Introduction to Toolbox

Simon Musgrave
Monash University

Overview

- What does Toolbox do (why bother?)
- Components of Toolbox
- Getting started
- Getting data in and out

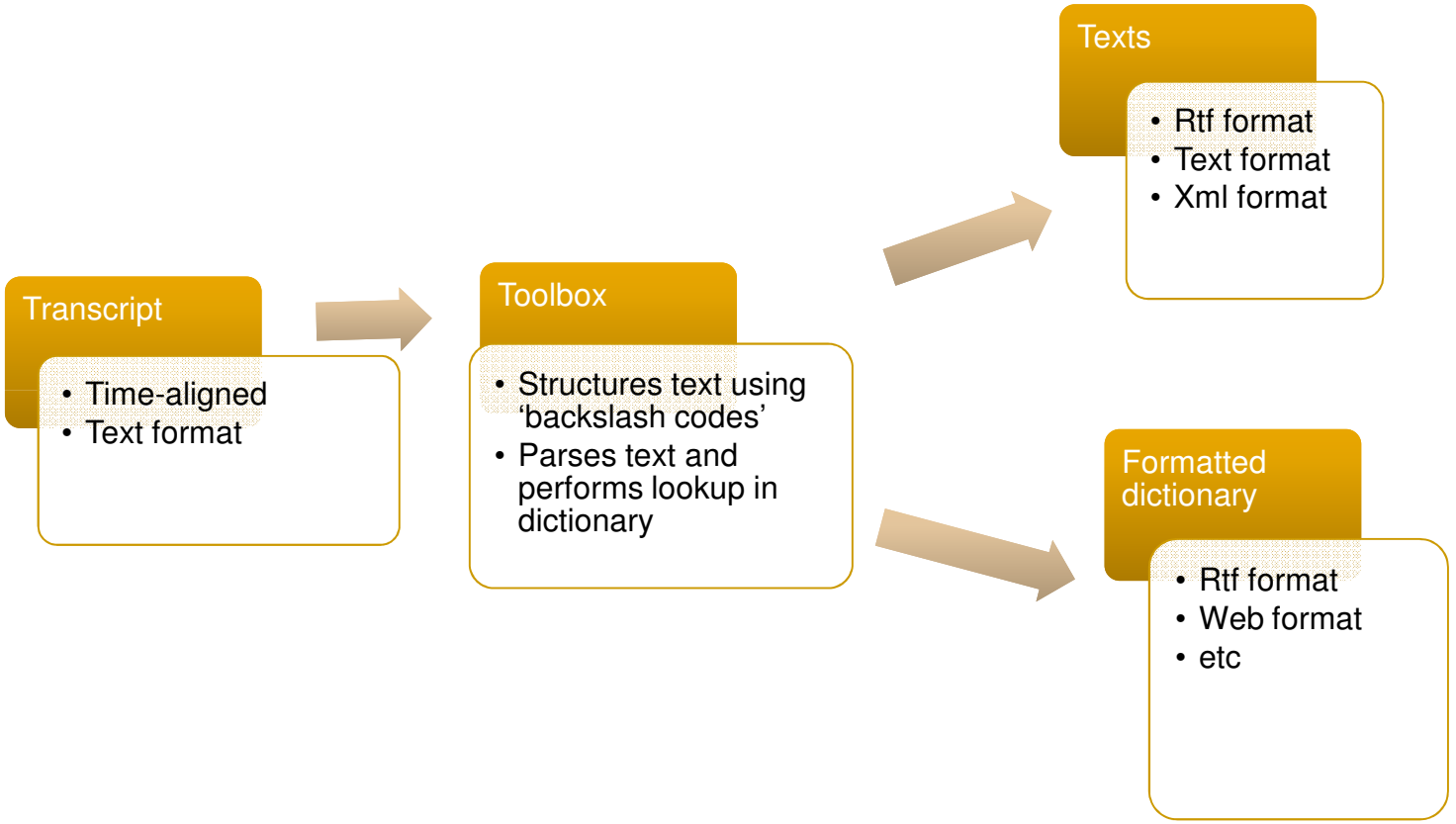
Toolbox

- What is it?
  - Data management program for language data.
  - Not a text editor.
  - Not really a database management system in the sense usually understood (it is not an implementation of a relational database system).

Toolbox

- What does it do?
  - Creates interlinear glossed text (IGT)
  - Creates a well-structured lexicon
  - Enforces explicit and exhaustive analysis
  - Provides useful import and (especially) export functions
  - Good search and filter capabilities aid analysis

Toolbox


Transcript
• Time-aligned
• Text format
Toolbox
• Structures text using 'backslash codes'
• Parses text and performs lookup in dictionary
Texts
• Rtf format
• Text format
• Xml format
Formatted dictionary
• Rtf format
• Web format
• etc

Why Toolbox?

- Good functionality
  - already detailed
- Choice of output possibilities
  - simple text files
  - well-formed xml
  - rtf files
  - html (at least for lexicon)
- Portability – simple file formats
  - underlying format is text file structured in specific way
  - data models are user-specified

Some disadvantages

- Data input not always easy
  - non-standard characters are problematic
  - but no other application is better!
- Manuals are not always easy to use
  - ditto for **Help** screens
- Way interlinear is stored means you need to revise often
- Everything is text, only weak data typing is possible
  - increases possibility of data entry errors not being detected

Components of Toolbox

- Project
- Definition files
  - database types
  - language encoding
- Database files
  - text files
  - lexicon files
  - (possibly others)

Project

- A project is the work unit in Toolbox
- A project file (.prj) is a shell file which holds information about what files are included in the project and what their properties are
- You have to open a project before you can do anything
- You can close Toolbox with an open project and it will be restored when you open the program again

Definition files – type 1

- Database type files store information about the structure of database files
  - what fields are required/allowed
  - what are the properties of those fields
  - how files of one type are linked to files of other types

Definition files – type 2

- Language encoding files store information about how to represent the languages you will work with
  - inventories of characters
  - relationships between groups of characters (case, sort order, variable sets)
  - display properties

Database files

- These are files which contain your data
- Each time you create a new one, you have to assign it one of the types previously defined
- At first sight, you may think:
  1 project = 1 text file + 1 lexicon file
- But we will see that there are good reasons to make things more complicated

Text files

- Contains transcribed textual data
- Metadata can be included in as much detail as desired
- Divided into chunks according to whatever practice you prefer
- Additional linear annotations are added automatically and stored

Lexicon files

- Contain data about lexical items (generally morphemes)
- Linked to text files in two ways
  - interlinear processes
  - Jump paths
- Built up as interlinearisation proceeds

  but also

- Source for automated interlinear processes

Getting started

- Database files

  need

- Type definitions

  which need

- Language encodings

Language Encoding

- You can do this in two ways:
  - Create a new encoding file then use the Language Encoding dialogues to work through all the bits and pieces which you want to do. (tricky)
  - Create a new encoding file, then open it in a text editor and manipulate it there. (easier)

Language Encoding

- Default language encoding works for
  - English
  - any other language which can be represented with ASCII set
- You need a language encoding for any language which does not meet this condition
- You should consider creating a language encoding for your object language even if not needed

What is in language encoding?

- The characters used to represent the language
- Toolbox is Unicode compliant
  - you can use any character
    - which is in Unicode
    - which is rendered by a font you have installed (problem discussed on next slide)
  - but entering characters is a problem (more later)
- Case pairs – you have to tell the program which pairs of symbols to treat as alphabetically equivalent, e.g. A = a for sorting and for parsing
- Sort order – you have to tell the program what order you want the alphabet to be in for sorting, e.g. if you use glottal stop, where should it be in the alphabet?

Rendering glyphs

- Problem comes with aligning interlinear
- Requires fixed-width font (not proportional)

\id 0305
\t palawane, pala'e, niere laha lainlain
\m palawan -i pala -i nier -i laha lain DUP
\g clove.tree -3.POSS nutmeg.tree -3.POSS coconut -3.POSS and other INTS
\p N -POSS N -POSS N -POSS CONJ ? SUFF

```
\id 0305
\t palawane,          pala'e,             niere          laha lainlain
\m palawan    -i      pala        -i      nier    -i     laha lain  DUP
\g clove.tree -3.POSS nutmeg.tree -3.POSS coconut -3.POSS and  other INTS
\p N          -POSS   N           -POSS   N       -POSS   CONJ ?     SUFF
```

Unicode fixed width

- Best option currently is GNU Unifont
  http://unifoundry.com/unifont.html
- Glyphs for every printable code point in the Unicode 5.1 Basic Multilingual Plane (BMP)
- First 65,536 code points of the Unicode space
- But not the prettiest font out there……

GNU Unifont

- Previous example:

```
\id 0305
\t palawane,          pala'e,             niere           laha lainlain
\m palawan    -i      pala        -i      nier    -i      laha lain  DUP
\g clove.tree -3.POSS nutmeg.tree -3.POSS coconut -3.POSS and  other INTS
\p N          -POSS   N           -POSS   N       -POSS   CONJ ?     SUFF
```

What is in language encoding?

- Fonts – you can specify the on-screen characteristics of each language encoding which you use. This is useful to make the screen easier to read.
- You can also specify screen characteristics for fields when you define a database type, overriding or modifying the language settings
- Neither of these options affects the presentation of data when you export to the Multi Dictionary Formatter (MDF) – MDF uses its own font settings regardless.

Language Encoding

- Variables – you have to specify which characters will be included in which sets of variables. The default groupings which are set in the program are:
  - Everything
  - Lower case
  - Upper case
  - Vowels
  - Consonants
  - Nasals
  - Punctuation
  - Digits
- These variable definitions are used for wildcards in searches, and for specifying some morphological processes in parsing.

Editing a language encoding

- Remember to add new characters to each section
  - case pairs
  - sort order
  - variables

Database types

- Relational database (e.g. Access)
  - One field (or a combination of fields) must have data and function as unique identifier
  - Every field specified in the definition occurs in every record
  - Every field specified in the definition occurs only once in each record
- Non-relational database (Toolbox)
- One field specified in the definition must occur in every record as unique identifier – the **record marker** (\itm, \ref, …)
  - Other fields can occur many times in each record

Database types – Markers

- Toolbox database files are a special sort of text file:
  - Standard Format Marker (SFM) files (also known as ‘backslash’ codes)
- Each field has the structure:
  - Marker – ‘\’ character + identifying string + obligatory space
  - Text content – whatever is stored in the field
  - Return – indicates end of field
- A new record starts with the occurrence of a record marker field (e.g., \lx)
- NB – database definitions and language encodings are also SFM files

Markers and fields

- Marker labels should be:
  - short (\pos better than \part_of_speech)
  - mnemonic
- You should provide text in the Description section of the marker definition dialog
  - useful for potential other users
  - useful for you after lapse of time
- Setting font/colour properties for common fields can make screen viewing easier

Range Set

- Range set – you can specify that a field will only contain one of a set of specified values, useful or e.g. part of speech, semantic domains
- When you have some data, Toolbox can automatically create a set of values for you from what is already entered
- You must remember to check the “Use a Range Set” box in the Marker properties section of the Database Type dialogue

Database types – dates

- Date stamping – you can include a date field (usually \dt) in your database and enable automatic date stamping
- Date stamping happens on insertion of a record and then again whenever a record is edited – if you want to preserve the information about when you first entered a record, this has to be done manually
- You have to create a date field before you can enable date stamping

Text file fields

- Minimally:
  - record marker field – NEVER has text in it, use reference numbers
  - text field
  - morphemic analysis
  - morpheme-by-morpheme gloss
  - free translation(s)
- Anything else is optional – obviously some elements are highly desirable

Text file fields - interlinear

- Toolbox has a (partially) automated set up for interlinear processes
- This assumes fields named:
  - \t – text
  - \m – morphemes
  - \g – morphemic gloss
  - \p – part of speech
- You can change these defaults at set-up, but it is easy to set up your text file definition using them

Lexicon file definition

- Multi Dictionary Formatter is very useful feature of Toolbox
- Using the preset MDF lexicon file (one of them) is worth considering
- But there are 100+ fields defined in this type – you probably don't need all of them!
  - You can delete fields from this definition
  - You can create a MDF Lite from scratch using the same marker names for the fields you want

Lexicon essentials

- A field to store the basic lexical form
  - we can assume this is unique, so it can be the record marker
- Field for alternate forms
- Field for gloss to be used in interlinear
- Field for more extensive definition
- Others are optional
  - again several elements are very desirable

Structuring databases

- Toolbox is not a relational database
- But some structure can be imposed:
  - hierarchies
  - following fields
  - segmentation of texts

Hierarchies

- Important in lexicon
- Used to organise:
  - homonyms
  - multiple senses
- Pre-set in MDF database type – another reason for using that definition

Following fields

- When you define a field marker, you can specify that another field follows it
- Consequences at data entry:
  - type data in field 1
  - hit ENTER
  - field 2 is added automatically
- Useful to set up a template for fields you want in every entry

Segmenting text

- Documentation assumes:
  - all texts in one database file
  - each text is one record in the file
  - lines of text have sub-references within the record
- Automatic segmentation feature works on this basis (example later)
- But this approach can pose problems in accessing data

Search v. filter

- Toolbox allows selection of data in two ways:
  - Search – move sequentially through each instance of search term in file
  - Filter – recover all records in file which contain search term as a subset
- Search works well on single text file
  - You get to see every instance you are interested in
- Filter works less well on single text file
  - If a whole text is a record, that is what is recovered (not e.g. individual clauses)

Is there a compromise?

- Multiple text files can be useful
- One master file with all texts as separate records
  - good for searching
- Individual files with clauses / intonation units as records
  - good for filtering (but you have to use the filter in each file)

Toolbox file format

- A Toolbox file
  - Requires a header

    _sh v3.0 400 ElanExport

    _DateStampHasFourDigitYear

Toolbox file format

Then the first record marker:

- e.g., Lexical database would have \lx
- Text database would have \ref (or whatever you determine)

So, to create a valid Toolbox file, paste the header into a text file and Toolbox will open it.

Original text

- Ore, ipiatlak malen kin 1980, malen tuksat independent. Teni esum̃ Erakor ruta sapot ki independent mau. Go rupreg tete problem p̃ur, rupreg tete nawesien nen ipi tap leg mau taon. Rupak taon rupuetlu flaik nen kin kafman ipsi ito taon. Rupuetlua ipak etan. Go tete krup rumpaki tanmaet Radio Vanuatu. Pregi tiawi laap rumtak, go rufit pan, tete rupan los elau, esan esto ni Fung Kuei. Tete rupan los ntas elau sa. Tete rufit mai pak Radio Vanuatu, rufit mai pak lakun. Go polis rupuetsok tete go ru-. Rusmolir nasum̃ malik. (NT) Me ni naur Erakor, ipiatlak tete muf ni natkon ne? (WW) Natkon ne? Naur Erakor? Ore naur Erakor gar ruta sapot independent malnen mau. Gar rupan kerkerai nlaken nam̃er ni Franis ruto sursrir go rupregi gar ruskot nam̃er tonanre ni Franis. Go rupreg ruta sapot ki independen mau. Me inrok nen, malen kin tusat independent, go rupo sapot ki independent. Go mees tupo leka tufri. Namroan nen kin upiatlaken malpei kin umalki independent, umroki na isa me mees upo pam̃ori na ipo iwi. Nlaken ipiatlak malnen ipitlak mal ni kolonialism go rupreg sa ki namroan ni natam̃ol. Ruto sursur natam̃ol, rupregi natam̃ol rusapot kir. Me inrok knen go upo pam̃ori na isa. Go mees uipe free. Utae preg tenmatun nen kin umurin, nlaken uipe slat independent nigmam.

Text plus header

_sh v3.0 400 ElanExport
_DateStampHasFourDigitYear

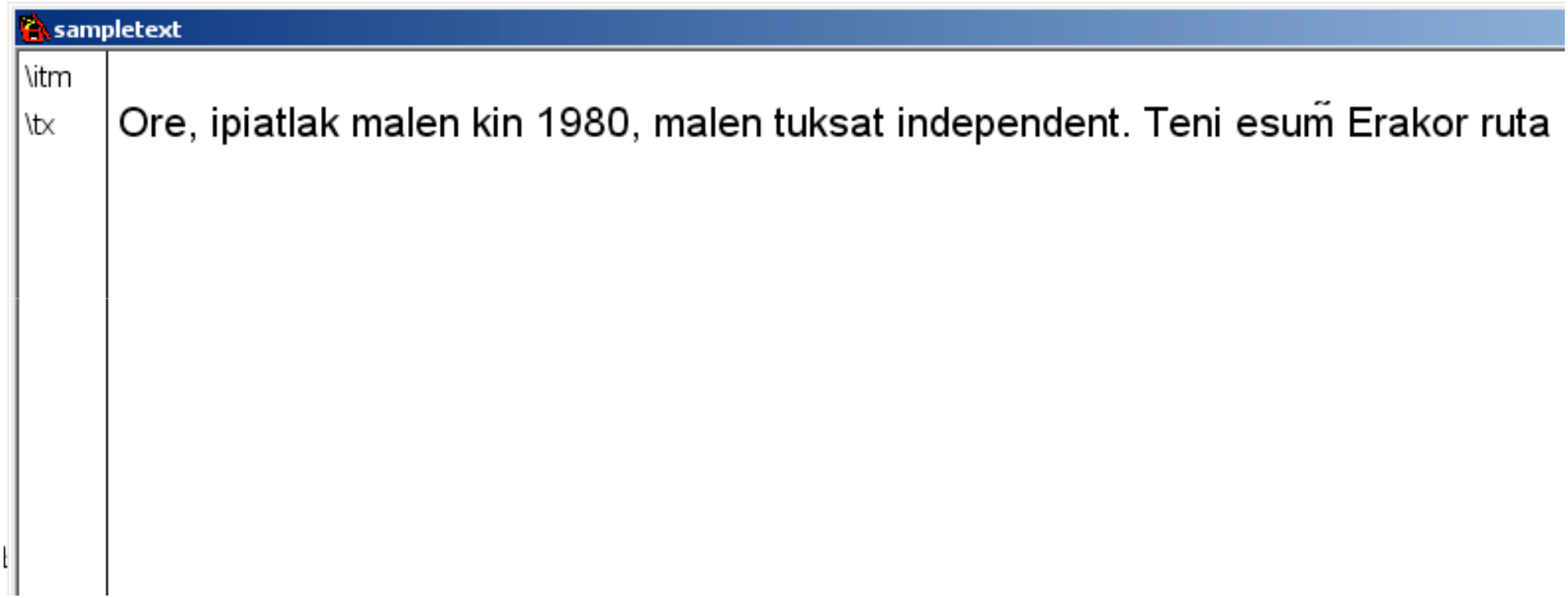
Ore, ipiatlak malen kin 1980, malen tuksat independent. Teni esum̃ Erakor ruta sapot ki independent mau. Go rupreg tete problem p̃ur, rupreg tete nawesien nen ipi tap leg mau taon. Rupak taon rupuetlu flaik nen kin kafman ipsi ito taon. Rupuetlua ipak etan. Go tete krup rumpaki tanmaet Radio Vanuatu. Pregi tiawi laap rumtak, go rufit pan, tete rupan los elau, esan esto ni Fung Kuei. Tete rupan los ntas elau sa. Tete rufit mai pak Radio Vanuatu, rufit mai pak lakun. Go polis rupuetsok tete go ru-. Rusmolir nasum̃ malik. (NT) Me ni naur Erakor, ipiatlak tete muf ni natkon ne? (WW) Natkon ne? Naur Erakor? Ore naur Erakor gar ruta sapot independent malnen mau. Gar rupan kerkerai nlaken nam̃er ni Franis ruto sursrir go rupregi gar ruskot nam̃er tonanre ni Franis. Go rupreg ruta sapot ki independen mau. Me inrok nen, malen kin tusat independent, go rupo sapot ki independent. Go mees tupo leka tufri. Namroan nen kin upiatlaken malpei kin umalki independent, umroki na isa me mees upo pam̃ori na ipo iwi. Nlaken ipiatlak malnen ipitlak mal ni kolonialism go rupreg sa ki namroan ni natam̃ol. Ruto sursur natam̃ol, rupregi natam̃ol rusapot kir. Me inrok knen go upo pam̃ori na isa. Go mees uipe free. Utae preg tenmatun nen kin umurin, nlaken uipe slat independent nigmam.

Text plus header plus record marker plus first field marker

_sh v3.0 400 ElanExport
_DateStampHasFourDigitYear

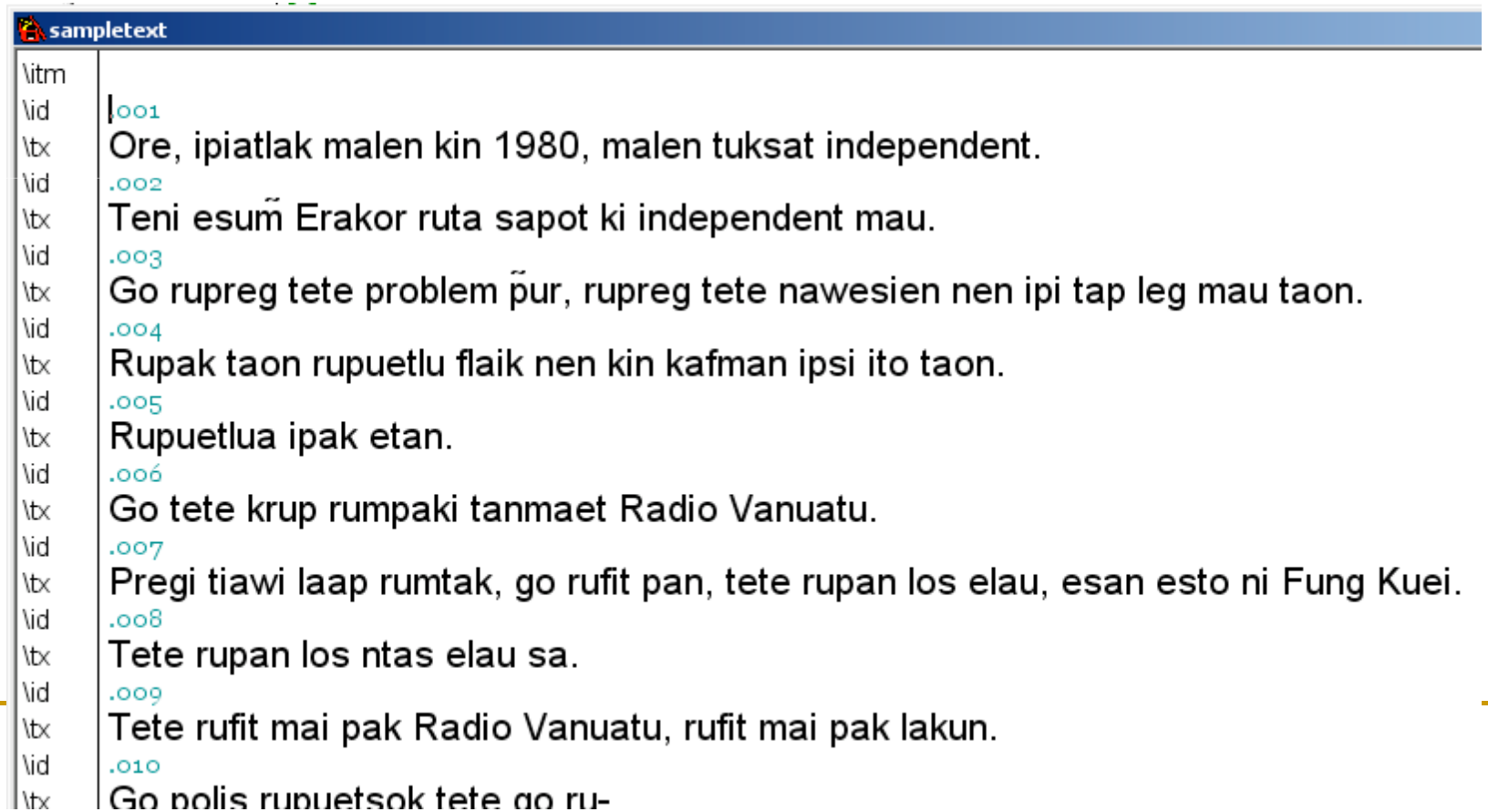
\itm
\tx Ore, ipiatlak malen kin 1980, malen tuksat independent. Teni esum̃ Erakor ruta sapot ki independent mau. Go rupreg tete problem p̃ur, rupreg tete nawesien nen ipi tap leg mau taon. Rupak taon rupuetlu flaik nen kin kafman ipsi ito taon. Rupuetlua ipak etan. Go tete krup rumpaki tanmaet Radio Vanuatu. Pregi tiawi laap rumtak, go rufit pan, tete rupan los elau, esan esto ni Fung Kuei. Tete rupan los ntas elau sa. Tete rufit mai p̃ak Radio Vanuatu, rufit mai pak lakun. Go polis rupuetsok tete go ru-. Rusmolir nasum̃ malik. (NT) Me ni naur Erakor, ipiatlak tete muf ni natkon ne? (WW) Natkon ne? Naur Erakor? Ore naur Erakor gar ruta sapot independent malnen mau. Gar rupan kerkerai nlaken nam̃er ni Franis ruto sursrir go rupregi gar ruskot nam̃er tonanre ni Franis. Go rupreg ruta sapot ki independen mau. Me inrok nen, malen kin tusat independent, go rupo sapot ki independent. Go mees tupo leka tufri. Namroan nen kin upiatlaken malpei kin umalki independent, umroki na isa me mees upo pam̃ori na ipo iwi. Nlaken ipiatlak malnen ipitlak mal ni kolonialism go rupreg sa ki namroan ni natam̃ol. Ruto sursur natam̃ol, rupregi natam̃ol rusapot kir. Me inrok knen go upo pam̃ori na isa. Go mees uipe free. Utae preg tenmatun nen kin umurin, nlaken uipe slat independent nigmam.

Open file in Toolbox



Segmenting

- Toolbox will segment and number a text automatically, dividing at punctuation as specified



Linking text and lexicon - interlinear

- The crucial functionality in Toolbox is automated generation of interlinear
- Links between text files and lexicon files have to be set up
- Two types of process:
  - Parse – segment text line into morphemes using list stored in lexicon
  - Look-up – retrieve information from lexicon based on morphemes parsed

Parsing - basic principles

- Toolbox parses by looking for possible analyses of a word form based on information stored in one or more lexical databases
- Large substrings are preferred to short substrings
  - 'throughout' will be found rather than a parse of 'through' and 'out'
- The parse process works from the outside to the inside – prefixes and suffixes are parsed before roots

Basic principles

- Where more than one parse is possible, the user is prompted to resolve the ambiguity
- The output is whatever information has been requested for each item from the lexicon, all formatted to preserve alignment
- Where no parse is found, a string of meaningless symbols (****** by default) is returned
- Text files underlie all processes

Processes – look-up

- Look-up processes are straightforward:
- Shoebox has found the required morpheme in the lexicon and has only to retrieve relevant information
- Only one lexicon will be relevant
- The input is whatever field the base form of lexical items is stored in
- The output is whatever other field you want information from e.g. gloss, word class

Processes - parse

- This is where complications come up
- You have to specify the location of any variant forms which should all resolve to a single morpheme – usually the \a field
- Typically, in the early stages of work on data, you will have many \a entries, often several for a single morpheme
- As you know more about a language, these should reduce – your transcription will be more consistent etc.
- But it is always possible that variants will remain
- You must specify each lexicon field that may contain forms of the morpheme
- And you must specify the field which holds the form to be returned (normally \lx)

Listing affixes

- Lexical entries for affixes use hyphens to specify the attachment of the affix
  - -xxx means that the morpheme is a suffix
  - xxx- means that the morpheme is a prefix
  - -xxx- means that the morpheme is an infix
- Shoebox does not parse circumfixes or internal modification (but there are possible work-arounds – ask me or Nick)

Reduplication

- Shoebox can parse reduplication
- All morphemes analysed as reduplication are entered in the lexicon with the letters ‘dup’ in their \lx field e.g. dupCV- could be the entry for a reduplicating CV- prefix
- Use \a fields to specify the actual forms of the morpheme
- Use variables (cons, vowel) to generalise
- The prefix above would be listed:
  \lx dupCV-
  \a [cons][vowel]-

Ambiguity

- Where more than one parse is possible, Shoebox will produce a dialogue asking you to choose
- The ambiguity can come about because of homophonous morphemes
- Or the ambiguity can come about because Shoebox can segment the string in more than one way
- It is not uncommon for several choice to be necessary to get a single complex word form to parse
- If nothing else works, you can force a parse by inserting dashes in the text line (and erasing them afterwards!)

Reducing ambiguity

- Many cases where Shoebox asks for resolution of an ambiguity are caused by the homophonous (or homographic) morphemes e.g.
  - –s in English must be listed twice in the lexicon, as a plural marker and as a 3rd person singular marker
  - But one attaches to nouns and one attaches to verbs
- Shoebox can be given this information by using word formulas, and this will prevent both parses being offered every time –s is encountered

When parsing fails

- When Shoebox cannot parse a string, it returns a series of ***** characters (or you can set a different character)
- This means that there is not enough information in the lexicon(s) for the string to be parsed
- You need to enter more information into the lexicon
- Setting the Jump Path is now useful!

Linking text and lexicon - jumping

- You can specify Jump links between texts and lexicons
- If a Jump Path is set, you can select a string in the text file, press Ctrl+J and jump to the lexicon, either:
  - To a new entry if the selected string doesn't match any existing entry
  - Or to an existing entry
- Jump Path is set in the database properties dialogue
- If you are using more than one lexicon
  - they can all be entered in the Jump Path
  - you can specify the order in which they will be accessed
  - you will get to choose which one you want each time

Multiple lexicons?

- Although using one lexicon might seem simplest, often it makes sense to use more than one
- Where there are many loan words used, especially if they are mainly from one other language, it may be easier to keep the languages in separate lexica
  - You won't have to separate the loans before you turn the main lexicon into a dictionary – you could do this also with filtering
  - Possibly, you will store less information about loan words (e.g. I list Malay loans without morphological analysis)
  - I also keep proper names in a separate lexicon
- Also worth considering: a 'junk' lexicon
  - Means that e.g. false starts, hesitations, errors can be parsed
  - 'junk' doesn't clutter your working lexicon

Getting data in and out

- Toolbox files are
  - Well-structured
  - Relatively simple
- Importing and exporting is quite flexible
- We already saw that adding a header allows any text file to open
- There are other possibilities….

Entering non-standard characters

- Inputting non-ASCII characters is a problem!
- One solution is to use a keyboard mapping utility – Tavultesoft Keyman is recommended for Toolbox
- A language encoding can have a keyboard mapping associated to it
- Problems:
  - Not free
  - Creating keyboard mappings is not simple
  - IPA is there but all symbols in one mapping (never all needed for one language)

An alternative

- Enter text in a Unicode editor, copy into Toolbox
- Good tool is Sharmahd Unipad
- Free version handles small amounts of text
- User-defined keyboards by drag-and-drop
- On-screen keyboard or remapping of actual keyboard

Conversion of Transcriber/Elan output to Toolbox

Upload Transcriber file, set markers and download converted data

http://linguisticsoftwareconverters.zong.mine.nu/

Transcriber import table:

http://www.sil.org/computing/toolbox/TranscriberImport.zip

ELAN has an Toolbox option on the **Export as…** menu

[You can import Toolbox files into ELAN, but your tier structure has to correspond to the structure of the Toolbox file]

Export possibilities

- Toolbox will export records or files in various formats
- Preset options are Rich Text Format, Standard Format (text file) and xml
- The RTF export is formatted according to the view you see on screen – if you have set a wide screen (using Reshape), you will get messy wrapping
  - But this allows you to set the wrap to the page width you require

Xml export

The xml export in Toolbox wraps each field in tags and marks the relationship between interlinear lines

```
<idgroup>
    <id>001</id>
        <aud>WitGoKusu.mp4 4.877 10.963</aud>
        <txgroup>
        <tx>Amurin gag puserek, Nick, kafo gag</tx>
            <mrgroup>
                <mr>a=</mr>
                <mg>1sgRS=</mg>
            </mrgroup>
            <mrgroup>
                <mr>mur</mr>
                <mg>want</mg>
            </mrgroup>
            <mrgroup>
                <mr>-i</mr>
                <mg>-TS</mg>
            </mrgroup>
             [etc]
        </txgroup>
<fg>I want to tell you, Nick, I'll tell you.</fg>
```

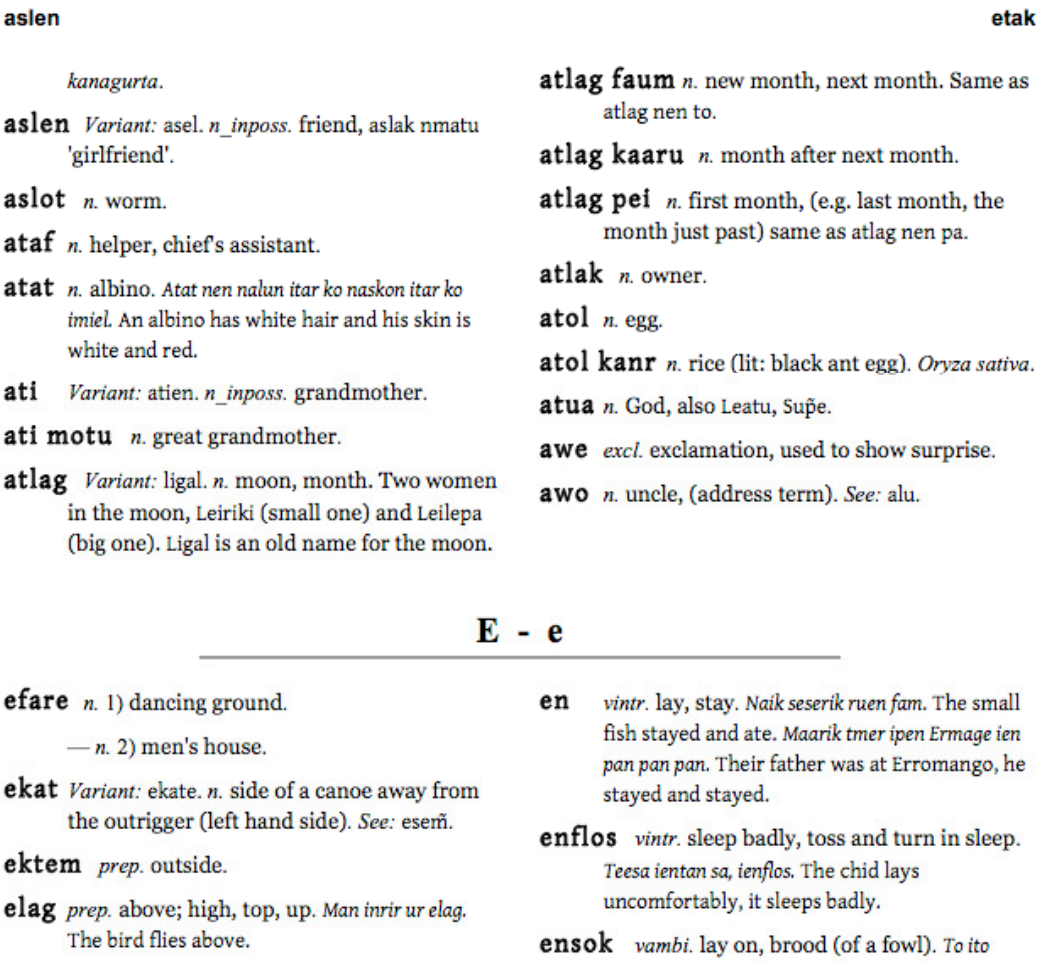
Multi Dictionary Formatter

- This is one of the best reasons for using Toolbox
- It gives you a dictionary in an good printable format with basically one click
  - You have to choose some options, but the defaults are fine
  - Changes only necessary if you want a trilingual output

MDF output

MDF - Multi-Dictionary Formatter

Formatted rtf document



kanagurta.

aslen *Variant:* asel. *n_inposs.* friend, aslak nmatu 'girlfriend'.

aslot *n.* worm.

ataf *n.* helper, chief's assistant.

atat *n.* albino. *Atat nen nalun itar ko naskon itar ko imiel.* An albino has white hair and his skin is white and red.

ati *Variant:* atien. *n_inposs.* grandmother.

ati motu *n.* great grandmother.

atlag *Variant:* ligal. *n.* moon, month. Two women in the moon, Leiriki (small one) and Leilepa (big one). Ligal is an old name for the moon.

atlag faum *n.* new month, next month. Same as atlag nen to.

atlag kaaru *n.* month after next month.

atlag pei *n.* first month, (e.g. last month, the month just past) same as atlag nen pa.

atlak *n.* owner.

atol *n.* egg.

atol kanr *n.* rice (lit: black ant egg). *Oryza sativa*.

atua *n.* God, also Leatu, Supe.

awe *excl.* exclamation, used to show surprise.

awo *n.* uncle, (address term). *See:* alu.

E - e

efare *n.* 1) dancing ground.

— *n.* 2) men's house.

ekat *Variant:* ekate. *n.* side of a canoe away from the outrigger (left hand side). *See:* esem̃.

ektem *prep.* outside.

elag *prep.* above; high, top, up. *Man inrir ur elag.* The bird flies above.

en *vintr.* lay, stay. *Naik seserik ruen fam.* The small fish stayed and ate. *Maarik tmer ipen Ermage ien pan pan pan.* Their father was at Erromango, he stayed and stayed.

enflos *vintr.* sleep badly, toss and turn in sleep. *Teesa ientan sa, ienflos.* The chid lays uncomfortably, it sleeps badly.

ensok *vambi.* lay on, brood (of a fowl). *To ito*

Lexique Pro

- Lexique Pro is a freeware tool distributed by SIL via www.lexiquepro.com
- It is intended to produce versions of lexicons for distribution to people who are not Shoebox users
- The program makes a version which is well-formatted for on-screen viewing
- It also makes an executable file (.exe) to distribute the lexicon to other people – this will install a run-time version of Lexique Pro and the database extracted from your lexicon onto another persons computer
- You can also export your lexicon as web pages
- http://paradisec.org.au/SELexicon/index-english/main.htm

South Efate

This dictionary is a work in progress and is presented here in the knowledge that there are many errors that still need correcting. This edition can be cited as: Nicholas Thieberger (2007-12-03) Dictionary of South Efate (http://paradisec.org.au/SELexicon/index-english/main.htm). This work is the result of a collaborative project between speakers of South Efate, mainly in Erakor village, Efate, Vanuatu, and Nicholas Thieberger who has written a grammar of the language. This version included examples spoken by Endis Kalsarap and Manuel Wayane. Sound files are playable by clicking on many of the headwords in the dictionary, unfortunately they open a new

English - South Efate

a b c d e f g h i j k l m n o p q r s t u v w y

A – a

abalone	**kaiaraskei**
abort	**kispun**
above	**elag**
absent	**puel**
·	**lalu**
accompany	**plak**
·	**ptaan**
act	**gien**
active	**polkirkir**
adam's apple	**npatnkafik**
add	**skar**
admire	**lewi**
adopt	**p̄as**
adult	**p̄afp̄of**
adze	**took**
·	**kram p̄el**
·	**limur**
aeroplane	**man ni nmalfa**
after	**me**
·	**ntakun**
afterbirth?	**naal ni teesa**
afternoon	**kotfan**

nafsan *n.* 1 • story.

2 • language.

nafsik *n.* flesh (of fruit or animal).

nafsup̄narun *Variant:* **nafsup̄nar**. *n_inposs.* elbow.

nafte *interog.* what. *See:* **ku**.

naftogin *Variant:* **naftog**. *n_inposs.* stem of a plant.

naftourwen *n.* wedding. Today also commonly **nlakwen**. *See:* **lak**.

naftuan *n.* gift. *See:* **tu** 'give'.

nafum̄ *Variant:* **nafum̄e**; **nafum̄kas**. *n.* flower. *See:* **fum̄**.

nafusrekwen *Variant:* **nafuserekwen**. *n.* story.

nafut *n.* bubble.

nagi *n.* tree, strong ironwood, grows on rocks by the sea, trunk can be 2 ft in diameter, succulent leaf and small flower with yellow centre. Medicinal uses. Leaves can be used to make your teeth strong. Also use it to decorate your house.

nagien *Variant:* **nagi**. *n_inposs.* name.

Getting the software

- www.sil.org/computing/toolbox
- Current (3/2013) version is 1.5.9
  - Windows 2000, XP, Vista
  - Mac and Linux with Windows emulation
  - Development has ceased
- Sources of advice and information:
- http://www.sil.org/computing/toolbox/links.htm
- http://www.linguistics.unimelb.edu.au/thieberger/CALW/Shoebox5.htm
- Do the tutorials that come with Toolbox