Getting Started with LaTeX

Or, why you might be interested in this LaTeX thing

Curt Anderson

Michigan State University

September 22nd, 2015

Introduction

What this workshop is:

- Introduction to LaTeX
- Why you might care to use LaTeX
- A quick start guide on getting a document set up
- Pointers on packages linguists use
- Where to go next

What this workshop is not:

- Extensive help on setting up LaTeX on your computer
- An extended LaTeX tutorial

These slides will be available at
http://www.msu.edu/~ande1472/latex.

What is LaTeX?

LaTeX is not a word processor!
LaTeX is a document typesetting system.

- Write specially formatted text
- Text gets sent to a program that can interpret it
- A document gets spit out (PDF is the most popular now)

It's a bit like your web browser: in goes HTML, and out comes the visual presentation of the website.

Why use LaTeX?

Why use LaTeX rather than Word, OpenOffice, or something else?

- It's free! (As in no cost)
- It's free! (As in open source software)
- It's cross-platform (Mac, Linux, Windows, and others)
- Your document is a simple text file, which makes it easy to share and edit.
- The end result looks nice.
- Lots of people in linguistics use it — specialized tools for linguists!

Why not use LaTeX?

There's some caveats.

- Takes some effort to set up, depending on your OS
- Has a non-negligible learning curve (but you don't have to learn everything at once!)
- Each document requires a bit of work to set up. Ratio of code to actual important stuff can be high in small documents.

Where to get it

LaTeX comes in distributions that contain the compiler, some set of packages, and often other utilities.

- On Windows: Get MikTeX.
- On Mac OS X: Get MacTeX.

Both MikTeX and MacTeX come preconfigured, and they each also come packaged with a decent editor (place to type your code).

Structure of a LaTeX document

Every document is divided into two sections, a preamble and a document body.

- Preamble
  - Document type (book, article, and other types)
  - Commands that set up the overall look of the document.
  - Specify packages for specialized things you need to do.
- Body: everything you want to say

How the code works

Semantic markup:

- What the code is (ideally) doing is specifying how things get interpreted.
- Specify high-level, meaningful roles for pieces of the document
- Titles, section headings, linguistic examples, lists, emphasis, ...
- This contrasts with specifying independently how each bit appears on the page
- Creates a document that's pretty readable by a human
- Makes it easy(ish) to make changes to how the thing looks in the future.

How the code works

Different types of code:

- **Commands** starts with a backslash and then some name of a command.
  - May need other information (arguments) via {..}, or have options set via [..].
  - Example: `\textbf(blah)` has output **blah**
- **Environments** start with `\begin{`*`identifier`*`}`, end with `\end{`*`identifier`*`}`, and have some code in the middle.
  - Enclose chunks of code.
  - May also have arguments or options.
  - Used when a bunch of stuff is related (numbers in a table, for instance, or items in a list)
  - Example: `\begin{document}` .. `\end{document}`, which defines where the contentful bits of a document are.

Sample document

Here's a very very small LaTeX document.

A sample document

```
\documentclass[12pt,letterpaper]{article}
\usepackage{times} % Use the Times New Roman font

\begin{document}
    \title{Hello World}
    \author{Me} % Hey, this is me!
    \date{January 1st, 1970}
    \maketitle

    \section{My first section}
        \subsection{Point one}
        This is a subsection. \textbf{This is bold.}

        \subsection{Point two}
        Here's another one. \textit{This is italics.}

    \section{Another section}
    Here I say stuff!
\end{document}
```

Sample preamble

The preamble in the sample is small:

- `\documentclass[12pt,letterpaper]{article}` sets up the document as being of type `article`, with a 12pt font and using letter-sized paper.
- `\usepackage{times}` says we are using the `times` package, which in turn sets the font to Times New Roman.
- Other packages you might want to use, or setting options that describe the look of the document, would (usually) go in the preamble.

The sample's preamble

```
\documentclass[12pt,letterpaper]{article}
\usepackage{times} % Use the Times New Roman font
```

Sample body

The sample's body

```
\begin{document}
    \title{Hello World}
    \author{Me} % Hey, this is me!
    \date{January 1st, 1970}
    \maketitle

    \section{My first section}
        \subsection{Point one}
        This is a subsection. \textbf{This is bold.}

        \subsection{Point two}
        Here's another one. \textit{This is italics.}

    \section{Another section}
    Here I say stuff!
\end{document}
```

Sample body explained

The commands:

- The entire body is surrounded by `\begin{document}` and `\end{document}`.
- `\title`, `\author`, and `\date` are self explanatory. `\maketitle` puts these together in the title.
- `\section` and `\subsection` (there's also `\subsubsection`) define where the sections in your paper are.
- `\textbf` and `\textit` set boldface and italics, respectively.

Sample body explained

Other things to note:

- `%` says that everything after the `%` on the same line is to be ignored (this called a comment).
- Capitals matter, but whitespace generally doesn't (extra spaces, tabs, and newlines aren't meaningful).
- Exception: leaving a blank line between paragraphs lets the compiler know that you have a paragraph.

Paragraphs

Paragraphs

```
This is a paragraph.

This is another paragraph. You can tell because
   there's a blank line between this one and the
   paragraph right before it.

And finally, here's yet another paragraph. All you
   have to do is have a blank line.
```

This is a paragraph.

This is another paragraph. You can tell because there's a blank line between this one and the paragraph right before it.

And finally, here's yet another paragraph. All you have to do is have a blank line.

Other things you may need

- Lists:
  - `itemize` environment and `\item` for each list item.
  - `\enumerate` environment allows you to change the list label, starting number and other things (use package `enumitem`).
- Tables:
  - `tablular` environment
  - Use the `booktabs` package for nicer tables.
- Images: `\includegraphics` from the `graphicx` package.

itemize example

itemize example

```
\begin{itemize}
    \item item a
    \item item b
    \item item c
        \begin{itemize}
        \item nested 1
        \item nested 2
        \end{itemize}
    \item item d
\end{itemize}
```

- item a
- item b
- item c
  - nested 1
  - nested 2
- item d

enumerate example

```
\begin{enumerate}[label=(\roman*)]
    \item item i
    \item item ii
\end{enumerate}

\begin{enumerate}[label=(\Alph*)]
    \item item A
    \item item B
\end{enumerate}

Hello world

\begin{enumerate}[label=(\Alph*), resume]
    \item item C
\end{enumerate}
```

tabular example

tabular example

```
\begin{tabular}{l c r}
left aligned & center aligned & right aligned\\
Meijer       & gourmet       & crackers\\
light        & flakey        & buttery\\
3            & stay fresh    & packs
\end{tabular}
```

left aligned	center aligned	right aligned
Meijer	gourmet	crackers
light	flakey	buttery
3	stay fresh	packs

tabular example using booktabs

tabular example using booktabs

```
\begin{tabular}{l c r}
\toprule
left aligned & center aligned & right aligned\\
\midrule
Meijer       & gourmet        & crackers\\
light        & flakey         & buttery\\
3            & stay fresh     & packs\\
\bottomrule
\end{tabular}
```

left aligned	center aligned	right aligned
Meijer	gourmet	crackers
light	flakey	buttery
3	stay fresh	packs

Using images

\includegraphics from graphicx

```
\includegraphics{code/crackers.jpg}
\includegraphics[scale=.5]{code/crackers.jpg}
```

Now, for the cool stuff for linguists:

- Numbered examples: `gb4e`
- Glosses: `gb4e`
- OT tableaux: `ot-tableau`
- Tree structures: `tikz-qtree`
- Logic/math: nothing special needed

Numbered Examples

Packages: gb4e (recommended), covington, linguex

Example with gb4e

```
\begin{exe}
\ex Here is an example. \label{ex-1}
\ex[*]{Here another.}
\ex Subexamples.
    \begin{xlist}
    \ex[]{No good.}
    \ex[??]{Marginal.} \label{marginal ex}
    \end{xlist}
\end{exe}

Here's a reference to example \ref{ex-1}. Here's a
   reference to \ref{marginal ex} now.
```

Numbered Examples

Packages: `gb4e` (recommended), `covington`, `linguex`

Output

(1) Here is an example.

(2) * Here another.

(3) Subexamples.

a. No good.

b. ?? Marginal.

Here's a reference to example 1. Here's a reference to 3b now.

Glosses

In addition to examples, `gb4e` supports glosses (demonstrated here), as do `covington` and `linguex`.

Gloss with `gb4e`

```
\begin{exe}
    \ex
    \gll Den Fritz habe ich zum Essen eingeladen.\\
    the fred have I {to the} eating invited.\\
    \glt I invited Fred for dinner.
\end{exe}
```

(4) Den Fritz habe ich zum Essen eingeladen.
the fred have I to the eating invited.
I invited Fred for dinner.

Optimality Theory Tableaux

Popular packages: `ot-tableau`, `OTtablx`

A tableau with `ot-tableau`

```
\ShadingOn
\begin{tableau}{c|c}
\inp{\ips{ba}}          \const{*VcdObs}
   \const*{\textsc{Ident-IO}-[nas]}
\cand{ba}               \vio{*!}          \vio{}
\cand[\HandRight]{pa}   \vio{}            \vio{*}
\end{tableau}
```

/ba/	*VcdObs	Ident-IO-[nas]
a. ba	*!	
☞ b. pa		*

OT Tableaux

More complex tables are possible, too:

A more complex tableau

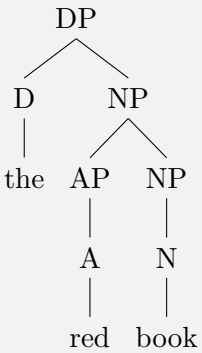
/stap/	*Complex	Anchor-IO	Contiguity-IO
a. stap	∗!		
☞ b. sap			∗
c. tap		∗!	

Trees

Popular packages: `tikz-qtree` (recommended), `forest`
If you can bracket a sentence, you can do a tree in LaTeX.

Tree using `tikz-qtree`

```
\Tree [.DP [.D the ]
    [.NP [.AP [.A red
    ]] [.NP [.N book
    ]]]]
```



Trees

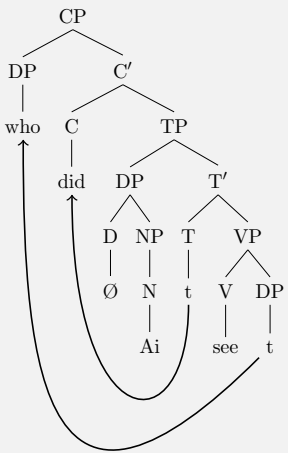
Movement is possible, too, by wrapping the tree in a `tikzpicture` environment and using `tikz`'s line-drawing capabilities.

Tree with movement

```
\begin{tikzpicture}
    \Tree [.CP [.DP \node(wh){who}; ]
    [.C$'$ [.C \node(C){did}; ]
    [.TP [.DP [.D {\O} ] [.NP [.N Ai ]]]
    [.T$'$ [.T \node(T){t}; ]
    [.VP [.V see ] [.DP \node(wht){t}; ]]]]]]
    \draw[semithick,->] (wht)..controls +(south west:6)
        and +(south:5) .. (wh);
    \draw[semithick,->] (T)..controls +(south:3)
        and +(south:5) .. (C);
\end{tikzpicture}
```

Trees

Output



Math stuff

No need for additional packages! LaTeX has the ability natively to typeset math (it was made for it).

- Two typesetting modes in LaTeX: mathmode and textmode
- Textmode is the default mode. You write most of your document in textmode.
- In mathmode, you get access to special notation and symbols for typesetting mathematics.

Math stuff

Surround mathmode code with `$..$`.

Some semantics

```
% Some simple stuff
Consider the formula $\exists x [f(x) \wedge
    g(x)]$.

% Can be wrapped in an example.
\begin{exe}
    \ex $\forall x [f(x) \rightarrow g(x)]$
\end{exe}
```

Consider the formula $\exists x[f(x) \wedge g(x)]$.

(5) $\forall x[f(x) \rightarrow g(x)]$

Math stuff

Some useful symbols (but there's many more):

- Existential quantifier: $\exists$, `\exists`
- Universal quantifier: $\forall$, `\forall`
- Negation and lambda: $\neg$ and λ, `\neg` and `\lambda`
- Denotation brackets: $[\![$ and $]\!]$, `\llbracket` and `\rrbracket`. These require the `stmaryrd` package.
- Conjunction and disjunction: $\wedge$ and $\vee$, `\wedge` and `\vee`
- Angle brackets (for types): $\langle$ and $\rangle$, `\langle` and `\rangle`
- Set theory symbols:
  - Curly brackets: {}, \{, \}
  - Union and intersection: $\cup$ and $\cap$, `\cup` and `\cap`
  - Subset and proper subset: $\subseteq$ and $\subset$, `\subseteq` and `\subset`
  - Element of: $\in$, `\in`
- Greek letters usually go by their names (α, `\alpha`).

If there's something you need, you can find it by drawing it in the square on DeTeXify.

Math stuff

Math and text can be mixed together. Use \text to briefly jump back into textmode. \mathbf and \mathit can be used for boldface and italics in mathmode.

Some more semantics

```
$\llbracket \text{dog} \rrbracket = \lambda x
    [\text{$x$ is a dog}]$

$\llbracket\textit{every dog}\rrbracket^{w,g} =
    \lambda P_{\langle e,t\rangle} \forall x
    [\mathbf{dog}(x) \rightarrow P(x)]$
```

$[\![\text{dog}]\!] = \lambda x[x \text{ is a dog}]$

$[\![\textit{every dog}]\!]^{w,g} = \lambda P_{\langle e,t \rangle} \forall x[\mathbf{dog}(x) \rightarrow P(x)]$

Bibliographies

It's possible to set up automatically formatting bibliographies as well using LaTeX and BibTeX.

- BibTeX is a system for managing bibliographies. Separate from LaTeX but often used it with it.
- You manage a database with information for each reference, such as authors, paper title, journal published in, year, and so on.
- Each of these references has a citekey, a unique code that acts as a name for the reference (e.g., something like `karttunen1977`).
- Use special commands+citekeys in your LaTeX document to cite things.

Here's a taste of how it works, but this is probably not enough to quite get you started.

Bibliographies

- `natbib` is probably most popular way to do references, but there are others.
- `natbib` needs you to set a style for your bibliography. I like APA, but there's others.
- This goes in your preamble.

Preamble

```
\usepackage{natbib}
\bibliographystyle{apalike}
```

Bibliographies

- `natbib` includes commands for putting in citations.
- `\citet` is a textual citation, like Lasersohn (1999).
- `\citep` is a parenthetical citation, like (Lasersohn, 1999).
- `\bibliography` tells the BibTeX compiler where to look for my references (a file I call `Papers.bib`).
- Finally, there's a multistage process to link up the output from LaTeX and BibTeX:
  - Document gets compiled by LaTeX compiler.
  - A file output by the compiler gets sent to the BibTeX compiler.
  - Then, your LaTeX is compiled again (and possibly once more after this).

Bibliographies

Using a bibliography

```
\citet{lasersohn1999pragmatic} notes that \dots

\dots punctual verbs individuate
   \citep{barner2008}.

\bibliography{Papers}
```

Lasersohn (1999) notes that ...
...punctual verbs individuate (Barner et al., 2008).

Barner, D., Wagner, L., and Snedeker, J. (2008). Events and the ontology of individuals: Verbs as a source of individuating mass and count nouns. *Cognition*, 106(2):805–832.

Lasersohn, P. (1999). Pragmatic halos. *Language*, 75(3):522–551.

Bibliography Management

- The `.bib` file where you keep references is human readable, but kind of ugly to work with.
- Most people will manage their bibliographies with bibliography management software.
- BibDesk on Mac OS X is good. JabRef is also popular.
- These programs can also manage libraries of PDFs and other documents, too.

Other cool stuff

- Do you do a lot of stats? R and LaTeX can integrate through `knitr`. Talk to Adam Liter (I'm still figuring it out).
- Need to project slides? `beamer` can help. (I used it here.)
- Want to typeset your CV? There's a class (`moderncv`) and a bunch of templates for it.
- Some journals have LaTeX stylesheets (e.g., *Semantics and Pragmatics*, and *Natural Language and Linguistic Theory* and other Springer journals).
- Increasing number of conference proceedings do, too (e.g., *Semantics and Linguistic Theory*, *Sinn und Bedeutung*). If you're submitting something, check the author information for LaTeX templates.
- Alan maintains a class for theses and dissertations here at MSU (`msu-thesis`).

Where to Go From Here

Important:

- Practice!
- Don't try to learn everything at once. Learn as you go.
- Practice with class assignments, class papers.

Where to get help:

- Google. Someone's encountered your error before.
- Google. Someone's done what you want to do.
- Documentation for packages that are giving you trouble.
- StackExchange. Community of experts on LaTeX.
- Others around the department (me, Adam, Ai, Marcin, Alan)

Other Resources

- Alan Munn's One Page Dictatorial Guide to LaTeX Packages.
  http://www.msu.edu/~amunn/latex/latex-guide.pdf
- DeTeXify. Draw the symbol you're looking for.
  http://detexify.kirelabs.org/
- LaTeX for Linguists. Kind of dated now, though.
  http://www.essex.ac.uk/linguistics/external/clmt/latex4ling/

Packages Mentioned

Most packages you'll use are already on CTAN. Check there for documentation.

- gb4e: http://www.ctan.org/pkg/gb4e
- linguex: http://www.ctan.org/pkg/linguex
- covington: http://www.ctan.org/pkg/covington
- ot-tableau: http://www.ctan.org/pkg/ot-tableau
- OTtablx: http://sanders.phonologist.org/OTtablx/
- tikz-qtree: http://www.ctan.org/pkg/tikz-qtree
- forest: http://www.ctan.org/pkg/forest
- booktabs: http://www.ctan.org/pkg/booktabs
- enumitem: http://www.ctan.org/pkg/enumitem
- natbib: http://www.ctan.org/pkg/natbib