERB PREFIX NODE:

function q_verbprefix(out, tones, gramtones, lextones, cats){

    objcats= cats " [objagr: " obperson obnumber " ]";

    if (obperson=="2"){
        tones=tones " [ H]";
        gramtones=gramtones " [ H]";
        lextones=lextones " [ ]";
        if (obnumber=="sing") out=out " [u-]";
        else out=out " [i-]";
    }

    q_verb(out, tones, gramtones, lextones, objcats);
}

#####
# VERB NODE:

function q_verb(out, tones, gramtones, lextones, cats){
    q_object(out " [bed]", tones " [H]", gramtones " [ ]", lextones " [H]", cats "
=await]");
#    q_object(out " [bed]", tones " [L]", gramtones " [ ]", lextones " [L]", cats "
=push]");
}

#####
# OBJECT NODE: set OBJECT registers

function q_object(out, tones, gramtones, lextones, cats){

# Cut out in order to get pronouns only
# Epsilon transition:
#    q_objquant(out, tones, gramtones, lextones, cats " [OBJECT:");
#
# Proper name:
    obnumber="sing"; obperson="3";
    q_end(out " [Ime]", tones " [L H]", gramtones " [ ]", lextones " [L H]", cats " [OBJECT:
[proname: " obnumber " " obperson " =Ime]]");

# Personal pronouns (NOTE initial H tone on 1st, 2nd):

# First:
    if(obperson=="1"){
        if(obnumber=="sing"){
            q_end(out " [miin]", tones " [H L]", gramtones "[H L]", lextones " [ ]", cats "
[OBJECT: [pron: " obnumber " " obperson " =I]]");
#            q_end(out " [mien]", tones " [H L]", gramtones " [H L]", lextones " [ ]", cats "
[OBJECT: [pron: " obnumber " " obperson " =I]]");
        }
        if (obnumber=="plur"){
            q_end(out " [nnjIn]", tones " [H L]", gramtones " [H L]", lextones " [ ]", cats "
[OBJECT: [pron: " obnumber " " obperson " =we]]");
        }
    }

# Second:
    if(obperson=="2"){
        if(obnumber=="sing"){
            q_end(out " [fiin]", tones " [H L]", gramtones " [H L]", lextones " [ ]", cats "
[OBJECT: [pron: " obnumber " " obperson " =you]]");
#            q_end(out " [fien]", tones " [H L]", gramtones " [H L]", lextones " [ ]", cats "
[OBJECT: [pron: " obnumber " " obperson " =you]]");
        }
        if(obnumber=="plur"){
            q_end(out " [ndufo]", tones " [H L L]", gramtones " [H L L]", lextones " [ ]", cats "

```

```

[OBJECT: [pron: " obnumber " " obperson " =you]]];
    }
}

# Third:
    if(obperson=="3"){
    if(obnumber=="sing"){
    q_end(out " [anje]", tones " [L H]", gramtones " [L H]", lextones " [ ]", cats "
[OBJECT: [pron: " obnumber " " obperson " =he-she-it]]");
    }
    if(obnumber=="plur"){
    q_end(out " [OmmO]", tones " [LH HL]", gramtones " [L H]", lextones " [ ]", cats "
[OBJECT: [pron: " obnumber " " obperson " =they]]");
    }
    }
}

#####
# OBJECT QUANTIFIER NODE

function q_objquant(out, tones, gramtones, lextones, cats){
    obquant="yes";
    obperson="3";
    obnumber="sing";
    q_objnounnum(out, tones, gramtones, lextones, cats);

    obnumber="plur";
    q_objnounnum(out " [uwak]", tones " [L H]", gramtones " [L H]", lextones " [ ]", cats "
[quant: =many]");
    q_objnounnum(out " [umiaN]", tones " [L H]", gramtones " [L H]", lextones " [ ]", cats "
[quant: =many]");
    q_objnounnum(out " [usVk]", tones " [L H]", gramtones " [L H]", lextones " [ ]", cats "
[quant: =some]");
    q_objnounnum(out " [ubaak]", tones " [H D]", gramtones " [H D]", lextones " [ ]", cats "
[quant: =some]");
    q_objnounnum(out " [afIt]", tones " [L H]", gramtones " [L H]", lextones " [ ]", cats "
[quant: =all]");
    q_objnounnum(out " [ukeet]", tones " [H L]", gramtones " [H L]", lextones " [ ]", cats "
[quant: =all]");
}

#####
# OBJECT NUMBER NODE
# Note that mme -> mm /_V

function q_objnounnum(out, tones, gramtones, lextones, cats){
    if (obnumber=="plur")
        q_objadj(out " [mme]", tones " [L L]", gramtones " [L L]", lextones " [ ]", cats "
[agr: " subnumber " " subperson "]);
    else
        q_objadj(out, tones, gramtones, lextones, cats " [agr: " obnumber " " obperson "]);
}

#####
# OBJECT ADJECTIVE NODE (Urua:87)

function q_objadj(out, tones, gramtones, lextones, cats){
    q_objnoun(out, tones, lextones, cats);
    q_objnoun(out " [utu]", tones " [L L]", gramtones " [ ]", lextones " [L L]", cats " [adj:
=yellow]");
}

#####
# OBJECT NOUN NODE

function q_objnoun(out, tones, gramtones, lextones, cats){
    q_objdet(out " [unen]", tones " [H HL]", gramtones "[ ]", lextones " [H HL]" , cats " [noun:
" subnumber " " subperson " =chicken]");
}

#####
# OBJECT DETERMINER NODE

function q_objdet(out, tones, gramtones, lextones, cats){
    if (obquant!="yes"){
    # Corrected from L L (Connell)
    # Also q_end for erroneous q_concord
    q_end(out " [emi]", tones " [H L]", gramtones " [H L]", lextones " [ ]", cats " [det: prox
=this]]");
    q_end(out " [odo]", tones " [L L]", gramtones " [L L]", lextones " [ ]", cats " [det: dist

```

```

=that]]");
    } }

#####
# PREP PHRASE - insert here !!!
# e.g. k/e /ob/ot = on the hill

#####
# FINAL NODE:

function q_end(out, tones, gramtones, lextones, cats) {
#   print ++counter ":\nLEXTONE:" lextones "\n   TONE:" tones "\n   SEGM:" out "\n   MORPH: [s:"
cats "]\n";
#   print ++counter ":"
#   print " LEXTONE:" lextones;
#   print "GRAMTONE:" gramtones;
#   print "   TONE:" tones;
#   print "   SEGM:" out;
#   print "   MORPH: [s:" cats "];
#   print "";
#   }

#####

' |
tr [" "X" | tr "]" "X" |
sed "s/ */ /g" |
sed "s/X X/X+X/g" |
tr "X" " " |
sed "s/-//g"
#EOF=====

```