XLingPaper's use of TEX Technologies

H. Andrew Black, Hugh J. Paterson III

Abstract

We discuss the use of TEX technologies by XLingPaper, an authoring tool for producing academically oriented publications with features required for linguistic publishing. We present the TEX modules used and the rationale for the history of XLingPaper development.

1 Introduction

Within the publishing industry, there are several notable products for producing complex documents in beautiful formats. TEX [25] [26] is one of the well-known publishing technologies used to meet these needs. Since 2000, XML-based technologies such as XSL-FO[1] or the TEXML[2] project [30] have been used to integrate content and compose complex documents such as textbooks and maintenance manuals. Requirements for composing these large, interlinked documents birthed the development of tools like XMLmind,[3] the <oXygen/> XML Editor,[4] and Xpublisher.[5] These tools can be used to compose content within predefined XML structures. XLingPaper, as discussed in [7] [8] [9], seeks to provide a constrained environment in which authors of complex works dealing with language descriptions and linguistic analyses can focus on content structure independently from the styling requirements of documents. In this way the underlying design principle of XLingPaper maximizes the SGML design practice of separating content from presentation. With XLingPaper, authors can keep content structure independent from page layout information and thereby provide maximal transferability between publishing styles. The software does this while providing authors a clear structured interface for authoring content.

XLingPaper is designed to reduce friction in the process of writing, composing, and publishing linguistic papers, grammars, and books by removing common time-sinks related to inconsistent formatting (especially citations, references, and numbered elements like examples). A full list of benefits to all parties in the publishing work flow is available [9].

The XLingPaper software has a growing number of users who have successfully typeset complex documents that, among others, include:

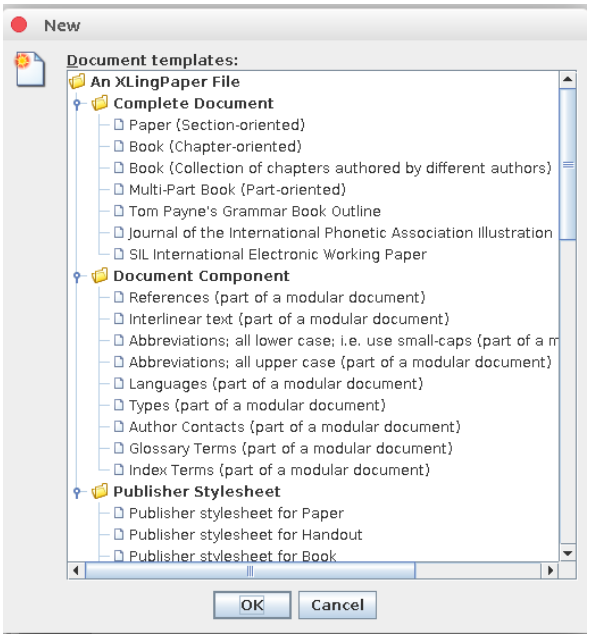


Figure 1: XLingPaper predefined document types via DTD

- master theses [55] [27] [36],
- doctoral dissertations [17] [41],
- textbooks [32],
- linguistic grammars [11],
- books [1] [42],
- journal articles [10] [37], and
- bilingual software documentation [2] [3].

2 What is XLingPaper?

XLingPaper[6] is a plug-in to the XMLmind XML Editor. XLingPaper benefits from the XMLmind XML Editor's Java-based implementation which allows it to be used on Mac OS X, Windows, and Linux. Via a DTD, XLingPaper defines several document classes (articles, books, chapters, etc., as illustrated in Figure 1), in each case providing document layout sections (paragraphs, examples, endnotes, etc.). Figure 2 illustrates the main screen of the user-interface of XMLmind XML Editor. By using this interface, formatting errors are reduced because users are constrained on where in the document flow they can introduce block and line level document elements. That is, first, authors cannot input page layout instructions directly into the document and second, the introduction of layout sections within the document flow is constrained via the DTD.

[1] w3.org/TR/xsl11
[2] getfo.org/texml
[3] xmlmind.com/xmleditor
[4] oxygenxml.com/xml_author.html
[5] xpublisher.com/products
[6] software.sil.org/xlingpaper

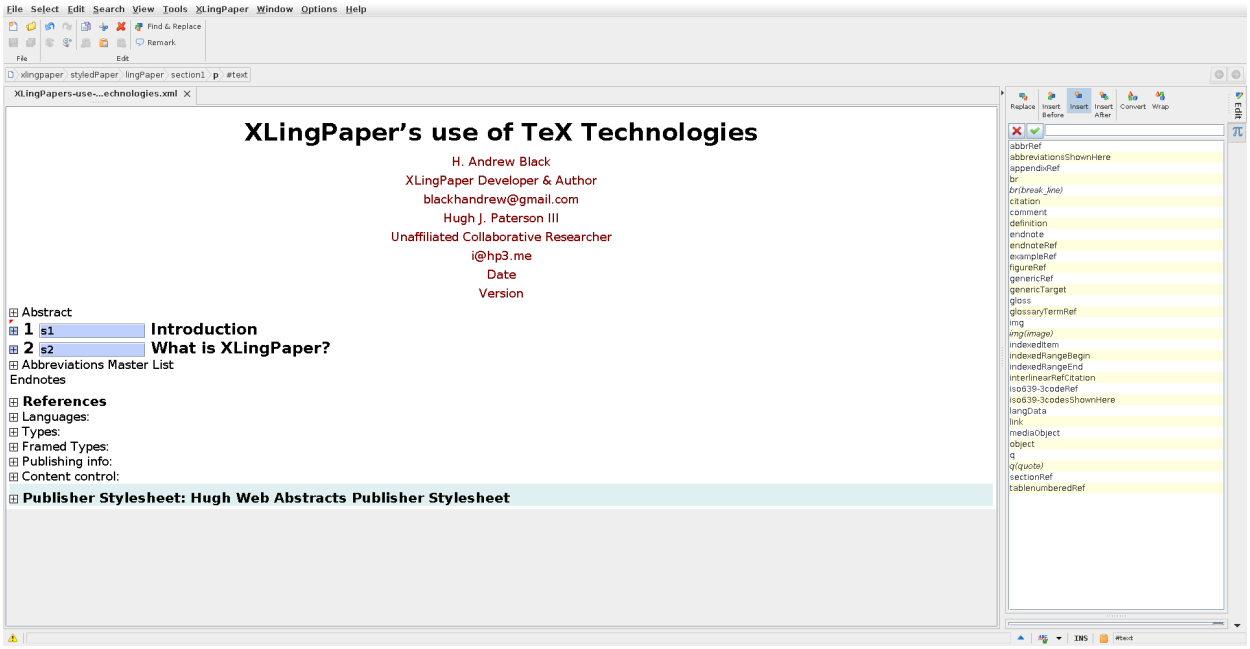


Figure 2: XLingPaper's user interface. Left side: document content editing. Right side: block and line level units available for use at the cursor location.

For many, the PDF format is the quintessential file format for final distribution of publishing outputs. XLingPaper supports PDF production; however, as illustrated in Figure 3, XLingPaper can also produce documents with at least five outputs, all from the same source document:

- PDF (version 1.5),
- Web pages (HTML 4),
- Microsoft Word (.doc),
- Open Office Writer Document (.odt), and
- ePUB.

XLingPaper automatically numbers tables, examples, figures, and sections. It keeps track of internal references to these entities along with citation references, abbreviations, and gloss abbreviations. This keeps numbering and reference links dependable and automated. XLingPaper also automatically generates indexes, a table or list of abbreviations used, and a section for references cited (using a custom references implementation).

Unlike most editing programs which are based on either the WYSIWYG paradigm or are unconstrained text editors such as those used to code or produce Markdown, XLingPaper (via the XMLmind XML Editor) is a structured editor much more like the block editors we see in tools like MailChimp[7] or WordPress's Gutenberg editor,[8] albeit without the drag-and-drop features. Rather than visually structuring the document to look the way it is to be formatted, the author "marks up" the items in the document according to their kind. One of the many benefits that using a DTD provides is that there is a "grammar" of what a well-formed linguistic document looks like. This makes moving, replacing, switching, or reordering sections, chapters, tables, figures, and examples less error prone because it prevents users from inadvertently creating ill-formed documents. The following sections of this paper discuss the TEX technologies used by XLingPaper.

3 XLingPaper and TEX

Linguistic publishing has unique requirements when compared to general publishing. The following sections provide more detail on the linguistic publishing context, design requirements and LATEX packages used by XLingPaper.

3.1 TEX and Linguistic document production

TEX has long been embraced by linguists. Peter [39] writes of a personal communication with Don Knuth where Knuth suggests that linguists were some of the earliest adopters outside of mathematicians. Thiele [50] in an interview given in 2007 states that she was typesetting linguistic journals via TEX in 1983—a date prior to the release of Knuth's book on using the TEX typesetting system [25]. Thiele [49]

[7] mailchimp.com
[8] developer.wordpress.org/block-editor

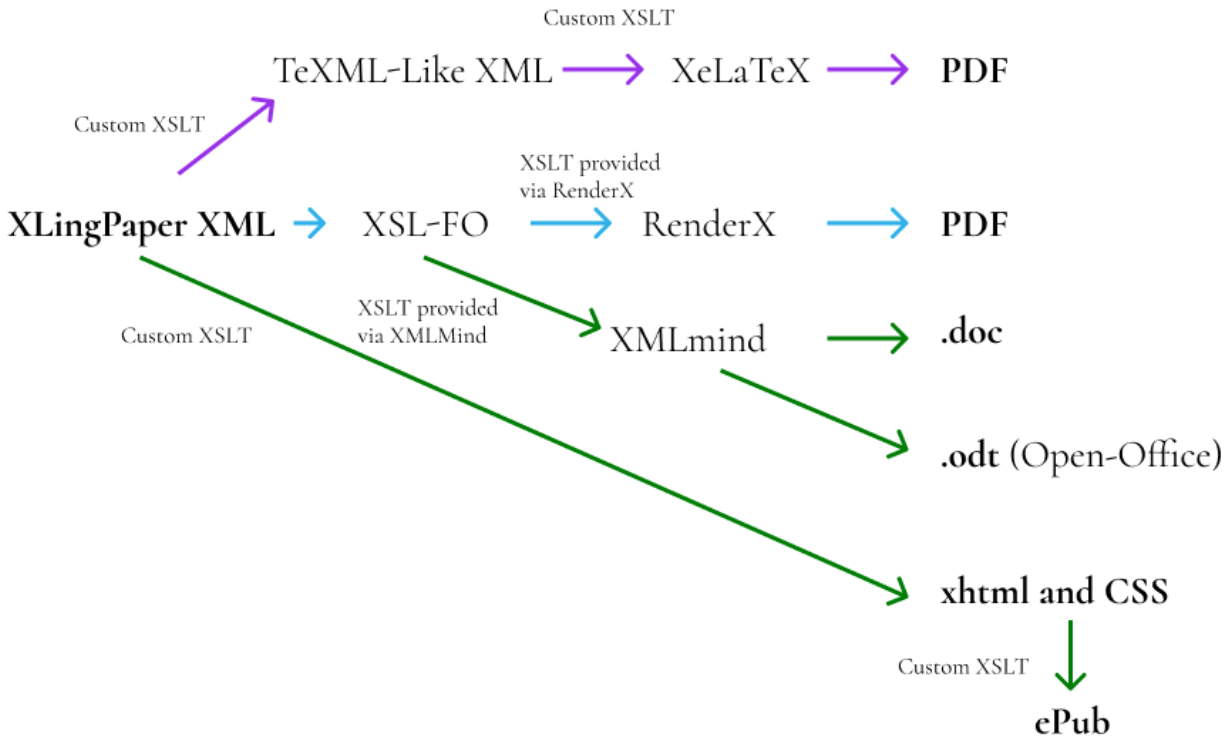


Figure 3: XLingPaper's data processing pipeline to multiple formats

gives an early overview of TEX use in linguistics with mention of significant repositories outside of CTAN. A slightly more recent (2004) update by Peter [39] provides some additional tips and tools for typesetting common information structures in linguistic publishing. The TEX community has produced many packages which have shaped the visual face of publishing in linguistics, including `tipa`[9] by Rei [43], which provided access to an excellent typeface for phonetic transcriptions, and `pst-asr`[10] by Frampton [18] for autosegmental representations. Some packages used in linguistic publishing are special purpose but are not exclusive to linguistics. For example, Donnelly [16] describes how to use various packages to draw phonetic pitch traces using TEX. Peter [39] and Thiele [49] list out and review (through 2004) various packages across several areas of linguistics. Among others, they discuss several packages used to draw syntax trees such as `qtrees`[11] and `forest`[12] and specialized packages for presenting examples and interlinear glossed texts such as `expex`.[13] Their reviews also discuss packages such as `covington`[14] and `gb4e`[15] whose collections of macros serve a variety of page layout functions targeted at publishing in linguistic topics.

The CTAN repository currently lists fifty-four different TEX packages for linguistic typesetting,[16] though some of these packages also include capabilities targeted as multi-lingual or multi-script publications or are specific style sheet implementations for publications at linguistic programs at institutions of higher education (there may be more packages which are not tagged but should be). Several of the packages tagged "linguistic" pre-date Unicode [51] but still see significant use. Sometimes it is the case that secondary packages are developed in an attempt to "fix" publishing outputs in different ways to bring Unicode features along with the features of the original package. For example, `tipa` is not Unicode compatible, but the packages `unitipa`[17] and `tipauni`[18] seek to address different implications of not publishing with Unicode while giving access to the beautiful typeface of `tipa`. Understanding the long history of publishing and the interdependency

[9] ctan.org/pkg/tipa
[10] ctan.org/pkg/pst-asr
[11] ctan.org/pkg/qtrees
[12] ctan.org/pkg/forest
[13] ctan.org/pkg/expex
[14] ctan.org/pkg/covington
[15] ctan.org/pkg/gb4e
[16] ctan.org/topic/linguistic
[17] ctan.org/pkg/unitipa
[18] ctan.org/pkg/tipauni

that packages have (including the order of loading packages) constitutes barriers of adoption to new TEX users.

We discuss TEX barriers of adoption for two reasons. First, it exemplifies some of the complexities that XLingPaper seeks to simplify as it presents authors not just a visual environment for document composition, but also a cohesive output solution. Second, it speaks to the software design process in finding the minimal viable product. That is, *how much (or little) of a software stack is needed to make a usable software product for linguistic publishing?* The TEX community is divided on this. While the diagrams in linguistic books and journals since the 1980's exemplify many beautiful, sharp, crisp, illustrations created directly in TEX, many trainers of TEX tools,[19] but not all,[20] have steered authors towards a more generic set of packages which do not include specific diagram creating macros. Rather, they suggest that authors use secondary illustration tools to generate illustrations and then include them as vector PDFs or images. In fact this second method is the document production path that the XLingPaper philosophy follows. That is, XLingPaper reduces the complexity of the typesetting task for authors by requiring complex visualizations to be produced via graphical tools. We have found tools like Figma[21] and Inkscape[22] very helpful in the graphic production task. The XLingPaper product seeks to lower barriers of entry, only produce valid documents, and keep the code base to a minimum.

As mentioned in the discussion of `tipa`, linguistic documents have not always been typeset with Unicode. Unicode was introduced in 1991 and by the early 2000's Unicode along with document and data storage in XML formats were being heralded in academic linguistics as a best practice in order to avoid vendor lock-in, increase interoperability across use cases, and to separate data life-cycles from encoding or software life-cycles [5] [6] [52]. Due to the heavy reliance on Unicode by today's practitioners of language documentation and linguistic work, XLingPaper specifically uses XeLaTeX and compatible packages to produce PDF outputs. This brings continuity to the text input process for users across their workflows. It also makes importing and using language or phonetically transcribed examples simpler by removing the need to use macros to derive characters.

3.2 Design desiderata for XLingPaper outputs via TEX

Three goals have driven the development of XLingPaper:

- separation of content and style,
- software accessibility (license and size), and
- beautiful multi-format outputs.

Deciding how TEX technologies fit within the project has been a journey. XLingPaper development started in 2001 without any use of TEX technologies. In 2006, XLingPaper added XSL-FO for PDF production. Prior to 2009, XLingPaper used RenderX's XEP[23] product to produce PDF documents. As far as we know, there are two cross-platform XSL-FO processors written in Java: RenderX's XEP application and the Apache FOP project.[24] Using a Java implementation reduces the size of the required stack because the XMLmind XML Editor requires Java. XSL-FO processors can have various degrees of implementation of the XSL-FO standard. RenderX has some limitations which affect page layout but has more complete coverage than the Apache FOP project which lacks certain required table-oriented capabilities.[25] The limitations of RenderX are discussed in Section 5. In 2009 plans were made to add XeLaTeX-based output to XLingPaper because, while there was a free version of RenderX, the output contained a watermark. By implementing the ability to export to PDF via XeLaTeX, watermarks could be avoided all together. The XeLaTeX method of PDF production is now the default method to produce PDF documents, although the RenderX method is still possible.

Maintaining a separation of content and style in the XLingPaper environment was a key design requirement. When the XeLaTeX method of PDF production was introduced, XLingPaper already had a way to format output per a user-created publisher style sheet — allowing great flexibility due to the separation of style and content. Using TEX technologies meant the developer (Andrew Black) needed to be able to map from an XLingPaper publisher style sheet to XeLaTeX. It was known that LaTeX was the ideal TEX implementation to target. However, pure LaTeX came with predefined output formatting for front matter, chapters, sections, back matter, etc. Pure LaTeX, then, would not allow direct control of

[19] Among others, see the Linguistics Dissertation guide for the University of Hawai'i at Mānoa [21], University of Pennsylvania [15], and Language Science Press Guidelines [35].

[20] For counter examples see [29] [47] [20] and [40].

[21] `figma.com`

[22] `inkscape.org`

[23] `renderx.com/tools/xep.html`

[24] `xmlgraphics.apache.org/fop/index.html`

[25] `xmlgraphics.apache.org/fop/compliance.html`

formatting of all of these per an XLingPaper user-defined publisher style sheet. This required overriding these standard features of LaTeX with a custom implementation of the TeX commands needed to control formatting. XLingPaper takes a custom approach in implementing flexibility here. Table 1 lists the custom commands implemented.

The programmer of XLingPaper recently discovered `memoir`[26] [53] [54]. As a package, `memoir` accomplishes many of the same tasks and could be considered to replace some of the custom code if it were shown to be easy to implement and that the size of the total XLingPaper code base would be reduced.

The distributability of the software was also seen as a design requirement. Distributability is understood to have two components: license and accessibility, including size.

From the outset, XLingPaper was designed to be costless to the end user. It is licensed under the MIT license, and its code is currently available on Github.[27] The XMLmind XML Editor had a costless Personal Use License that met this requirement for the vast majority of the target audience of XLingPaper. The few XLingPaper users who did not meet the terms of that license most likely would be able to afford to purchase (or have their organization purchase) a professional license of the XMLmind XML Editor. The actual XLingPaper plug-in has always been free.

The software size of XLingPaper is a major design influencer. Many of the expected users of XLingPaper live and work in places around the world where Internet connections are characterized by high costs, low bandwidth capacity, and general unavailability. Therefore, the download required to install XLingPaper needed to be as small as possible. On Windows the current full XLingPaper installer is 146MB, and the XMLmind XML Editor installer is 116MB. Both are required. This stands in contrast to the TeXLive 2010 installer which has a size of about 1.2GB when downloaded and 2.38GB when uncompressed. The size constraint impacts XLingPaper because its distribution must be independent of larger mainstream TeX distribution solutions which have a large footprint. This, of course, includes TeXLive. Therefore the developer identified which LaTeX packages and binaries were needed and created a custom installation package which met the required specifications.

XLingPaper currently uses the following LaTeX packages (in alphabetical order):

`attachfile2`	`lineno`
`booktabs`	`longtable`
`calc`	`lscape`
`color`	`mdframed`
`colortbl`	`multirow`
`etoolbox`	`normalem`
`fancyhdr`	`polyglossia`
`fontspec`	`setspace`
`footmisc`	`tabularx`
`hyperref`	`xltxtra`

The twenty LaTeX packages that are part of the custom XLingPaper distribution are still rather large (29MB) for someone for whom Internet bandwidth is an expensive and inconsistent commodity.

To reduce bandwidth requirements two assumptions were made which have more or less proven to obtain. The first assumption that the developer made was that the twenty packages and binaries would not need to change over time; in contrast, the second assumption was that XLingPaper would acquire new features and need bug fixes. These assumptions resulted in an architecture where page layout information expressed in XML is translated via custom TeX commands to either TeX directly or to commands understood by LaTeX packages distributed with XLingPaper. This abstraction layer was then executed when the XeLaTeX file was processed. This middle layer has granted XLingPaper flexibility in adding new code and capabilities while keeping the "heavy" LaTeX packages stable. The net result is a "heavy" first install package (116MB), but light-weight upgrade packages (6.21MB). In the thirteen year history of development, there have been a few occasions where upgrades have required the download of new "heavy" packages. One such case was when the ability to use framed units was added. These elements depend on the `mdframed`[28] package [13]. The architecture separating stable packages from custom code, however, has generally worked out well and kept update sizes low.

3.3 PDF production

We know of two pathways for converting XML content into PDFs. The first is via XSL-FO, and the second is via TeXML which converts XML content to TeX formatted documents for further processing to PDF. Given certain limitations in both XSL-FO and TeXML, XLingPaper uses a custom (or third) method. When an author instructs XLingPaper to

[26] `ctan.org/pkg/memoir`
[27] `github.com/sillsdev/XLingPap`
[28] `ctan.org/pkg/mdframed`

Table 1: Custom commands used by XLingPaper

Command for	Purpose
Table of contents	Store and retrieve page numbers; format the contents.
Lists	Numbered and bulleted lists with control over indents, etc.
Examples	Example number and example content, where the content can be a line, a list of lines, a set of words, a list of a set of words, interlinear, a list of interlinears, etc.
Indexes	Handle keeping track of XLingPaper's indexing capability, including page numbers.
Interlinears	Handle lines in an interlinear text or example, including dealing with an ISO 639-3 code in an interlinear example.
Block quotes	Handle special cases needed for block quotes.
Table headers	Attempt to calculate a column's width via its contents.

produce PDF output via XeLaTeX, XLingPaper produces a TeXML-like XML file. This is then converted into a LaTeX formatted document via a set of XSLT transforms and processed via XeLaTeX to produce the PDF. Figure 3 contains a diagram of the data handling process.

3.4 TeXML

TeXML was discovered in the process of planning for the transition of the default PDF renderer from RenderX's XEP to XeLaTeX. Initial analysis conducted in 2009 understood TeXML to have two infelicities for use-cases required in linguistic publishing with XLingPaper:

1. TeXML has Python as a dependency and the XLingPaper developer did not want to require their users to install a version of Python specifically for TeXML. Of particular concern were potential conflicts with other installed versions of Python on user's operating systems. Moreover, the TeXML approach would make the installation package for XLingPaper much larger due to the inclusion of Python.
2. XLingPaper users require a high degree of control for white space. The fine grain control of whitespace was not immediately clear how to accomplish with TeXML.

3.5 Control characters

Even with the use of Unicode in the text of documents, there are some features of typesetting with TeX-based implementations which require the use of control characters. Additionally, XML also has control characters. In TeX these include `[`, `]`, `<`, and `>`. When transforming data between XML and TeX, TeX control characters and commands need to be escaped to ensure proper data processing. This has been implemented via Java since Java was already present in the dependency stack due to the XMLmind XML Editor requiring it. Additionally,

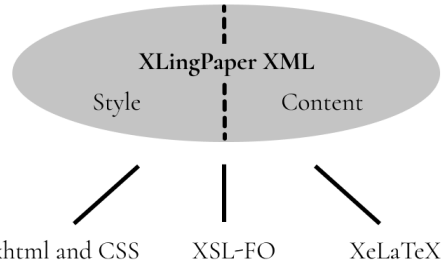


Figure 4: XLingPaper combines style and content information contained in its custom XML and then exports it into three different formats for further processing.

some small methods have been written in Java to provide additional access to features via the graphical user interface. Among other things, these include adding rows and/or columns to tables, automatically converting glosses to abbreviation references, and importing references from various XML formats.

3.6 Ling-TeX

One might ask, "Why not add more linguistic related TeX packages to the available stack, or use those instead of creating custom code?" The answer has two simple parts: First, in 2009 the linguistic capabilities of TeX packages were different than they are today. Second, XLingPaper is more than a TeX document producer. For example, some authors [2] [3] use XLingPaper to manage multilingual content on websites.

Besides TeX, XLingPaper also produces XSL-FO and XHTML/CSS outputs. When new features are considered for inclusion, they must be considered for all output formats.

After excluding TEXML as a viable option, and still seeking to create XELATEX-based output, a solution was needed to determine which set of minimal TEX packages would be needed. The Ling-TEX group,[29] which also ran the Ling-TEX mailing list from 1995–2018, was discovered.[30] Ling-TEX seemed to be the locus of activity in linguistic typesetting via TEX even though other web pages discussing linguistics and TEX also existed, e.g., Essex[31] and UPenn.[32] Today, now that the mailing list is no longer in operation, many of the mailing list participants can be found interacting on the TEX stackexchange.[33]

State-of-the-Art for TEX-based linguistic publishing in 2009, as recommended by the Ling-TEX website, suggested using `covington` and *ling-mac*—the list of macros discussed by Thiel in [49]. These macros were used to solve similar use cases, among others, to those already implemented by XLingPaper. Their approaches and outputs, however, had more limitations than what XLingPaper already offered. XLingPaper had the following capabilities for typesetting interlinears:

- no limits on the number of lines within an interlinear grouping;
- no limits on the number of free translation and literal translation lines;
- the ability to include a source reference within the interlinear; and especially
- the ability to tag interlinear items with an ISO 639-3 code for the language used in the interlinear.

At the time the best solution given the state of the TEX packages available was custom TEX scripts, although now similar features may be possible via other packages. For example in 2019 Pellard [38] discussed the limiting approaches in various TEX packages related to interlinear glosses and his solution `typgloss`.[34] XLingPaper's examples can be seen in Figures 6–7 which contain output illustrating some of the special capabilities XLingPaper offers.

4 Typesetting tasks XLingPaper users often encounter

Linguistic documents have several formatting needs that other kinds of documents do not. This section discusses some of them.

[29] `web.archive.org/web/20150702123633/http://heim.ifi.uio.no/~dag/ling-tex`
[30] `ling-tex.ifi.uio.narkive.com`
[31] `essex.ac.uk/linguistics/external/clmt/latex4ling`
[32] `ling.upenn.edu/advice/latex.html`
[33] `tex.stackexchange.com`
[34] `github.com/tpellard/typgloss`

(9)
a.	ko-sis-o	[– ¯ ¯]	move forward
b.	kɔ-kɨj-ɑ	[– ¯ ¯]	act
c.	ko-ɓund-o	[– ¯ ¯]	break
d.	kɔ-ɓʉt-ɑ	[– ¯ ¯]	become long
e.	ko-ɓeɲ-o	[– ¯ ¯]	wink
f.	kɔ-kɛk-ɑ	[– ¯ ¯]	decorate
g.	ko-sok-o	[– ¯ ¯]	cackle
h.	kɔ-mvɔɗ-ɑ	[– ¯ ¯]	suck
i.	kɔ-bɑb-ɑ	[¯ – ˊ]	carry

Figure 5: List of words as seen in [41]

4.1 Numbered example layouts

Linguistic documents usually have many numbered examples. The prose often refers to examples which are typographically nearby or to previous examples. XLingPaper automatically keeps track of the example identifiers. This is especially important in linguistic publishing because authors, and publishing style sheets, often make use of different kinds of examples, including sub-examples, and table-like design layouts which can contain lists of words along with their glosses (as shown in Figure 5) and interlinear clauses (as shown in Figure 6). Some cases even have headings in portions of the example.

4.2 Interlinear glossed texts

There is a long tradition within linguistics and language study of presenting phrases containing different languages (but the same content) as interlinear texts. Di-Biase-Dyson et al. [14] trace the practice back as far as the 1652 publication of Kircher [24]. More recent publications display significant variation in page layout related to interlinear glossed texts and interlinear examples. Variation exists in three dimensions:

- content elements,
- data-structure of the encoded elements, and
- page layout (visual display of the elements).

A full demonstration of the variation in content and its positioning across common style sheets in linguistics is beyond the scope of this paper. Significant variations include the presence or absence of the following elements:

- index elements such as example numbers or subnumbers (as shown in Figure 6),
- headings to the interlinear,
- speaker indicator,
- language indicator,
- citation indicator pointing to the larger text from which the example element is taken (see Figure 6 for an example), and

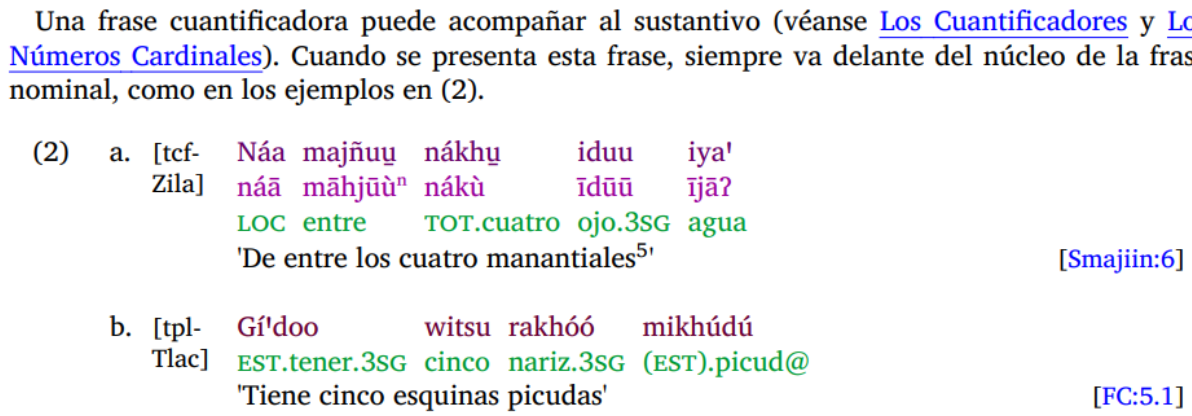


Figure 6: Interlinear example from [31]. Note the example numbers on the left followed by example groups (a) and (b). Each interlinear then also has a language indicator in square brackets. Customization allows for as many rows per group as is required. Finally, on the right the hyperlinked citation to the reference for the source text is indicated.

- limits on the number of rows in the original, gloss, translation, and free translation tiers.

Existing TEX packages approach these content requirements in different ways. As far as we can tell, the following commonly used packages for interlinear glossing all have limitations to some degree. The `expex` package does not offer a content solution for the language code or the citation. The package `langsci-gb4e`,[35] a fork of `gb4e`, supports the *Leipzig Glossing Rules*,[36] a commonly adopted set of linguistic typesetting conventions. While the Leipzig Glossing Rules do call for the language name or identifier to appear on the right hand side of the interlinear glossed text, however, it does not have a place for the citation. The package `linguex` does not have either language or citation content places built in. With these considerations, it was clear in 2009 that XLingPaper offered more to authors than any single package in the TEX ecosystem. In order to implement existing XLingPaper features, it meant creating custom TEX scripts to implement interlinear texts.

There are also some reasons related to data structure for considering XLingPaper over alternatives. Interlinear glossed texts are often stored in one of a few formats: ELAN files,[37] FLEx Text files,[38] Standard Format files,[39] LATEX files,[40] custom project-specific XML files, or relational databases such as MySQL, PostgreSQL, or FileMakerPro. Moving content from analysis and markup tools to typesetting tools is an ever present need for linguists. Several tools such as ELAN and FLEx have well-established workflows for data transfer [44]. FLEx is often considered the tool of choice for many field linguists, language documenters, and lexicographers. For many linguists entering the field, it is the tool of choice over older tools like Toolbox (which uses standard format files) due to built-in collaborative features and grammar parsers [4]. Interlinear text in FLEx can be exported into XML and the data used within XLingPaper documents. This presents FLEx users the opportunity to typeset their texts rather easily. Enabled by XML's modular document referencing features, XLingPaper documents can reference components. Using the XML document referencing strategy with XML-encoded FLEx texts allows authors to reflow typesetting outputs easily if they make content changes in their FLEx environment.

XLingPaper does not have a direct ELAN import process. However, we have had reports of linguists using the FLEx-XLingPaper publication pathway to typeset ELAN texts in LATEX documents. One user reports capturing the XELATEX document prior to rendering and then copying the relevant TEX sections to their primary document and adding any

[35] `ctan.org/pkg/langsci-gb4e`
[36] `eva.mpg.de/lingua/pdf/Glossing-Rules.pdf`
[37] `archive.mpi.nl/tla/elan`
[38] `software.sil.org/fieldworks`
[39] `software.sil.org/toolbox`
[40] For examples see [45] and [48].

required packages required by XLingPaper to the header of their primary TEX document.

Still finally, there is the matter of page layout. The main types of variation in page layout we have seen include the grouping of lines into sets or subsets (see Figure 6 for example), the labeling of sets and subsets, wrapping of interlinear glosses across lines (recall that these may themselves include three or more lines), and the alignment of the various elements of content within the interlinear glosses. We have seen word and morpheme aligned interlinears. XLingPaper automatically wraps interlinears which makes the author's job much easier. Figure 7 demonstrates the wrapping of interlinear glossed texts. It does so by formatting each aligned word in an `hbox` and then having XELATEX put them together in a hanging indent paragraph. This is based on the work of Kew & McConnel 1990 [23]. Similar examples can be seen in [33] and [34], among others.

4.3 Gloss abbreviations

Linguists use two types of abbreviations. First, they might use abbreviations for names, titles, or commonly used words. This is much like standard publishing. The second way that linguists use abbreviations is for indicating the grammatical meaning of pieces of words (morphemes). This second usage is often referred to as 'glossing' with the abbreviations referred to as 'glosses.' One common set of glosses is the *Leipzig Glosses*. Leipzig Glosses, however, are not universally used for several reasons including:

- some authors have established their own tradition within their works which they started prior to the release of the Leipzig Glosses,[41]
- the typeset examples are quoted from a database which does not use Leipzig Glosses,
- they are not comprehensive, and
- they are not theoretically sufficient for some linguists.

XLingPaper supports both types. XLingPaper approaches this by providing built-in access to Leipzig Glosses, but also allowing the author to fine-tune a set of abbreviations and their definitions. When producing the output, XLingPaper creates hyperlinks between the abbreviation and its definition. This allows readers to quickly find the meaning of glosses and for the automatic generation of a table or list of abbreviations used.

[41] For examples of the variation and scope of coverage consider the works of Greville Corbett, William Croft, Denis Creissels, and Martin Haspelmath.

4.4 Bibliographies

For better or worse XLingPaper has rolled its own bibliography solution. Import options are provided for MODS and EndNote XML formats. This enables users to import from tools like EndNote,[42] Zotero,[43] and JabRef.[44] XLingPaper uses custom TEX scripts to output TEX code for final rendering. It does not rely on BIBTEX or BIBLATEX. Figure 8 shows an example of a bibliography created with XLingPaper.

5 Outputs LATEX allow that others do not

While XLingPaper has a large array of linguistically-oriented formatting capabilities across all output formats, there are some that only the XELATEX output can produce. This is, of course, due to the formatting power of TEX and XELATEX.

5.1 Automatically wrapping interlinears

One of the most popular features of XLingPaper is its ability to automatically wrap long interlinear examples and lines in interlinear texts. As seen in Figure 7, wrapping occurs for the glossed text tiers and free translation tiers.

For the XHTML and RenderX outputs, the interlinear examples do not wrap; they run off to the right, which means completely off the page for RenderX. To fix this, the XLingPaper user must break the interlinear into smaller units by hand.

5.2 Font rendering

XELATEX renders fonts extremely well. We show three cases where XSL-FO (via RenderX) and/or XHTML outputs have text rendering issues while XELATEX does not.

First, when a line of text contains material rendered in different fonts on the same line, the two fonts may not line up evenly in the vertical direction. See Figure 9. This mismatch is due to the two fonts having different ascender and descender values. In order to overcome this when using XSL-FO, one has to add custom commands to deal with the font that differs from the primary font.

Second, the RenderX way of producing PDF cannot handle stacked diacritics, but the XELATEX way does it very well. See Figure 10.

Third, XELATEX can even handle special features requiring Graphite[45] processing. Graphite is a multi-part technology which includes a rendering engine and a rule-based grammar which compiles against a TrueType font and effectively extends the

[42] `endnote.com`
[43] `zotero.org`
[44] `jabref.org`
[45] `graphite.sil.org`

DN-1:10

आऊर	जानू	बाजा	मन	धरला,	हून मन	के	बाजाला		आऊर
aur	dʒanu	badʒa	mən	dʰərla	hun mən	ke	badʒala		aur
CONJ	PRT	N	PRT	V	PPRON	CASE	V		CONJ
and	focus	drum	=PL	take hold-3P.PTC	they	GOL	play drum-3P.PTC		and

जानू	गूलाए	बूलाला		हून	सड़क	सड़क	आऊर	फेर	जानू
dʒanu	gulae	bulala		hun	səɽək	səɽək	aur	pʰer	dʒanu
PRT	ADJ	IT		DEM	N	N	CONJ	ADV	PRT
focus	everywhere-EMP	walk around-CAUS-3P.PTC		that	road	road	and	again	focus

हाई ईसकूल	ने	ईलू।
hai iskul	ne	ilu
N	POSTP	V
high school	=LOC	come-1P.PTC

And they took hold of drums and they played the drums and made us walk all over the place, on the roads, and later we came to the high school.

Figure 7: Wrapped interlinear text as seen in [56].

Chao, Yuen Ren. 1930. ə sistim əv "toun-letəz" [A system of "tone-letters"]. *Le Maître Phonétique (Troisième Série du Le Maître Phonétique)* 30. 24–27.

赵元任 [Chao, Yuen-Ren]. 1980. 一套标调的字母(英文). 方言 1980(2). 81–83.

Chelliah, Shobhana Lakshmi, Willem Joseph de Reuse. 2011. *Handbook of descriptive linguistic fieldwork*. Dordrecht, Netherlands; New York: Springer. doi:10.1007/978-90-481-9026-3

Chen, Yiya & Carlos Gussenhoven. 2015. Shanghai Chinese. *Journal of the International Phonetic Association* 45(3). 321–337. doi:10.1017/S0025100315000043

Cheung, Kwan-hin [張群顯]. 2016. Chao Tone Letters: Original theory Versus Current Practice. In 錢志安, 郭必之 and 鄒嘉彥, *Commemorative Essays for Professor Yuen Ren Chao: Father of Modern Chinese Linguistics* 現代漢語語言學之父 — 趙元任先生紀念論文集, 65–76. 臺北市 [Taipei City]: 文鶴出版有限公司 [Crane Publishing Company].

Figure 8: An XLingPaper bibliography demonstrating mixed Latin and Chinese scripts.

font allowing for additional glyph selection and context shaping. [22] [12] [46] Figure 11 illustrates the special font handling needed for the Awami Nastaliq font. Of the four output renderings, the only one that renders correctly is the XeLaTeX one that uses Graphite.[46]

5.3 Hyphenation for non-English languages

Since we use the `polyglossia` package, one can write an XLingPaper document in any of the sixty-one non-English languages listed in the `polyglossia` documentation and indicate the language code for this language in a document-wide attribute. XLingPaper passes this information to XeLaTeX which will hyphenate according to that language's hyphenation rules.

5.4 Author contact information

XLingPaper allows one to define sets of contact information for authors containing things like name, address, affiliation, email address, phone number, etc. With the XeLaTeX output, these author contact information boxes will wrap if there are more of them than will fit on one line on the page. The lines containing these boxes will also be justified. Neither the XHTML nor the RenderX outputs automatically wrap these boxes. They also do not justify them.

5.5 Vertical fill

For title page material, only the XeLaTeX output allows using vertical fill between items on a particular page of output. This can be useful for automatically inserting whatever vertical space is needed between, say, the last author's name and some publishing information that needs to appears at the bottom of the page. RenderX requires using overt, fixed vertical spacing values. For the XeLaTeX output, then, one does not need to manually adjust this vertical space for each document. One must do so for the RenderX output, however. This is a non-issue for the XHTML output because there are no page breaks like there are in PDF output.

5.6 Line numbering

When submitting an article for review, some publishers want the PDF to have continuous line numbers throughout the document. Only the XeLaTeX output does this.

[46] One must use XLingPaper's XeLaTeX package from TeXLive 2020 (`software.sil.org/xlingpaper/xelatex-package-from-tex-live-2020/`) for this particular font. The Graphite included in the 2010 version of XeLaTeX is not capable of rendering Awami Nastaliq well.

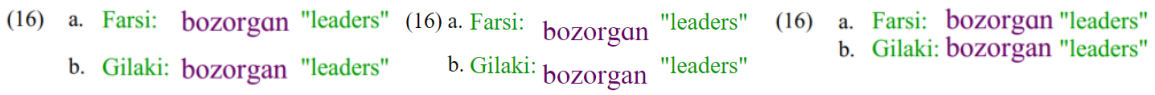


Figure 9: Ascender/descender font differences: The RenderX output is on the left; XHTML output is in the middle; the XeLaTeX output is on the right

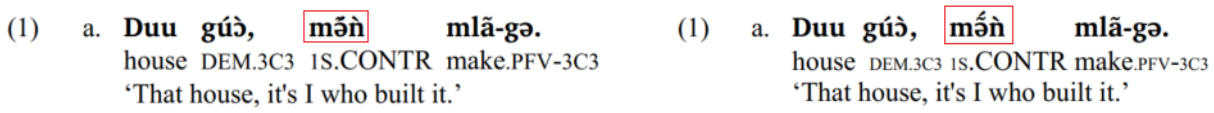


Figure 10: Stacked diacritics on the third word from the left: The RenderX output is on the left; the XeLaTeX output is on the right.

6 Features other outputs have that the LaTeX output does not

XeLaTeX does not allow for custom table cell padding and spacing. Having said that, the developer cannot remember any XLingPaper user ever asking for a way to do this for the XeLaTeX output. It just looks great.

Background color is not available for section titles.

Section 11.17.1.1 "Known limitations of using XeLaTeX" in the XLingPaper user documentation lists known problems.

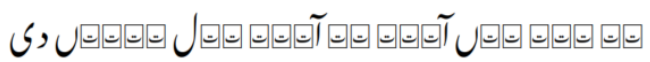
7 Conclusion

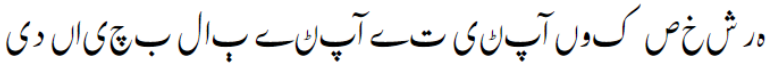
While the XLingPaper approach to composing documents via DTD-controlled user interface limitations has great value in and of itself, the fact that it can produce great looking output via XeLaTeX makes the learning curve rewarding. We feel that being able to produce PDF via XeLaTeX has made XLingPaper a fantastic tool for linguists.

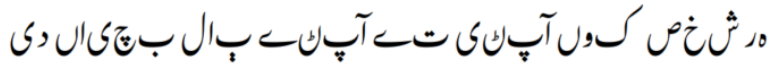
Additionally, XLingPaper serves as a model for other developers who are seeking a modular approach to creating custom publishing solutions. That is, one does not need to deploy the whole TeXLive system to create great looking outputs. Specific packages can be combined and redistributed to fit market needs.

A Hyphenation supported languages

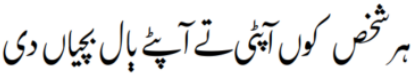
Language Name	Two letter code	Three letter code
Albanian	sq	sqi
Amharic	am	amh
Arabic	ar	ara
Asturian		ast
Basque	eu	eus
Bengali	bn	ben
Bulgarian	bg	bul
Catalan	ca	cat
Coptic		cop
Croatian	hr	hrv
Czech	cs	ces
Danish	da	dan
Dutch	nl	nld
English	en	eng
Esperanto	eo	epo
Estonian	et	est
Farsi	fa	fas
Finnish	fi	fin
French	fr	fra
Galician	gl	glg
German	de	deu
Greek	el	ell
Hebrew	he	heb
Hindi	hi	hin
Hungarian	hu	hun
Icelandic	is	isl
Indonesian	id	ind
Interlingua	ia	ina
Irish	ga	gle
Italian	it	ita
Lao	lo	lao
Latin	la	lat
Latvian	lv	lav
Lithuanian	lt	lit
Lower Sorbian		dsb
Malay	ms	msa
Malayalam	ml	mal
Marathi	mr	mar
Nynorsk	nn	nno
Occitan	oc	oci
Polish	pl	pol
Portuges	pt	por
Romanian	ro	ron
Russian	ru	rus

RenderX: 

XHTML: 

XeLaTeX: 

(not using Graphite)

XeLaTeX: 

(using Graphite)

Figure 11: Awami Nastaliq rendering. Only the XeLaTeX output using Graphite is correct.

Sanskrit	sa	san
Scottish	gd	gla
Serbian	sr	srp
Slovak	sk	slk
Slovenian	sl	slv
Spanish	es	spa
Swedish	sv	swe
Syriac		syr
Tamil	ta	tam
Telugu	te	tel
Thai	th	tha
Turkish	tr	tur
Turkmen	tk	tuk
Ukrainian	uk	ukr
Urdu	ur	urd
Upper Sorbian		hsb
Vietnamese	vi	vie
Welsh	cy	cym

References

[1] Bartholomew, Doris A, and Louise C Schoenhals. 2019. *Bilingual Dictionaries for Indigenous Languages*. Edited by Thomas L Willett. 2nd edn. Tlalpan, Ciudad de México, México: Instituto Lingüístico de Verano, A. C. [SIL International in Mexico]. sil.org/resources/archives/80401.

[2] Beadle, Jennie, and Matthew Lee. 2020a. *Paratext 9 Manual – in English*. SIL International. lingtran.net.

[3] Beadle, Jennie, and Matthew Lee. 2020b. *Paratext 9 Manual – in French*. SIL International. outilingua.net.

[4] Beier, Christine, and Lev Michael. 2022. *Managing Lexicography Data: A Practical, Principled Approach Using FLEx (FieldWorks Language Explorer)*. In The Open Handbook of Linguistic Data Management, edited by Andrea L. Berez-Kroeker, Bradley McDonnell, Eve Koller, and Lauren B. Collister, 301–14. Open Handbooks In Linguistics. Cambridge, Massachusetts: The MIT Press. doi.org/10.7551/mitpress/12200.003.0029.

[5] Bird, Steven, and Gary Simons. 2002. Seven Dimensions of Portability for Language Documentation and Description. In ISCA SALTMIL SIG: "Speech and Language Technology for Minority Languages," 23–30. Las Palmas, Canary Islands, Spain: ELRA. lrec-conf.org/proceedings/lrec2002/pdf/ws15.pdf#page=29.

[6] Bird, Steven, and Gary Simons. 2003. Seven Dimensions of Portability for Language Documentation and Description. *Language* 79 (3):557–82. jstor.org/stable/4489465.

[7] Black, Cheryl A., and H. Andrew Black. 2012. *Grammars for the People, by the People, Made Easier Using PAWS and XlingPaper*. In Electronic Grammaticography, edited by Sebastian Nordoff, 103–28. LD&C Special Publication 4. Honolulu, Hawai'i: University of Hawai'i Press. hdl.handle.net/10125/4532.

[8] Black, H. Andrew. 2009. Writing Linguistic Papers in the Third Wave. *SIL Forum for Language Fieldwork* 2009 (004): 11 pages. sil.org/resources/publications/entry/7790.

[9] Black, H Andrew. 2017. *Why Learn to Use XLingPaper*. Dallas, Texas: SIL International. software.sil.org/downloads/r/xlingpaper/resources/documentation/WhyUseXLingPaper.pdf.

[10] Brownie, John. 2013. Adverbs in the Mussau-Emira Verb Phrase. *Language & Linguistics in Melanesia* 31(1):1–11. langlxmelanesia.com/issues.

[11] Buck, Marjorie J. 2018. *Gramática del amuzgo Xochistlahuaca, Guerrero*. (Serie de gramáticas

de lenguas indígenas de México №16.) Tlalpan, Ciudad de México, México: Instituto Lingüístico de Verano, A.C. [SIL International in Mexico]. sil.org/resources/archives/75518.

[12] Correll, Sharon. 2000. *Graphite: An Extensible Rendering Engine for Complex Writing Systems.*. Paper presented at the 17th International Unicode Conference, San Jose, California. rabbits.continuation.org/w/images/7/73/Graphite_paper.pdf..

[13] Daniel, Marco, and Elke Schubert. 2013. *The mdframed Package: Auto-split Frame environment* version 1.9b.

[14] Di-Biase-Dyson, Camilla, Frank Kammerzell, and Daniel A Werning. 2009. Glossing Ancient Egyptian. Suggestions for Adapting the Leipzig Glossing Rules. *Lingua Aegyptia. Journal of Egyptian Language Studies* 17: 343–66. wwwuser.gwdg.de/~lingaeg/lingaeg17.htm.

[15] Dimitriadis, Alexis. 2016. *TeX/LaTeX Information*. Web page. ling.upenn.edu/advice/latex.html.

[16] Donnelly, Kevin. 2013. Representing Linguistic Pitch in XeLaTeX. *TUGboat* 34 (2): 223–27. tug.org/TUGboat/tb34-2/tb107donnelly.pdf.

[17] Ebarb, Kristopher J. 2014. *Tone and variation in Idakho and other Luhya varieties*. University of Indiana Ph.D. dissertation. pqdtopen.proquest.com/doc/1625743679.html?FMT=ABS.

[18] Frampton, John. 2006. *Pst-Asr: Tex Macros for Typesetting Autosegmental Representations*. Version:1.1. CTAN. bakoma-tex.com/doc/generic/pst-asr/pst-asr-doc.pdf.

[19] Frampton, John. 2012. *ExPex for Linguists: Example Formatting, Glosses, and Reference*. Version: 4.1. mathserver.neu.edu/~ling/tex/expex/base/doc/expex-doc.pdf.

[20] Freitag, Constantin and Antonio Machicao y Priemer. 2019. *LaTeX-Einführung Für Linguisten*. Berlin, Germany: Humboldt-Universität zu Berlin. doi.org/10.13140/RG.2.2.29299.27682.

[21] Holton, Gary. 2021. *Writing You Dissertation with LaTeX*. Typescript. Hawai'i. Github.com. gmholton.github.io/files/DissertationWriting.pdf.

[22] Kew, Jonathan. 2007. XeTeX Live. *TUGboat* 29 (1): 146–50. root.tug.org/tugboat/tb29-1/tb91kew.pdf.

[23] Kew, Jonathan and Stephen McConnel. 1990. *Formatting Interlinear Text*. Occasional Publications in Academic Computing, Number 17. Dallas, Texas: Summer Institute of Linguistics.

[24] Kircher, Athanasius. 1652. *Œdipus Ægyptiacus, hoc est Vniuersalis Hieroglyphicæ Veterum Doctrinæ temporum iniuria abolitæ Instauratio*. Opus ex omni Orientalium doctrina & sapientia conditum, nec non viginti diuersarum linguarum authoritate stabilitum, Romæ: Ex Typographia Vitalis Mascardi.

[25] Knuth, Donald Ervin. 1984. *The TeXbook*. A. Computers & typesetting. Reading, Massachusetts: American Mathematical Society; Addison-Wesley.

[26] Knuth, Donald Ervin. 1986. *TeX: The Program*. B. Computers & typesetting. Reading, Massachusetts: Addison-Wesley.

[27] Lamicela, Andrew Charles. 2020. *Distinguishing Passive from MP2-marked Middle in Koine Greek*. University of North Dakota M.A. thesis. commons.und.edu/theses/3277.

[28] Lehmann, Christian. 2004. *Interlinear morphemic glossing*. In Morphologie: Ein internationales Handbuch zur Flexion und Wortbildung Morphology: an international handbook on inflection and word-formation, edited by Geert E Booij, Christian Lehmann, Joachim Mugdan, and Stavros Skopeteas, 2:1834–57. Handbücher zur Sprach- und Kommunikationswissenschaft Handbooks of Linguistics and communication science 17. Berlin ; New York: Walter de Gruyter. doi.org/10.1515/9783110172782.2.20.1834.

[29] Liter, Adam. 2017. *LATEX Workshop (for Linguists)*. adamliter.org/content/LaTeX/latex-workshop-for-linguists.pdf.

[30] Lovell, Douglas. 1999. TeXML: Typesetting XML with TeX. *TUGboat*. 20 (3): 176–183. tug.org/TUGboat/tb20-3/tb64love.pdf.

[31] Marlett, Stephen A, (compiler). 2012. La Frase Nominal. In Stephen A. Marlett (ed.) *Los Archivos Lingüísticos Me'phaa*. Instituto Lingüístico de Verano, A.C. [SIL International in Mexico]. mexico.sil.org/publications/i-wpindex/work_papers_-_mephaa_grammar_files.

[32] Marlett, Stephen A. 2019. *Phonology From the Ground Up: The Basics*. Dallas, Texas: SIL International. sil.org/resources/archives/79207.

[33] Marlett, Stephen A. 2019. Presentation of three short texts in Isthmus Zapotec. SIL-Mexico Electtronic Working Papers #25. Ciudad de México: Instituto Lingüístico de Verano, A.C.. https://www.sil.org/resources/archives/80964.

[34] Neri Méndez, Emilia and Stephen A. Marlett. 2011 (Nov). Presentación Analítica del Texto "Flor de Calabaza". In Stephen A. Marlett (ed.) *Los Archivos Lingüísticos Me'phaa* (versión preliminar). Instituto Lingüístico de Verano, A.C. [SIL International in Mexico]. mexico.sil.org/publications/i-wpindex/work_papers_-_mephaa_grammar_files.

[35] Nordhoff, Sebastian, and Stefan Müller. 2020. *Language Science Press Guidelines*. Berlin, Germany: Language Science Press. langsci.github.io/guidelines/latexguidelines/LangSci-guidelines.pdf.

[36] Paterson III, Hugh J. 2021. *Language Archive Records: Interoperability of Referencing Practices and Metadata Models*. University of North Dakota M.A. thesis. commons.und.edu/theses/3937.

[37] Paterson III, Hugh J. 2021. *On Rights Management in Anthropological and Linguistic Sound Collections*. *ARSC Journal* 52(3): 547–563. http://www.arsc-audio.org/journal.html.

[38] Pellard, Thomas. 2019. Automatic formatting of interlinear glosses with LaTeX. *Cipanglossia* cipanglo.hypotheses.org/1221.

[39] Peter, Steve. 2004. TEX and Linguistics. *TUGboat* 25 (1): 58–62. tug.org/TUGboat/tb25-1/peter.pdf.

[40] Machicao y Priemer, Antonio, and Constantin Freitag. 2019. *LaTeX-Einführung Für Linguisten*. Presentation at the MGK Workshop – SFB 1412, Berlin. linguistik.hu-berlin.de/de/staff/amyp/latex20sfb/07-l4l-math2-trees-handout.pdf.

[41] Rasmussen, Kent. 2018. *A Comparative Tone Analysis of Several Bantu D30 Languages (DR Congo)*. University of Texas Arlington Ph.D. dissertation. hdl.handle.net/10106/27483.

[42] Rastorgueva, V. S., A. A. Kerimova, A. K. Mamedzade, L. A. Pireiko, and D. I. Edel'man. 2012. *The Gilaki Language*. Edited by Ronald M. Lockwood. Acta Universitatis Upsaliensis; Studia Iranica Upsaliensia 19. Uppsala, Sweden: Acta Universitatis Upsaliensis. urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-182789.

[43] Rei, Fukui. 1996. TIPA: A System for Processing Phonetic Symbols in LATEX. *TUGboat* 17 (2): 102–14. tug.org/TUGboat/tb17-2/tb51rei.pdf.

[44] Salffner, Sophie, and Tim Gaved. 2014. Working with ELAN and FLEx Together: An ELAN-FLEx-ELAN Teaching Set. Electronic Manuscript. SOAS, London, England. soas.ac.uk/elar/helpsheets/file122785.pdf.

[45] Schenner, Mathias, and Sebastian Nordhoff. 2016. Extracting Interlinear Glossed Text from LATEX Documents. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, edited by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, et al., 4044–48. Portorož, Slovenia: European Language Resources Association (ELRA). aclweb.org/anthology/L16-1638.pdf.

[46] SIL International. 2012. Comparison of OpenType and Graphite shaping speeds in a Nastaliq context. (accessed: 27 April 2022) scripts.sil.org/cms/scripts/page.php?site_id=projects&item_id=graphite_otcompare.

[47] Smith, Zac, Todd Snider, and Mia Wiegand. 2016. *LaTeX and Linguistics - How to Make Your Research Pretty*. Presentation at the Cornell Linguistics Circle, Cornell, New York. conf.ling.cornell.edu/miawiegand/Latex_Slides.pdf.

[48] So Miyagawa, and Vincent W.J. van Gerven Oei. 2021. Building Web Corpus of Old Nubian with Interlinear Glossing as Digital Cultural Heritage for Modern-Day Nubians. In *The Proceedings of the 11th Conference of Japanese Association for Digital Humanities*, vol. 2021. 144–147. Tokyo: Historiographical Institute, The University of Tokyo. hi.u-tokyo.ac.jp/JADH/2021/Proceedings_JADH2021_rev0905.pdf.

[49] Thiele, Christina. 1995. TEX and Linguistics. *TUGboat* 16 (1): 42–44. tug.org/TUGboat/tb16-1/tb46ling.pdf.

[50] Thiele, Christina. 2007. *Christina Thiele Interview by Dave Walden for the TEX Users Group*. Transcript. tug.org/interviews/thiele.html.

[51] Unicode Consortium, ed. 1991. *The Unicode Standard: Worldwide Character Encoding*. Version 1.0. Reading, Massachusetts: Addison-Wesley. unicode.org/versions/Unicode1.0.0.

[52] Ward, Monica. 2002. Reusable XML Technologies and the Development of Language Learning Materials. *ReCALL* 14 (2): 285–94. doi.org/10.1017/S0958344002000629.

[53] Wilson, Peter. 2007. The Memoir Class. *TUGboat* 28 (2): 243–46. tug.org/TUGboat/tb28-2/tb89wilson.pdf.

[54] Wilson, Peter. 2021. *The Memoir Class for Configurable Typesetting: User Guide*. version

3.70. Normandy Park, WA: The Herries Press. texdoc.org/serve/memoir/0.

[55] Wood, Joyce Kathleen. 2012. *Valence-Increasing Strategies in Urim Syntax*. Graduate Institute of Applied Linguistics M.A. thesis. diu.edu/documents/theses/Wood_Joyce-thesis.pdf.

[56] Woods, Frances. In preparation. *Halbi Interlinear Texts: Everyday Village Life*. ms.

◇ H. Andrew Black
blackhandrew (at) gmail dot com

◇ Hugh J. Paterson III
i (at) hp3 dot me
http://hp3.me