Typing Linguistics with `covington.sty`

Michael A. Covington
Artificial Intelligence Center
The University of Georgia
Athens, Georgia 30602 U.S.A.
mcovingt@ai.uga.edu
`http://www.ai.uga.edu/~mc`

Version 1.1, July 5, 2016[*]

Contents

New in This Version

- The package now uses PSNFSS font commands if available (fallback for LaTeX 2.09 is still provided)

[*]Current maintainer: Jürgen Spitzmüller. Please report issues via `https://github.com/jspitz/covington/issues`

- License has been changed to LPPL (in agreement with M. Covington)
- New maintainer: J. Spitzmüller
- Introduce version numbers. Arbitrarily, we start with 1.1

New in Preceding Versions

2001 March 27

- It is no longer necessary to type `\it` to get proper italic type in feature structures.
- Instructions have been rewritten with LaTeX 2_ε users in mind.

Older versions

- Multiple accents on a single letter (e.g., $\acute{\tilde{a}}$) are supported.
- This package is now called `covington` (with the o) and is compatible with LaTeX 2_ε and NFSS as well as LaTeX 2.09.
- The vertical placement of labeled feature structures has been changed so that the category labels line up regardless of the size of the structures.

Introduction

This file, `covington.tex`, is the documentation for Version 1.1 of `covington.sty` (July 5, 2016), which is a LaTeX style option for typing many of the special notations common in linguistics.

In emTeX under MS-DOS, `covington.sty` is called `covingto.sty`. The missing *n* has no effect.

To use `covington.sty`, you should have a copy of it in either your current directory or the directory where LaTeX styles are kept on your system.

Then, under LaTeX 2_ε, include the command `\usepackage{covington}` after your `\documentclass` command.

In LaTeX 2.09, include `covington` among the optional parameters of `\documentstyle`, like this:

```
\documentstyle[12pt,covington]{article}
```

Note the spelling `covington` (9 letters).

In what follows I presume that you know how to use LaTeX and have access to the LaTeX manual. Note that `covington.sty` does not provide any special fonts or character sets. However, it can be used in combination with other style sheets that do.

If you are using `covington.sty` and `uga.sty` (UGa thesis style) together, you should mention `uga` before `covington`.

1 Accents

LaTeX provides a generous range of accents that can be placed on any letter, such as:

x̀ x́ x̂ ẍ x̃ x̄ x̋ x͡x x̧ x̣ x̱

which are typed, respectively, as:

`` \`{x} \'{x} \^{x} \"{x} \~{x} \={x} \H{x} \t{xx} \c{x} \d{x} \b{x} ``

LaTeXalso provides the foreign characters

ı ȷ æ Æ œ Œ å Å ø Ø ł Ł ß ¿ ¡

which are typed as:

``\i \j \ae \AE \oe \OE \aa \AA \o \O \l \L \ss ?` !` ``

But by itself, LaTeX doesn't give you a convenient way to put two accents on the same letter. To fill this gap, `covington.sty` provides the following macros:

`\twoacc[...|...]` to combine any 2 accents, e.g., `\twoacc[\~|\={a}]` = ā̃

`\acm{...}`	for acute over macron, e.g., `\acm{a}` = ā́
`\grm{...}`	for grave over macron, e.g., `\grm{a}` = ā̀
`\cim{...}`	for circumflex over macron, e.g., `\cim{a}` = ā̂

The first of these is the general case and the latter three are special cases that occur often in transcribing Greek. Now you can type *Koinḗ* with both accents in place.

Note the peculiar syntax of `\twoacc` — its arguments are in square brackets, not curly brackets, and are separated by `|`. The first argument is the upper accent (only) and the second argument is the letter with the lower accent indicated.

Note also that not all accents work in the `tabbing` environment. Use `tabular` or see the LaTeX manual for workarounds.

2 Example numbers

Linguistics papers often include numbered examples. The macro `\exampleno` generates a new example number and can be used anywhere you want the number to appear. For example, to display a sentence with a number at the extreme right, do this:

```
\begin{flushleft}
This is a sentence. \hfill (\exampleno)
\end{flushleft}
```

Here's what you get:

This is a sentence. (1)

The example counter is actually the same as LaTeX's equation counter, so that if you use equations and numbered examples in the same paper, you get a single continuous series of numbers. If you want to access the number without changing it, use `\theequation`.

Also, you can use `\label` and `\ref` with example numbers in exactly the same way as with equation numbers. See the LaTeX manual for details. This applies to the `example` and `examples` environments, described next, as well as to `\exampleno` itself.

3 The `example` environment

The `example` environment displays a single example with a generated example number to the left of it. If you type

```
\begin{example}
This is a sentence.
\end{example}
```

you get:

(2) This is a sentence.

The `example` environment is a lot like `flushleft`. The example can be of any length; it can consist of many lines (separated by \\), or even whole paragraphs.

One way to number sub–examples is to use `itemize` or `enumerate` within an example, like this:

```
\begin{example}
\begin{itemize}
\item[(a)] This is the first sentence.
\item[(b)] This is the second sentence.
\end{itemize}
\end{example}
```

This prints as:

(3) (a) This is the first sentence.
    (b) This is the second sentence.

However, the `examples` environment, described next, is usually more convenient.

4 The `examples` environment

To display a series of examples together, each with its own example number, use `examples` instead of `example`. The only difference is that there can be more than one example, and each of them has to be introduced by `\item`, like this:

```
\begin{examples}
\item This is the first sentence.
\item This is the second sentence.
\end{examples}
```

This prints as:

(4) This is the first sentence.

(5) This is the second sentence.

5 Glossing sentences word–by–word

To gloss a sentence is to annotate it word–by–word. Most commonly, a sentence in a foreign language is followed by a word–for–word translation (with the words lined up vertically) and then a smooth translation (not lined up), like this:[1]

Dit	*is*	*een*	*Nederlands*	*voorbeeld.*
This	is	a	Dutch	example.

‘This is an example in Dutch.’

That particular example would be typed as:

```
\gll Dit is een Nederlands voorbeeld.
     This is a Dutch example.
\glt ‘This is an example in Dutch.’
\glend
```

Notice that the words do not have to be typed lining up; instead, TEX counts them. If the words in the two languages do not correspond one–to–one, you can use curly brackets to show the intended grouping. For example, to print

Dit	*is*	*een*	*voorbeeldje*	*in*	*het*	*Nederlands.*
This	is	a	little example	in		Dutch.

‘This is a little example in Dutch.’

you would type:

```
\gll Dit is een voorbeeldje     in het Nederlands.
     This is a {little example} in {}  Dutch.
\glt ‘This is a little example in Dutch.’
\glend
```

All together, `covington.sty` gives you five macros for dealing with glosses:

- `\gll` introduces two lines of words vertically aligned, and activates an environment very similar to `flushleft`.

[1]The macros for handling glosses are adapted with permission from `gloss.tex`, by Marcel R. van der Goot.

- `\glll` is like `gll` except that it introduces *three* lines of lined–up words (useful for cited forms, morphology, and translation).
- `\glt` ends the set of lined–up lines and introduces a line (or more) of translation.
- `\gln` is like `\glt` but does not start a new line (useful when no translation follows but you want to put a number on the right).
- `\glend` ends the special `flushleft`–like environment.

Here are several examples. First, a sentence with three lines aligned, instead of just two:

Hoc	*est*	*aliud*	*exemplum.*
n.sg.nom	3.sg	n.sg.nom	n.sg.nom
This	is	another	example.

'This is another example.'

This is typed as:

```
\glll  Hoc est aliud exemplum.
       n.sg.nom 3.sg n.sg.nom n.sg.nom
       This is another example.
\glt   'This is another example.'
\glend
```

Next, an example with a gloss but no translation, with an example number at the right:

Hoc	*habet*	*numerum.*	(6)
This	has	number	

That one was typed as:

```
\gll  Hoc habet numerum.
      This has number
\gln  \hfill (\exampleno)
\glend
```

Finally we'll put a glossed sentence inside the `example` environment, which is a very common way of using it:

(7)	*Hoc*	*habet*	*numerum*	*praepositum.*
	This	has	number	preposed

'This one has a number in front of it.'

This last example was, of course, typed as:

```
\begin{example}
\gll  Hoc habet numerum praepositum.
      This has number preposed
\glt  ‘This one has a number in front of it.’
\glend
\end{example}
```

Notice that every glossed sentence begins with either `\gll` or `\glll`, then contains either `\glt` or `\gln`, and ends with `\glend`. Layout is critical in the part preceding `\glt` or `\gln`, and fairly free afterward.

6 Phrase structure rules

To print the phrase structure rule $S \rightarrow NP\ \ VP$ you can type `\psr{S}{NP~VP}`, and likewise for other phrase structure rules.

7 Feature structures

To print a feature structure such as:

$\left[\, \mathit{case} : \mathit{nom} \,\right]$

you can type:

```
\fs{case:nom \\ person:P}
```

The feature structure can appear anywhere — in continuous text, in a displayed environment such as `flushleft`, or inside a phrase–structure rule, or even inside another feature structure.

To put a category label at the top of the feature structure, like this,

$$\overset{N}{\begin{bmatrix} \mathit{case} : \mathit{nom} \\ \mathit{person} : P \end{bmatrix}}$$

here's what you type:

```
\lfs{N}{case:nom \\ person:P}
```

And here is an example of a PS–rule made of labeled feature structures:

$$\overset{S}{\begin{bmatrix} \mathit{tense} : T \end{bmatrix}} \rightarrow \overset{NP}{\begin{bmatrix} \mathit{case} : \mathit{nom} \\ \mathit{number} : N \end{bmatrix}} \quad \overset{VP}{\begin{bmatrix} \mathit{tense} : T \\ \mathit{number} : N \end{bmatrix}}$$

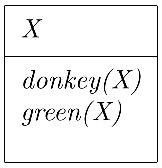
which was of course typed as:

```
\psr{\lfs{S}{tense:T}}
    {\lfs{NP}{case:nom \\  number:N}
     \lfs{VP}{tense:T \\ number:N} }
```

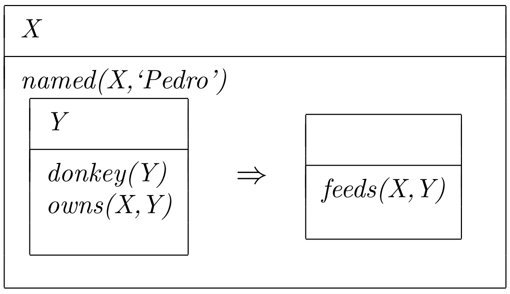
8 Discourse representation structures

Several macros in `covington.sty` facilitate display of discourse representation structures (DRSes) in the box notation originally used by Hans Kamp. The simplest is `\drs`, which takes two arguments: a list of discourse variables joined by `~`, and a list of DRS conditions separated by `\\`. Nesting is permitted. Note that the `\drs` macro itself does not give you a displayed environment; you must use `flushleft` or the like to display the DRS. Here are some examples:

```
\drs{X}{donkey(X)\\green(X)}
```



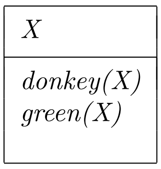
```
\drs{X}
{named(X,‘Pedro’) \\
\drs{Y}{donkey(Y)\\owns(X,Y)}~~
       {\large $\Rightarrow$}~
        \drs{~}{feeds(X,Y)}
}
```



To display a sentence above the DRS, use `\sdrs`, like this:

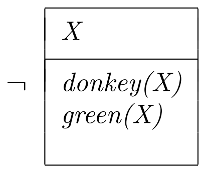
```
\sdrs{A donkey is green.}{X}{donkey(X)\\green(X)}
```

A donkey is green.



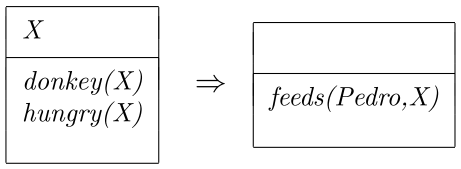
Some DRS connectives are also provided (normally for forming DRSes that are to be nested within other DRSes). The macro `\negdrs` forms a DRS preceded by a negation symbol:

```
\negdrs{X}{donkey(X)\\green(X)}
```



Finally, `\ifdrs` forms a pair of DRSes joined by a big arrow, like this:

```
\ifdrs{X}{donkey(X)\\hungry(X)}
      {~}{feeds(Pedro,X)}
```



If you have an "if"–structure appearing among ordinary predicates inside a DRS, you may prefer to use `\alifdrs`, which is just like `\ifdrs` but shifted slightly to the left for better alignment.

9 Exercises

The `exercise` environment generates an exercise numbered according to chapter, section, and subsection (suitable for use in a large book; in this example, the subsection number is going to come out as 0).

Exercise 9.0.1 (Project) *Prove that the above assertion is true.*

This was typed as

```
\begin{exercise}[Project]
Prove that the above assertion is true.
\end{exercise}
```

and the argument `[Project]` is optional (actually, any word could go there).

10 Reference Lists

To type an LSA–style hanging–indented reference list, use the `reflist` environment. (*Note:* `reflist` is not presently integrated with BibTEX in any way.) For example,

```
\begin{reflist}
Barton, G. Edward; Berwick, Robert C.; and Ristad, Eric Sven.  1987.
Computational complexity and natural language.  Cambridge,
Massachusetts: MIT Press.

Chomsky, Noam.  1965.  Aspects of the theory of syntax.  Cambridge,
Massachusetts: MIT Press.

Covington, Michael.  1993.  Natural language processing for Prolog
programmers.  Englewood Cliffs, New Jersey: Prentice--Hall.
\end{reflist}
```

prints as:

Barton, G. Edward; Berwick, Robert C.; and Ristad, Eric Sven. 1987. Computational complexity and natural language. Cambridge, Massachusetts: MIT Press.

Chomsky, Noam. 1965. Aspects of the theory of syntax. Cambridge, Massachusetts: MIT Press.

Covington, Michael A. 1993. Natural–language processing for Prolog programmers. Englewood Cliffs, New Jersey: Prentice–Hall.

Notice that within the reference list, "French spacing" is in effect — that is, spaces after periods are no wider than normal spaces. Thus you do not have to do anything special to avoid excessive space after people's initials.

11 Displayed sentences

The macro `\sentence` displays an italicized sentence (it is a combination of `flushleft` and `\itshape`). If you type

```
\sentence{This is a sentence.}
```

you get:

This is a sentence.

12 Big curly brackets (disjunctions)

Last of all, the 2–argument macro `\either` expresses alternatives within a sentence or PS–rule:

`the \either{big}{large} dog` = the $\left\{\begin{array}{c}\text{big}\\ \text{large}\end{array}\right\}$ dog

`\psr{A}{B~\either{C}{D}~E}` $= A \rightarrow B \left\{\begin{array}{c}C\\ D\end{array}\right\} E$

That's all there is. Suggestions for improving `covington.sty` are welcome, and bug reports are actively solicited. Please note, however, that this is free software, and the author makes no commitment to do any further work on it.