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 1.2dev, July 7, 2016*

Abstract

This package, initially a collection of Michael Covington's private macros, provides numerous minor LaTeX enhancements for linguistics, including multiple accents on the same letter, interlinear glosses (word-by-word translations), Discourse Represenation Structures, and example numbering. The package works both with LaTeX 2.09 and LaTeX 2_ε.

Contents

1 Introduction 2
2 Stacked accents 2
3 Example numbers 3
4 The **example** environment 4
5 The **examples** environment 5
6 Glossing sentences word-by-word 5
7 Phrase structure rules 7
8 Feature structures 7
9 Discourse Representation Structures 8
10 Exercises 9
11 Reference Lists 10
12 Displayed sentences 11
13 Big curly brackets (disjunctions) 11
14 Release history 11

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

1 Introduction

This file, `covington.tex`, is the documentation for version 1.2dev of `covington.sty` (July 7, 2016), which is a LaTeX package providing macros for typing some special notations common in linguistics.[1]

To use `covington.sty` with LaTeX 2ε, simply add the command `\usepackage{covington}` to your document preamble.

> In LaTeX 2.09, include `covington` among the optional parameters of **`\documentstyle`**, as in: **`\documentstyle`**`[12pt,covington]{article}`

The package has the following options:

- **`force`**: Force the redefinition of environments that are already defined. This applies to the **`example`**, **`examples`** and **`exercise`** environments, which are by default not touched if they are already defined before `covington` is loaded. See sec. 4, 5 and 10 for details.

In what follows we presume that you know how to use LaTeX and have access to LaTeX manuals. 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 load `uga` before `covington`.

2 Stacked accents

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

x̀ x́ x̂ ẍ x̃ x̄ x̋ x̌ x̧ x̨ x̣ x̲

which are typed, respectively, as:

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

LaTeX also provides support for many non-ASCII characters, such as[2]:

ı ȷ æ Æ œ Œ å Å ø Ø ł Ł ß ¿ ¡

via the macros:

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

But out of the box, 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:

[1] The package has a long history. It started off as a collection of private macros back in the LaTeX 2.09 days and was initially released as `covingtn.sty` (following the old 8.3 FAT file name limit). In emTeX under MS-DOS, the file was distributed as `covingto.sty`. Eventually, it has been renamed to `covington.sty` and adapted to LaTeX 2ε. Its LaTeX 2.09 traces are however still visible, and the style should actually still work with LaTeX 2.09 (if not, drop us a note).

[2] Please refer to [1] for a comprehensive list of special characters and symbols.

\twoacc[...|...] to combine any two accents, e. g., **\twoacc**[\~|\={a}] = ã

\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 are often used in Greek transcription. Now you can type *Koinḗ* with both accents in place.

The distance between the two accents can be adjusted via the length `\twoaccsep`, which is preset to `-0.8ex`. For instance, if you use `\setlength{\twoaccsep}{-1.05ex}`, the above examples will come out as

\twoacc[...|...] to combine any two accents, e. g., **\twoacc**[\~|\={a}] = ã

\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

with a slightly better matching distance for the font used here.

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 refer to [1] for alternative solutions.

3 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. Refer to your LaTeX manual for details. This applies to the **example** and **examples** environments, described next, as well as to **\exampleno** itself.

4 The `example` environment

The **`example`** environment (alias **`covexample`**) 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}
```

or

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

you get:

(2) This is a sentence.

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

If you need more space between the example number and the text, you can increase it by means of the length **`\examplenumbersep`** (which is preset to zero). Doing `\setlength\examplenumbersep{1em}`, for instance, will increase the space by 1 em.

Note that, as of version 1.1, `covington` checks if there is already an **`example`** environment defined (e. g., by the class). If so, `covington` does not define its own one. However, there is always the alias environment **`covexample`** which can be used in order to produce `covington's` example. If you use the package option **`force`**, `covington` will override existing **`example`** environments. In any case, the package will issue a warning if **`example`** is already defined (this is the case, for instance, if you use `covington` with the `beamer` class).

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.

5 The `examples` environment

To display a series of examples together, each with its own example number, use **`examples`** (or **`covexamples`**) instead of **`example`** or **`covexample`**. 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}
```

or, respectively:

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

This prints as:

(4) This is the first sentence.

(5) This is the second sentence.

As for **`example`**, `covington` checks if there is already an **`examples`** environment defined, and if this is the case, `covington` does not define its own one. The alias environment **`covexamples`** is always available as a fallback. If you use the package option **`force`**, `covington` will override existing **`examples`** environments. The package will issue a warning if **`examples`** is already defined (this is the case, for instance, if you use `covington` with the `beamer` class), telling you how it has dealt with the situation.

6 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:[3]

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
```

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

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 group words. 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` provides five macros for dealing with glosses:

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

7 Phrase structure rules

To print phrase structure rules such as $S \rightarrow NP\ VP$ you can use `covington's` macro `\psr{<constituent>}{<sub-constituents>}` (for the given example, `\psr{S}{NP~VP}`).

8 Feature structures

To print a feature structure such as

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

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,

$$\begin{array}{c} N \\ \begin{bmatrix} case : nom \\ person : P \end{bmatrix} \end{array}$$

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:

$$\underset{\left[\, tense : T \,\right]}{S} \quad \rightarrow \quad \underset{\left[\begin{array}{l} case : nom \\ number : N \end{array}\right]}{NP} \quad \underset{\left[\begin{array}{l} tense : T \\ number : N \end{array}\right]}{VP}$$

which was obviously coded as:

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

9 Discourse Representation Structures

Several macros in `covington.sty` facilitate display of discourse Representation Structures (DRSes) in the box notation introduced by Hans Kamp. The simplest one 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:

```
\begin{flushleft}
  \drs{X}{donkey(X)\\green(X)}
\end{flushleft}
```

X
donkey(X) *green(X)*

```
\begin{flushleft}
  \drs{X}
  {named(X,'Pedro') \\
    \drs{Y}{donkey(Y)\\owns(X,Y)}~~
    {\large $\Rightarrow$}~
    \drs{~}{feeds(X,Y)}
  }
\end{flushleft}
```

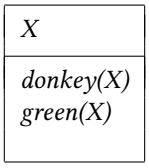
X		
named(X,'Pedro')		
Y	$\Rightarrow$	
donkey(Y) *owns(X,Y)*		*feeds(X,Y)*

To display a sentence above the DRS, use **`\sdrs`**, as in:

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

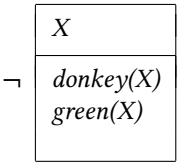
which prints as:

A donkey is green.



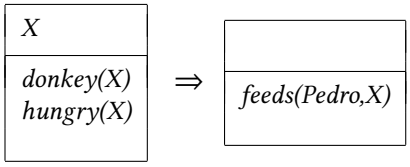
Some DRS connectives are also provided (normally for forming DRSes that are to be nested within other DRSes). The macro **\negdrs** forms a DRS preceded by a negation symbol:

```
\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.

10 Exercises

The **exercise** environment (alias **covexercise**) 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). Here is an example:

Exercise 10.0.1 (Project) *Prove that the above assertion is true.*

This was coded as

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

The argument (`[Project]` in the example) is optional.

Note that, as of version 1.1, `covington` checks if there is already an **`exercise`** environment defined (e. g., by the class). If so, `covington` does not define its own one. However, there is always the alias environment **`covexercise`** which can be used in order to produce `covington`'s exercise. If you use the package option **`force`**, `covington` will override existing **`exercise`** environments. In any case, the package will issue a warning if **`exercise`** is already defined.

11 Reference Lists

To type a simple LSA-style hanging-indented reference list, you can use the **`reflist`** environment. (*Note:* **`reflist`** is not integrated with BibTEX in any way.[4]) 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.

[4] For BibTEX, there are several options: the LSA style, as used in the journal *Language*, can be obtained by means of the style files `lsalike.bst` (`http://www.icsi.berkeley.edu/ftp/pub/speech/jurafsky/lsalike.bst`) or `language.bst` (`http://ron.artstein.org/resources/language.bst`); the latter uses `natbib`. The so-called *Unified Style Sheet for Linguistics*, as proposed by the CELXJ (*Committee of Editors of Linguistics Journals*), which slightly differs from the LSA style, is followed by the style file `unified.bst` (available at `http://celxj.org/downloads/unified.bst`). A `biblatex` style file for the unified style is available at `https://github.com/semprag/biblatex-sp-unified`.

12 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.

13 Big curly brackets (disjunctions)

Last of all, the two-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 for now. Suggestions for improving `covington.sty` are welcome, and bug reports are actively solicited (via `https://github.com/jspitz/covington`). Please note, however, that this is free software, and the authors make no commitment to do any further work on it.

14 Release history

1.2 (forthcoming)

- New length **\examplenumbersep** to adjust (increase) the horizontal space between example number and example text.
- Add some more info about bibliography generation.

1.1a (2016 July 7)

- Fix encoding problem in documentation and some typos. No change in functionality.

1.1 (2016 July 6)

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

- Work around clash with classes/packages that define their own **example** and **examples** environments (most notably the beamer class) as well as **execise** environments. The covington package no longer blindly attempts to define these environments. By default, it does not define them if they are already defined (covington's own environments, however, are still available via aliases). By means of a new package option, a redefinition can also be forced. See sec. 4 and 5 for details.
- New length **\twoaccsep** allows for the adjustment of the distance between stacked accents (see sec. 2).
- Update manual.
- New maintainer: J. Spitzmüller.
- License has been changed to LPPL (in agreement with M. Covington)
- Introduce version numbers. Arbitrarily, we start with 1.1.

2014 May 16

- Patches by Robin Fairbairns:
  - Setting of \textfloatsep uses \setlength rather than \renewcommand
  - Style file converted to un*x line endings

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., $\acute{\acute{a}}$) 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.

References

[1] Pakin, Scott. The Comprehensive LaTeX Symbol List. 30 November 2015. http://www.ctan.org/tex-archive/info/symbols/comprehensive.