

The LuwianType Package

Jonathan Walls

July 15, 2024

1 Introduction

This package aims to allow platform-agnostic typing in Luwian hieroglyphs using LaTeX. It is based on the Noto Sans Anatolian Hieroglyphic font by Google, and incorporates functionality from the fontspec, stack-engine, and graphicx packages in order to create a system for the aforementioned purpose.

2 Setup

The system aims to be maximally user-friendly (to the extent that LaTeX ever is), allowing more-or-less free typing using Latin characters as LaTeX commands which are then converted to hieroglyphic text in the PDF. To begin, in the project folders, one must include:

1. The Fonts folder containing the necessary fonts
2. The luwiantype.sty file containing the commands necessary for the Unicode conversions.

In the preamble of the main.tex file (or however you name your main project file), one must also include the following line to import the package:

```
\usepackage{luwiantype}
```

This will incorporate the fontspec package allowing switching to the Unicode font, bring in the LuvianType functionality from the .sty file, and create a

```
\Luvian
```

command which will allow easy switching to the cuneiform typing mode. To switch back, simply use the \Lat command.

3 Usage

3.1 Basic Hieroglyphic Typing

To use the package, in a given section, you can trigger the cuneiform writing mode by using the \Luvian command as listed above. Each character is designated by an uppercase command (to differentiate it from other stock LaTeX commands), and subscripts for sign variants are given in brackets. Typing is made fairly easy by simply turning on caps lock. An example of use is shown below:

```
\Luvian\DEUS\KU\AVIS\PA\PA\SA\Lat
```

This displays the hieroglyphic inline text corresponding to:

(DEUS)ku-AVIS-pa-pa-sa



Luvian of course is written in boustrophedon pattern, and the graphicx package has been utilized to create a \boust{} command which takes arguments and then mirrors the text for R-to-L reading as shown:

```
\Luvian\boust{\DEUS\KU\AVIS\PA\PA\SA}\
```



Luwian, however, was written not in strict linear fashion, but included columns as well. This, naturally, is much more easily reproduced in physical media than in text, but the stackengine package allows us to approximate the look and feel of real Luwian text, should this be required. For this purpose, we have the `\gstack{\argi} {\argii} {\etc}` command which stacks glyphs in a “glyph stack,” with as many glyphs as needed.

```
\Luwian\gstack{{\DEUS} {\KU} {\AVIS}}\gstack{{\PA} {\PA} {\SA}}
\HUGE\gstack{{ } {\REX}}
```



In this example, we can see the two glyph stacks, with each glyph in brackets and separated by a space in the `\gstack` command, shown next to each other, as well as a following stack made up of a single large REX sign and an empty argument. Often, single signs may be written large and take up an entire column, which is best approximated by what is shown above. Here, the font size is changed with `\HUGE`, and a space argument is fed into the glyph stack to help with spacing. This system is very flexible, and able to relatively neatly approximate the look of most Luwian text.

To change between $S\acute{A}$ and $S\grave{A}$, use the following syntax:


```
\SA[2]
\SA[3]
%the base SA character can be had with
\SA
```

Thus 𐎗 , 𐎗 , 𐎗 , in order.

Commands which correspond to numbers, i.e., 1, 2, etc., are in the LaTeX code with

```
\ONE
\tWO
```

as LaTeX does not allow numeric characters in command names. Signs with varying vowels like *wa/i* have both vowels in the command name, un-separated, as in `\WAI`. Those signs which are compound, such as `LITUUS + na`

will be designated without any break between the signs as `\LITUUSNA`, except on the rare occasion where a distinction needs to be made between two signs joined by `.` or `+`, in which case the latter is designated `PLUS` in the command name. The signs which do not have a standard reading such as  will be designated by an L (Laroche) and the appropriate Laroche number in roman numerals, as in `\LIII`. The special signs for word divisions, logograms, and personal identifiers are `\VIRIIA`, `\LOGS` ("logogram start"), `\LOGE` ("logogram end"), and `\PERS`.

3.2 Formatting Transcriptions

This package also contains a number of commands which are useful for formatting transcriptions of Luwian, which follows conventions that are often tedious to work through. Some of these will create special characters often used in such transcriptions, as in the following table.

Command	Effect
<code>\badeq</code> ("bad equal")	≠
<code>\hl</code> ("half left")	┌
<code>\hr</code> ("half right")	┐

Some other commands are included which serve to facilitate formatting, as shown in the following table.

Command	Effect
<code>\tsup{}</code>	^M
<code>\tsub{}</code>	₂

`\tsup` and `\tsub` create super- and subscripts, respectively.

3.3 Working with expex

If one is using the `expex` package for interlinear translation or example glossing, the easiest thing to do is to define a new `glw` level as shown in this example:

```

\defineglwlevels{luw}
\beginl[glwordalign=center,glhangstyle=none, aboveglftskip=0pt,
everygla=\normalfont, everyglluw=\Luwian]

\gla REX \textsuperscript{I}t'a-i-t'a-sa EGO-wa/i-mi
(VIR\textsubscript{2})HEROS (VIR\textsubscript{2})pa-l'a/'i-s`a-
ti-[n'i]-za-sa
REX a-wa/i x mi-i-*a//

\glluw {\REX} {\PERS\TA[2]\I\TA[2]\SA} {\EGO\WAI\MI} {\VIR[2]\HEROS}
{\VIR[2]\PA\LAI[2]\SA[3]\TI\NI[2]\ZA\SA} {\REX} {\A\WAI} x {\MI\I\A}//

\glb hantawattis Taitas amu\badeq wa\badeq mi hastalis Falistinizzas
hantawattis a\badeq wa x ammi//
\glc king.nom Taitas I.quot.myself Hero Falistina.nom king.nom and.quot x
my.dat//
\glft I am King Taitas, Hero, Falistinian king, and I\newline//
\endgl

```

Note the section at the beginning which defines a “luw” (Luwian) glw level, which needs to be done only once at the beginning of the document, and note also the everyglluw=\Luwian specification in the parameters which assigns the entire line to be in Luwian hieroglyphs. You may of course omit one of the transcription levels, in which case you can simply add the specification to the a, b, or c level instead. This code will produce the following output:

REX	^I tá-i-tá-sa	EGO-wa/i-mi	(VIR ₂)HEROS
△	∩ ∩ ∩ ∩ ∩	⊙ ∘ ∥∥∥	□□ ∩∩
hantawattis	Taitas	amu=wa=mi	hastalis
king.nom	Taitas	I.quot.myself	Hero
(VIR ₂)pa-lá/í-sà-ti-[ní]-za-sa		REX	a-wa/i x mi-i-*a
□□ ∩ ∩ ∩ ∩ ∩ ∩ ∩	△	∪ ∘	× ∥∥ ∩ ∪
Falistinizzas	hantawattis	a=wa	x ammi
Falistina.nom	king.nom	and.quot	x my.dat

I am King Taitas, Hero, Falistinian king, and I

Happy Luwian!