TshwaneLex
A State-of-the-Art Dictionary Compilation Program

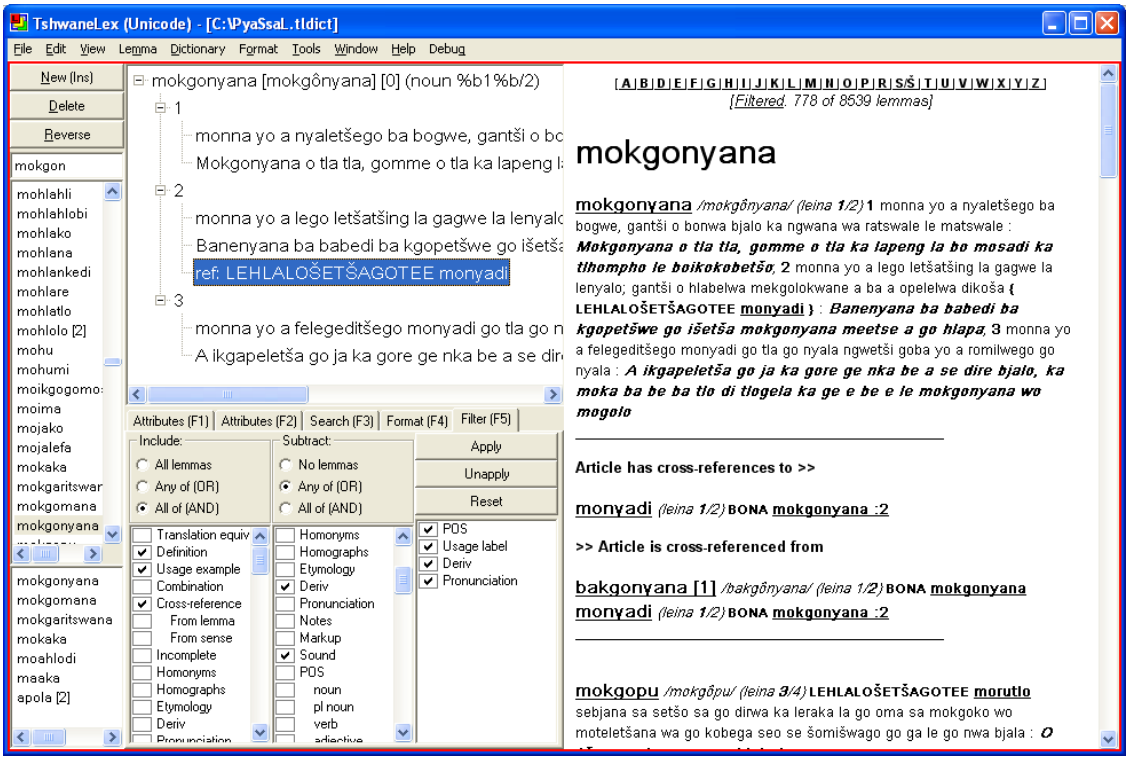
David Joffe[#], Gilles-Maurice de Schryver[°,#]
TshwaneDJe Human Language Technology, PO Box 299, Wapadrand 0050, South Africa[#] (david.joffe@tshwanedje.com) || Department of African Languages and Cultures, Ghent University, Rozier 44, 9000 Ghent, Belgium[°] (gillesmaurice.deschryver@UGent.be)

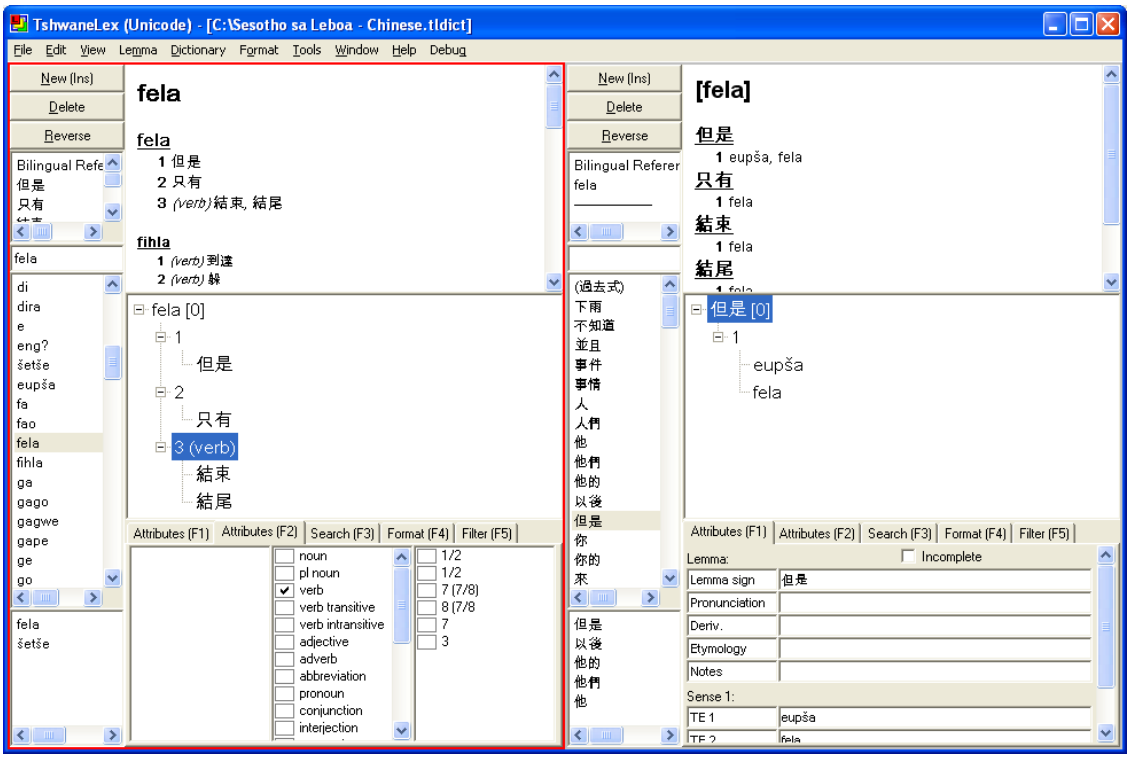
Abstract
A new state-of-the-art dictionary compilation program called *TshwaneLex* is briefly introduced. Core features include user-friendliness, automatic cross-reference tracking, an advanced compare/merge function, various levels of customisation, and provision for virtually all the world's languages by means of full Unicode support. TshwaneLex requires a PC with Windows 98 / Me / 2000 / XP. For full Unicode support, Windows 2000 / XP is recommended. Storage space and memory requirements are dependant on the size of the dictionary project.

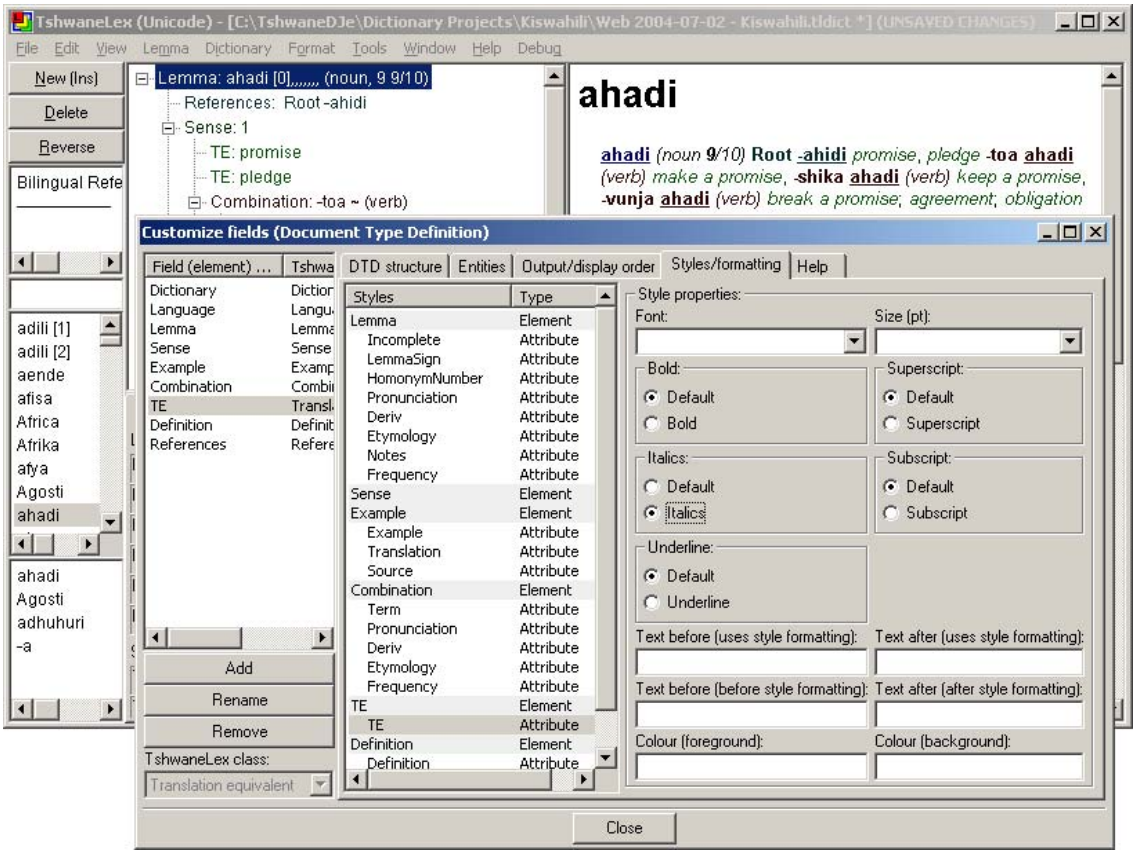
1. Introduction & Background

It is somewhat contradictory to note that although thousands of dictionaries are being compiled at any given point in time, dedicated tools to assist lexicographers are not readily available. Large dictionary publishers such as Collins simply build their own in-house systems; others such as Longman or Oxford base their latest lexicography environments on rather complex million-dollar applications that are specifically customised for them (in this case the *DPS* by IDM, cf. McNamara 2003). SIL International (http://www.sil.org/) have developed some lexicography tools that are used by field linguists only; these include *Shoebox* and *Toolbox*, *LinguaLinks*, *FieldWorks*, etc. There have been attempts to produce off-the-shelf lexicographic workbenches in the past – examples include *GestorLEX* by TEXTware (http://www.textware.dk/) and *Onoma* by Lexilogik (Ridings 2003) – yet these products still needed quite some customisation and investments before they could be used in any real-world dictionary project. Calls for dictionary writing systems are also recurring periodically on various lexicography and corpora mailing lists, but remain largely unanswered. The result of this state of affairs is that too many lexicographers – freelance compilers as well as large dictionary teams – end up storing their lexicographic gems in simple word processors or databases of their own making.

Against this background the need was felt at TshwaneDJe HLT (http://tshwanedje.com/) to build a robust, truly off-the-shelf, modern dictionary compilation program with which any dictionary maker(s) could compile reference works for any language(s). The new software is known as *TshwaneLex*, and is already in use for the compilation of monolingual, bilingual and hybrid dictionaries at several of South Africa's National Lexicography Units, and copies have been acquired by Macmillan, as well as by a number of lexicography projects based in Europe. A general overview and computational aspects can be found in Joffe et al. (2003a and 2003b respectively). A lexicographic perspective of TshwaneLex follows, and starts with three screenshots, illustrating, *inter alia*, Unicode support and the use of a DTD.


TshwaneLex (Unicode) - [C:\PyaSsaL.tldict]
File Edit View Lemma Dictionary Format Tools Window Help Debug
New (Ins)
Delete
Reverse
mokgon
mokgonyana [mokgônyana] [0] (noun %b1%b/2)
1
monna yo a nyaletšego ba bogwe, gantši o bc
Mokgonyana o tla tla, gomme o tla ka lapeng l
2
monna yo a lego letšatšing la gagwe la lenyalc
Banenyana ba babedi ba kgopetšwe go išetš
ref: LEHLALOŠETŠAGOTEE monyadi
3
monna yo a felegeditšego monyadi go tla go n
A ikgapeletša go ja ka gore ge nka be a se dir
[A|B|D|E|F|G|H|I|J|K|L|M|N|O|P|R|S/Š|T|U|V|W|X|Y|Z]
[Filtered: 778 of 8539 lemmas]
mokgonyana
mokgonyana /mokgônyana/ (leina 1/2) 1 monna yo a nyaletšego ba bogwe, gantši o bonwa bjalo ka ngwana wa ratswale le matswale : Mokgonyana o tla tla, gomme o tla ka lapeng la bo mosadi ka tlhompho le boikokobetšo; 2 monna yo a lego letšatšing la gagwe la lenyalo; gantši o hlabelwa mekgolokwane a ba a opelelwa dikoša (LEHLALOŠETŠAGOTEE monyadi) : Banenyana ba babedi ba kgopetšwe go išetša mokgonyana meetse a go hlapa; 3 monna yo a felegeditšego monyadi go tla go nyala ngwetši goba yo a romilwego go nyala : A ikgapeletša go ja ka gore ge nka be a se dire bjalo, ka moka ba be ba tlo di tlogela ka ge e be e le mokgonyana wo mogolo
Article has cross-references to >>
monyadi (leina 1/2) BONA mokgonyana :2
>> Article is cross-referenced from
bakgonyana [1] /bakgônyana/ (leina 1/2) BONA mokgonyana
monyadi (leina 1/2) BONA mokgonyana :2
mokgopu /mokgôpu/ (leina 3/4) LEHLALOŠETŠAGOTEE morutlo
sebjana sa setšo sa go dirwa ka leraka la go oma sa mokgoko wo moteletšana wa go kobega seo se šomišwago go ga le go nwa bjala : O
Attributes (F1) Attributes (F2) Search (F3) Format (F4) Filter (F5)
Include:
All lemmas
Any of (OR)
All of (AND)
Subtract:
No lemmas
Any of (OR)
All of (AND)
Apply
Unapply
Reset
Translation equiv
Definition
Usage example
Combination
Cross-reference
From lemma
From sense
Incomplete
Homonyms
Homographs
Etymology
Deriv
Pronunciation
Homonyms
Homographs
Etymology
Deriv
Pronunciation
Notes
Markup
Sound
POS
noun
pl noun
verb
POS
Usage label
Deriv
Pronunciation
mokgonyana
mokgomana
mokgaritswana
mokaka
moahlodi
maaka
apola [2]
mohlahli
mohlahlobi
mohlako
mohlana
mohlankedi
mohlare
mohlatlo
mohlolo [2]
mohu
mohumi
moikgogomo
moima
mojako
mojalefa
mokaka
mokgaritswar
mokgomana
mokgonyana


TshwaneLex (Unicode) - [C:\Sesotho sa Leboa - Chinese.tldict]
File Edit View Lemma Dictionary Format Tools Window Help Debug
New (Ins)
Delete
Reverse
fela
fela
1 但是
2 只有
3 (verb) 結束, 結尾
fihla
1 (verb) 到達
2 (verb) 躲
fela [0]
1
但是
2
只有
3 (verb)
結束
結尾
Attributes (F1) Attributes (F2) Search (F3) Format (F4) Filter (F5)
noun
pl noun
verb
verb transitive
verb intransitive
adjective
adverb
abbreviation
pronoun
conjunction
interjection
1/2
1/2
7 (7/8)
8 (7/8
7
3
Bilingual Refe
但是
只有
fela
di
dira
e
eng?
šetše
eupša
fa
fao
fela
fihla
ga
gago
gagwe
gape
ge
go
fela
šetše
[fela]
但是
1 eupša, fela
只有
1 fela
結束
1 fela
結尾
但是 [0]
1
eupša
fela
Lemma:
Incomplete
Lemma sign
但是
Pronunciation
Deriv.
Etymology
Notes
Sense 1:
TE 1
eupša
TE 2
fela
Bilingual Referer
fela
(過去式)
下雨
不知道
並且
事件
事情
人
人們
他
他們
他的
以後
但是
你
你的
來
但是
以後
他的
他們
他



2. User-Friendly Design: Article Preview & Tree View

One of the primary design goals behind the development has been to produce a user-friendly tool: the software should be easy to learn and as intuitive as possible to use. One of the underlying principles for this goal is that lexicographers should not *need* to have a high level of computer literacy in order to perform the day-to-day tasks of compiling a dictionary. The level of abstraction presented to the lexicographer should be that of a *dictionary article*, and not that of a *database*. The generic Input/Output architecture supports this idea by 'hiding', wherever possible, the technical details of *how* the dictionary is stored.

TshwaneLex displays, at all times, a full list of the lemma signs of all articles in the dictionary. Clicking on a lemma sign selects that lemma for viewing or editing, and displays a WYSIWYG preview of the currently selected lemma. This preview updates immediately whenever changes are made. All lemma signs appearing in the preview are hyperlinked, allowing the lexicographer to immediately select a lemma. The different fields of an article (e.g. definitions, translation equivalents and usage examples) are displayed in different colours. These colours are customisable, as well as the font face and formatting information for each field type, through the use of a styles system.

Editing of lemmas is centred around a window called the tree view. This view displays the hierarchical structure of a lemma, broken down into the various word senses, sub-senses, combinations or multi-word units (MWUs), usage examples, cross-references, etc. that constitute the article. By clicking on a node in the tree view, the various fields associated with that node type are instantly displayed in the attributes window, where they may then be modified. Nodes in the tree view may also be displayed in different colours depending on the field type. These have the same colours as used in the article preview.

Another important underlying design philosophy has been that any lexicography tasks that *can* reasonably be automated by computer software, *should* be. One prominent example is the automatic tracking and updating of changes to cross-references; this prevents many of the problems associated with the manual tracking of cross-references. Other examples are automated lemma reversal, and assisted dictionary error and consistency checks.

3. Extendible Input/Output (I/O) Architecture

The I/O architecture of TshwaneLex is designed to support multiple different types of data storage mechanisms. The primary data storage type for networked (multi-user) use is a relational database system, such as MySQL or Microsoft SQL. This is implemented internally using Object Database Connectivity (ODBC) and Structured Query Language (SQL). Dictionaries may also be stored in a binary file on disk, or in XML format.

Some of the export (output) formats supported in TshwaneLex are static HTML (with or without style sheets) and Rich Text Format (RTF) (for producing print output in word processors such as Microsoft Word, OpenOffice and Corel WordPerfect).

The I/O architecture has been genericised, which allows additional interfaces for different types of data sources to be developed in the form of add-ons or plug-ins. This allows for the possibility of custom importers to be created in cases where there may be existing dictionary data developed in another system. Add-on modules may also be created in order to support other output formats.

4. Importing Data, Full Dictionary Searches, Lemma Filter, Error Checks

Wordlists as well as frequency counts calculated by corpus query tools can be imported directly into TshwaneLex. Frequency information can be used to select only a subset of the most frequent lemmas while producing the dictionary output.

A text search function allows the entire dictionary to be searched quickly for a particular piece of text. This includes options such as case-sensitivity, whole-word/partial-word matching, and support for regular expressions.

Using the filter function, the lexicographer can specify criteria to select and work with only a particular subset of lemmas in the dictionary database. This includes conditions for both inclusion and exclusion of lemmas, and these may be combined. The first screenshot e.g. shows how a simple filter may be built to extract all articles that have both definitions and examples, as well as cross-references, yet from which all articles that either include derivations or sound fields are subtracted. In this case there are 778 out of 8539 such articles.

TshwaneLex further assists the lexicographer by performing automatic checks for a variety of possible errors in the dictionary, such as incorrectly nested brackets, duplicate translation equivalents, white-space errors, etc.

5. Bilingual Features: Linked View, Lemma Reversal & Customisation

While working on a bilingual dictionary, whenever a lemma is selected for viewing, TshwaneLex automatically generates and displays clickable shortcuts to related lemmas in the reverse side of the dictionary, i.e. to lemmas that contain as a translation equivalent the lemma sign of the selected lemma. In the optional linked view mode, shown in the second screenshot, this is taken a step further: whenever a lemma is selected in one language, related lemmas in the other language are automatically displayed in the preview window of the other language. This allows the lexicographer to immediately determine if the treatment of an article is well balanced in both sides of the dictionary.

Automatic lemma reversal functions provide the lexicographer with assistance in the process of reversing lemmas by automating as much as possible. A full reversal may be performed on an entire side in one go, or lemmas may be reversed individually. In each case, various reversal options are provided regarding MWUs.

Text labels used to display meta-language such as cross-reference types, parts of speech or usage labels, are configurable. This allows different versions of the same dictionary to be created from the same database; a bilingual dictionary, for instance, could be produced in two different versions, each customised according to the mother tongue of the target users.

6. Cross-references: Automatic Updating of Homonym & Sense Numbers

TshwaneLex automatically keeps track of cross-references; the lexicographer does not need to keep track of cross-references manually. Cross-reference integrity is in fact enforced in TshwaneLex, unlike in e.g. the DPS (McNamara 2003: 9-10), as cross-reference addresses must exist: the lexicographer *cannot* create a cross-reference to a non-existent item.

Whenever a sense number or homonym number changes, TshwaneLex automatically updates all cross-references that refer to that sense or homonym. This prevents the accidental creation of incorrect cross-references when a sense or homonym number is modified and the lexicographer does not also update the relevant cross-references. If in a certain language (see the first screenshot) there is a cross-reference from *monyadi* to the second sense of *mokgonyana*, and senses two and three at *mokgonyana* are swapped around, then the cross-reference from *monyadi* automatically adapts to refer to the third sense of *mokgonyana*. Likewise, if the singular *mokgonyana* is cross-referenced from the plural *bakgonyana*, which is homonymous with another *bakgonyana*, and the two homonyms are changed around, then the cross-reference is again resolved automatically.

As can be seen from the same screenshot, the article preview area also automatically displays all related articles linked by cross-references to or from the currently selected article, allowing all cross-references to quickly be verified as the lexicographer works.

7. Teamwork: Network Lemma Locking & Dictionary Compare/Merge

When working in a team, TshwaneLex prevents multiple lexicographers from attempting to modify the same lemma at the same time through the use of lemma locking. While one lexicographer is working on a lemma, nobody else may work on that lemma.

A compare/merge tool allows two dictionary databases to be compared, with the differences being shown visually, side-by-side. Differences may then be merged into or added to the current dictionary. The dictionary compare/merge tool is particularly useful in situations where lexicographic teams are geographically dispersed, and the option of a high-speed network connection between team members and the main database server is economically or logistically infeasible.

8. Extensions: Sorting, Electronic & Online Dictionary Modules, etc.

In order to be able to handle any possible strategy for the sorting of lemmas, TshwaneLex may be extended to support new sorting methods by means of the development of plug-ins. By default, TshwaneLex implements the ISO 14651 standard, using configurable tables.

A module is available that allows a TshwaneLex dictionary to be published to CD-ROM, and a PHP-based software module is also available that allows a TshwaneLex dictionary to be placed on the Internet. See De Schryver & Joffe (2004) for more details. TshwaneLex furthermore provides the optional capability to include a built-in spellchecker.

For future developments of TshwaneLex, please visit http://tshwanedje.com/tshwanelex/

References

De Schryver, G-M and Joffe, D. 2004. 'On How Electronic Dictionaries are Really Used' (*see elsewhere in the current Proceedings*)

Joffe, D., De Schryver, G-M and Prinsloo, D. J. 2003a. 'Introducing TshwaneLex – A New Computer Program for the Compilation of Dictionaries' in G-M de Schryver (ed.), *TAMA 2003 South Africa: Conference Proceedings*: 97–104. Pretoria: $(SF)^2$ Press.

Joffe, D., De Schryver, G-M and Prinsloo, D. J. 2003b. Computational features of the dictionary application "TshwaneLex". *Southern African Linguistics and Applied Language Studies* 21.4: 239–250.

McNamara, M. 2003. 'Dictionaries for all: XML to Final Product' in *Online Proceedings of XML Europe 2003 Conference & Exposition. Powering the Information Society*. Available from: <http://www.xmleurope.com/2003/>.

Ridings, D. 2003. 'Lexicographic workbench: A case history' in P. van Sterkenburg (ed.), *A Practical Guide to Lexicography*: 204–214. Amsterdam/Philadelphia: John Benjamins.