

LaTex for Linguists+

Sandra Auderset
MPI for Evolutionary Anthropology &
University of California Santa Barbara

MPI-SHH
SUMMER SCHOOL
2021

Doorway
to Human History

Roadmap

Topics	Modality
1. Intro to the what and why of LaTeX (ca. 5 min)	slide presentation (SA)
2. General components of a .tex document (ca. 40 min)	slides (SA) + practice (you)
3. Linguistics hacks (ca. 30 min)	slides (SA) + practice (you)
4. Time for questions/discussion (ca. 15 min)	open format (you)

What is LaTeX and why use it?

What is LaTeX?

- document preparation system for high-quality typesetting developed in 1985
- main idea is separation of content and formatting
  - editors like Word, LibreOffice, OpenOffice are what-you-see-is-what-you-get (WYSIWYG)
  - focusing on content/writing
  - well designed documents
- uses plain text and markup tagging for defining the document structure as a whole, as well as parts of the document

Why use LaTeX?

- lets you focus on content (rather than fiddling around with formatting constantly)
- generates many useful components automatically, e.g. table of contents and bibliography
- stable formatting (numbering, bold face, different scripts etc.)
- inherently cross-platform (natively works on Mac, Windows, and Linux)
- no compatibility issues across operating systems and versions (a tex-document will look exactly the same no matter what operating system or computer it was compiled on)
- open source, non-proprietary
- still actively developed

Why use LaTeX as a linguist?

- glossing is a breeze (aligns glosses and generates a glossary automatically)
- easy to write in multiple scripts (also non-latin ones) and languages
- supports any fonts installed on your computer (e.g. for using stacked diacritics)
- trees can be drawn within LaTeX (instead of using another program)

Some drawbacks of using LaTeX

- it takes more time to learn than Word or Markdown (especially in the beginning)
- you have to familiarize yourself with reading error messages and trouble-shooting
- tables in LaTeX are a bit of a pain in the a** (but there are work-arounds)
- it generates a bunch of helper files (so each tex-file should have its own folder)
- it cannot easily be converted to Word (although it is possible)
- some journals don't accept LaTeX submissions (but they can be avoided)
- some collaborators might not want to use it (but they might be convinced to try Overleaf)

A note before we get started

- there are almost always multiple ways of achieving the same thing in LaTeX (as in any other programming language) – I'll show you the way I do things, because it works well for me, but that does not mean that is the 'correct' or only way

➢ Task: open the testdoc.tex in the LaTeX editor you installed

General components of a tex-document

tex-things useful to everyone

SUMMER SCHOOL 2021

Doorway to Human History

The structure of a LaTeX document

Preamble:

- specify type of document (article, presentation, book, etc.)
- set up the document (margins, fonts, etc.)
- load packages (for customizing the document)

Content/Main part:

- content (your text) with sections, subsections, tables, images etc.
- bibliography info
- optionally: glossaries, appendices etc.

➢ Practice: identify the preamble and the content section in the testdoc.tex and post the line numbers in the Zoom chat

➢ to show line numbers in TeXstudio: Preferences/Options > Editor > Show Line Number [select: All Line Numbers]

Structuring your document

- Titles: details specified in the preamble, includes author and date; then called in the main section
- Sections: in the main section with \section{Section Title}; add asterisk for unnumbered section
- Table of contents: generated automatically with \tableofcontents
- Paragraphs and indentation: white space is ignored, two empty lines make a new paragraph with indentation

```
% set up the title page
   \title{Template}
   \author{Me Me}
   \date{\today}


\begin{document}


% generate title
\maketitle

% generate a table of contents
\tableofcontents


% this is a section without numbering
\section*{Introduction}
% this is a numbered section
\section{Theoretical Background}
```

Structuring your document - Practice

- give your document a new title
- enter your name as the author
- create a numbered and an un-numbered section (section titles are up to you)
- add two new paragraphs with indentation (use some already written text you have or some dummy text from here: https://www.lipsum.com/)
- compile!

Writing text and formatting it

- text formatting bold, italics, etc.: not with buttons, but with commands (or shortcuts!) like \textit{italic text}, \textbf{bold text}, \textsc{small caps}, etc.
- quotes and dashes: beginning and end quotes are different, namely ` ‘ (single quotes) and `` ‘’ (double quotes)
- special characters (and how to escape them): some characters have special meanings in LaTeX and need to be escaped with a \ to be typset (complete list here: https://newbedev.com/escape-character-in-latex)
- footnotes: just use \footnote{and place the relevant text inside}

```
\textit{italics}
\textbf{bold}
\textbf{\textit{bold and italic}}

Single quotes: `hello?' and
double quotes ``hello?!''

The \% is usually for comments,
so it has to be escaped.\footnote{And here a footnote.}
```


Writing text and formatting it - Practice

- write some text in bold face, italics, small caps, and bold face and italics
- add a footnote
- put a word or sentence in single quotes and one in double quotes
- type a special character and escape it
- compile!

Math mode

- since LaTeX was originally developed by engineers, it has a Math Mode to write pretty formulas
- math mode is delimited by $...$
- let's look at the typesetting of Bayes' Theorem

```
Here is Bayes' Theorem: $P(A|B) \dfrac{P(B|A) \times P(A)}{P(B)}$.
```

Here is Bayes' Theorem: $P(A|B)\frac{P(B|A) \times P(A)}{P(B)}$.

- in linguistics mostly used in formal semantics

Labels and references

- everything and anything can be cross-referenced (sections, tables, images, examples, you name it!)
- this is done with labels and references: first create the \label{label} and then you can refer to it with \ref{label}
- labels are placed directly underneath the element they are labeling

```
\section*{Introduction}
\label{sec:intro} % this is a label for the section
```

```
\begin{table}
  \centering
  \begin{tabular}{|l|l|r|c|} \hline
    a  & b  & d  \\ \hline
    df  & df  & eg  \\ \hline
    bla & bla & bla \\ \hline
  \end{tabular}
  \caption{Lots of blabla}
  \label{tab:firsttable}
\end{table}
```

Labels and references - Practice

- make a label for a section and refer to it further down in the document

- make a reference to a table or example that already has a label

- compile (in case the references show up as ??, just compile again)

Floats and captioning - Basics

- floats are containers for things that cannot be broken over a page, like tables and figures
- floats are always numbered (automatically) and captioned
- LaTeX automatically places floats depending on how much space is left on the page
  - this means they do not (necessarily) show up in the pdf where you placed them in the tex doc
  - you can force them to appear in a certain spot, but in general you should not tinker with this!
  - there are specifiers to give LaTeX hints where the float should be placed

Specifier	Permission
`h`	Place the float *here*, i.e., at the same point it occurs in the source text.
`t`	Position at the *top* of the page.
`b`	Position at the *bottom* of the page.
`p`	Put on a special *page* for floats only.
`!`	Override internal parameters LaTeX uses for determining `good' float positions.

Figures/Images

- reminder: figures are floats!
- package for including images: graphicx
- environment: figure
- images are placed with:
  \includegraphics[options]{filename.extension}
- the image has to be in the same folder as the tex doc (or you need to specify the filepath)
- the options control the size of the image: you can specify width, height, or scale
- always add a caption

  and a label

```
\begin{figure}
   \centering
   \includegraphics[width=0.5\textwidth]{testpic.jpg}
   \caption{Not the type of latex we are working with}
   \label{fig:testpic}
\end{figure}
```


Figures/Images - Practice

- in the preamble, add the graphicx package
- add a new figure to your document (just pick any image/graphic you have on your laptop)
- add a caption and a label
- compile!
- tinker with the size options to find the best fit – compile again

Tables

- reminder: tables are floats!
- tables are a bit nasty in LaTeX, they get chaotic very fast
- small ones can be done directly in the tex doc, but for bigger ones I recommend another workflow
  1. make your table in a spreadsheet software (I recommend LibreOffice's Calc over Excel, because the latter often leads to encoding issues) without any formatting
  2. copy/paste it into https://www.tablesgenerator.com/
  3. adjust the settings there, copy to clipboard
  4. paste the generated table in the tex document
  5. adjust further in the tex doc (if needed)

Tables - Practice

- get the csv-file from the Zoom chat
- open it with a text editor (or spreadsheet software)
- copy & paste it into https://www.tablesgenerator.com/
- adjust settings if you like
- ’copy to clipboard’ and insert the table into your tex document
- compile!

Citations and bibliography

- package: natbib

```
% and advanced bibliography stuff
    \usepackage{natbib}
        \renewcommand{\bibfont}{\small}
        \renewcommand{\bibsep}{1pt} % space between entries
        \setcitestyle{comma,aysep={},notesep={:}} % comma betw
        post-citation note
        \newcommand\cites[1]{\citeauthor{#1}'s\ (\citeyear{#1})} %
```

- to include the biblography (usually at the end of a document):
  \bibliographystyle{style}

  \bibliography{bibfile}

```
\bibliographystyle{unified.linguistic}
\bibliography{testbib}
```

- to cite something: \cite{citekey} – or with parentheses \citep{citekey}, all text \citealt{citekey}
- bibliography will be autmatically generated from the citations when you compile (sometimes it needs 2 rounds to do this)

Citations and bibliography - Practice

- check the citations in the testdoc.tex and the pdf to understand what the different \cite commands do
- open the testbib.bib in a text editor/bibliography editor
- add a citation (with or without parentheses) in the testdoc of the entry that hasn't been cited yet
- compile! (if a question marks show up, just compile again)

Linguistics hacks

tex-things specifically useful to linguists

SUMMER SCHOOL 2021

Doorway to Human History

Fonts, diacritics, and writing systems

- good news: LaTeX is pretty awesome at dealing with diacritics and different writing systems
- packages: fontspec, xunicode
- fonts: depends on your needs
  - Linux Libertine is generally good (latin, greek, cyrillic)
  - LaTeX can use any font installed on your computer
  - fonts can be added in different ways, many common ones now have their own packages

```
% this loads the unicode and font-specific packages
\usepackage{fontspec,xltxtra,xunicode}
\usepackage{libertine}
\newfontfamily\hindifont{Devanagari MT}[Script=Devanagari]
```

- if you use different languages/scripts a lot, check out the polyglossia package

Glossed examples and glossaries

- packages/style files:
  - gb4e – for glossing
  - cgloss – style extension for gb4e
  - leipzig – standardized glosses and automatic glossary
  - glossaries – for glossaries
- before doc begins, we add \makeglossaries so LaTeX knows to build one
- before the bibliography, we add \printglossaries to output it

```
% this loads the linguistic stuff
% for doc consult http://mirror.switch.ch/ftp/mirror/tex/macros/lat
   \usepackage{gb4e, cgloss}
% formatting the examples for nicer look
      \let\eachwordone=\itshape % italic data line (default is \rm)
      \let\eachwordtwo=\small % for small print glosses
```

```
% using leipzig glosses
   \usepackage{leipzig}
      \newleipzig{conv}{conv}{converb}


% tells LaTeX to actually make the glossary
   \makeglossaries
```

Glossed examples and glossaries

- glossed examples rely on the environment *exe* and *xlist* (for nested examples)
- \ex starts the example, you can add a description
- gll starts the aligned part; per default this is a line for the language data and one for the glosses; additional lines can be addes with l, e.g. \glll
- \glt starts the translation line
- always add a label
- fun thing: all the formatting is done by LaTeX!

```
\begin{exe}
   \ex Bernese German \\
   \gll Bärndütsch isch di beschti sprach! \\
   Bernese.German \Cop.\Tsg{} \Art.\Def.\F{} best.\Conv.\F{} language      \\
   \glt `Bernese German is the best language!' \hfill (Source of Ex.)
   \label{ex:first}
\end{exe}
```

(5) Bernese German
Bärndütsch isch di beschti sprach!
Bernese.German COP.3SG ART.DEF.F best.CONV.F language
'Bernese German is the best language!' (Source of Ex.)

Glossed examples and glossaries

- nested examples with xlist within the exe environment
- each examples should have a label!
- otherwise works just the same

```
\begin{exe} \ex
  \begin{xlist}
  \ex \gll Gli isch summer.   \\
     soon \Cop.\Tsg{} summer        \\
  \glt `Soon it's summer.' \hfill \citep[]{}
  \label{ex:third}
  \ex \gll Was isch los?  \\
     what \Cop.\Tsg{} loose          \\
  \glt `What's happening?' \hfill \citep[]{}
  \label{ex:fourth}
  \end{xlist}
\end{exe}
```

Glosses with the Leipzig package

- leipzig has a set of predefined glosses, but you can add your own
- new glosses are added in the preamble
- to use the glosses in examples you need a slash, the gloss abbreviation types with initial capital letter
- curly brackets {} after if there is nothing following (manages spacing)

A Pre-defined abbreviations

Command	Short	Long
\First{}	1	first person
\Second{}	2	second person
\Third{}	3	third person
\Abl{}	ABL	ablative
\Abs{}	ABS	absolutive
\Acc{}	ACC	accusative
\Adj{}	ADJ	adjective

```
% using leipzig glosses
  \usepackage{leipzig}
    \newleipzig{conv}{conv}{converb}
```

```
\begin{exe}
  \ex Bernese German \\
  \gll Bärndütsch isch di beschti sprach! \\
  Bernese.German \Cop.\Tsg{} \Art.\Def.\F{} best.\Conv.\F{} language \\
  \glt `Bernese German is the best language!' \hfill (Source of Ex.)
  \label{ex:first}
\end{exe}
```

Glossed examples - Practice

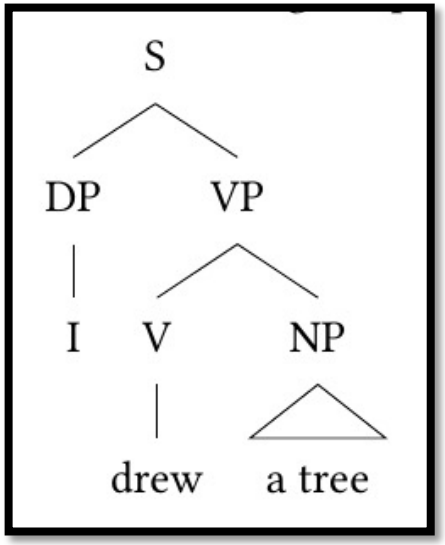
- for non-linguists: if you don't know what glossing is, you can skip this and just have a look at the examples provided in the document

- add a glossed example with a sentence in your native language
- make sure to have a line spelling out what language it is
- add at least one new gloss
- compile! (might need two runs to resolve everything correctly)
- if the glossary doesn't show up run (in TexStudio): Tools > Glossary (F9)

Drawing trees

- packages:
  - tikz
  - forest (with option [linguistics] if desired;
- tree topologies are set up by square bracket []
- the node label is placed immediately after the opening bracket [
- there are many customizing options, e.g. roof to make a little triangle

```
\begin{forest}
  [S
    [DP [I]]
    [VP
      [V [drew]]
      [NP [
        a tree, roof]
      ]
    ]
  ]
\end{forest}
```



Drawing trees - Practice

- add \usepackage{tikz} and \usepackage{forest} to your preamble

- draw a simple tree
  - with a head node
  - and at least two leaves
  - if you know some syntax trees, you can draw a simple one in your preferred theory
  - you can also draw a small family tree of languages
  - or just use arbitrary labels

- compile!

More cool things that can be done but that we can't cover today

- OT tableaus
- annotating the pdf with notes etc.
- adding todo's in the tex doc that can be compiled into a separate checklist
- integrating data from csv-files
- making slides (beamer)
- designing posters
- knitr for directly combining R code and LaTeX in one document
- modifying style files and making your own
- …

Question and discussion time / Free practice time