Putting it all together — A blind person's perspective on developing a toolbox for document preparation

A. Jonathan R. Godfrey
Institute of Fundamental Sciences, Massey University
Palmerston North, New Zealand

Abstract

Blind people need access to information if we are to succeed in education and employment. Transformation of this information from print into accessible forms is all too often laborious, especially production of alternative formats such as braille and sound recordings. However, there are ways of making the processes more efficient. This paper is aimed at starting the discussion about making the process of turning raw print information into useful formats for blind people more efficient. Greater efficiency should lead to greater use of solutions and therefore an increased range of accessible information for blind people. Increased automation is seen as the way forward, and the tools used by the author (himself blind) are highlighted as part of the discussion.

Creation of information for the blind person, which includes mathematical and scientific material, by the sighted publisher should be as easy as it is for a blind person creating this type of document for the sighted world in which we live. The author demonstrates that it is possible to create a number of formats from a single source file by example. Gaps in the process are exposed, which provide potential research directions. Projects that show promise in improving the process of turning print into alternative formats are also identified. Some ideas of the past are remembered for the solutions they offered and may offer again in the future.

1 Introduction

This presentation aims to bring as much information together as possible about document preparation in order to assist in the efficient production of documents for everyone, including a blind audience. In all respects, the author believes that making the process of inclusion as efficient and practical as possible will improve the uptake of initiatives that will benefit blind people. The ideas presented are for the ultimate benefit of a blind creator of documents which include mathematical and scientific content, but should also help blind readers of this material.

The material for this presentation has for the most part been gathered through the experiences of the author and has therefore been a little haphazard in its accumulation. While it is far from comprehensive, it has nonetheless proven invaluable to the author's success as a blind person creating documents for the sighted world. Some of the practices described have created opportunities to do tasks efficiently, while others have made the near impossible task a reality. Developing a more comprehensive summary, including tools not used by this author, is an outcome that will benefit an even wider audience. This paper is therefore just the beginnings of a resource that needs further development.

2 Starting points and endpoints

The vast majority of printed text information available today starts in one of two forms; it is either generated by word-processing software such as Microsoft Word, Open Office and the like, or it has been created using a mark-up language such as HTML or LaTeX. Document creators have their preferences but

G. Sutcliffe, A. Voronkov (eds.): DEIMS 2012, volume 1, issue: 1, pp. 1-7

the major distinction is the way in which the two alternatives are operated. When working with word-processing software, the user controls the look and feel of the document as much as the content, but the document creator using a mark-up language creates the structure and the content and then relies on the initial settings to control the overall formatting of the material. In the 21st century, word-processing software has moved towards the mark-up language way of working, through the use of styles and templates. The majority of access issues for text and mathematical content in either type of document have for the most part been resolved. I base this judgment on the ability for a blind person to create mathematical expressions, and to be able to read them once they have been altered by a second (sighted or blind) author. Access to graphical material in any form remains work in progress.

The problem we face as blind people is that the starting point is not the usual mode for delivering the final product for public consumption. For example, journal articles can be downloaded in at least three different ways: plain HTML (often incomplete), a PDF file (accessible and not accessible forms), or a truly graphical form such as a TIF file. It is rare to find a publisher use an immediately accessible file format such as Microsoft Word. Often the plain HTML files are created without the graphs, tables or equations commonly found in mathematical and scientific disciplines. Accessible PDF files do exist but are often confused with the non-accessible scans of printed pages. There are ways of reading much of the text material found in them, but access to the graphical material (including equations) is not possible for the average blind student who has not availed themselves of software being developed by the Infty Project[1]. This project is laudable for its efforts in making mathematical material available to blind people and it is pleasing to note its progress. This solution, as with other optical character recognition software, can only provide a rendering of the original document that is hopefully as reliable for the blind person as it is for the sighted reader. These solutions offer a cure when I think there is a need for preventative medicine.

The question that I think needs to be asked is why publishers continue to produce documents in unusable forms when they could so easily produce multiple document formats to suit a wider audience. We know that HTML or XML solutions provide accessible alternatives to PDF documents. The issue is that not enough of them are being produced.

Irrespective of the original source, both HTML and PDF formats can be produced. It is a labour intensive activity for Microsoft Word users, but is able to be automated by LaTeX users as is demonstrated in this presentation.

Most LaTeX users are working towards the production of a PDF document, much of which is not accessible to the blind user, but with some effort a parallel process could be used to create fully accessible documents for the blind. In many instances, a simple HTML rendering of the document is all that is required. but once the need for mathematical equations is factored into consideration, we know that XML is to be preferred. An example of the parallel process being implemented includes the vast set of printed volumes of documentation that support SAS statistical software[2]. This encyclopedia-like resource can be purchased in print form or viewed as webpages. The content is the same, but the presentation is different. The HTML version of this document is created using a tool known as Plastex[3]. Plastex processes LaTeX source files using a set of scripts written in the Python programming language. A simpler alternative to Plastex, which is part of LaTeX, is TeX4ht[4].

3 Automation

It is the author's firm belief that the processes involved in the creation of alternative formats are limited by the lack of automation that currently exists. By the term 'automation', I mean the ability to complete a task or entire process without human intervention. For example, the task of turning raw LaTeX into a PDF document for public distribution can be done using the mouse clicks required to process the document

the necessary number of times to ensure labels and references are correct, the table of contents page works out correctly, and then that the index and bibliographic references are all done properly; or, a single batch file can be written that does all this for you and requires just one mouse action. The former process requires the opening of the software application that has the icons or menu items that link to specific tasks, while the second method just needs the user to browse to the folder where all the files reside (including the batch file).

The batch file approach has been possible since well before the advent of the Windows operating system and has not depended on any software. Under more recent versions of the Windows operating system, the batch files could be scheduled as a further element in the automation of the entire process.

The batch file mode of operating software is often thought to be a little outmoded, but with an increasing number of computer users operating in command-line mode operating systems instead of graphical user interfaces, I can see more and more people turning back to these efficient modes of operation. The Plastex project (see above) is an example of a command-line driven application. In the blind and low vision community, the command line mode of operation has its advantages over menu-driven or icon based modes of operation because graphical user interfaces are even less efficient for the blind user than they are for the sighted user.

At first, an automated process needs to be monitored to ensure results are what was expected, but over time greater understanding of the process will lead to only needing an occasional audit or quick check to make sure results continue to be produced in the way that was expected.

Automation comes in quite a few forms. Some software applications, such as OmniPage 17 Professional[5], will monitor a folder on a computer and act on any file put there. The outcome of this example of an automated process is to convert a PDF into another file type (often Microsoft Word) and put the outcome in another folder for the user to uplift at their leisure. This type of process is not new to blind people. We had a means of converting a PDF into a text file, (with the associated risks of course), by sending the file as an email attachment to an address monitored by a machine not a person. The converted text document would come back by email as an attachment. This service is somewhat redundant now that solutions exist that can be resident on the user's computer instead of relying on having access to the internet, but the OmniPage Professional software does perform this task if needed.

4 Reproducible research

In a document preparation context, reproducible research is the activity of re-creating the documentation once something has changed in the scope of the research. Often this is due to an update of the data, perhaps because new data is added, but it might also be an update required by enhancements to the research outcomes. There are many instances where government agencies publish documents that detail the latest trends or estimates for social or economic indicators for example. A reproducible research approach to publishing these documents would see the data and text processed in such a way that the latest figures are interwoven with the commentary, and any changes in the commentary are then made. In some people's way of thinking this is similar to taking the old version of the document and updating it but the copy-and-paste way of working is often tedious, especially if formatting of the material is required. The efficiency of using the reproducible research approach can be illustrated using an example. Imagine a document lists the results for a time series of various indicators and also includes suitable graphs of the results; the data alterations lead to minor changes in the length of tables rather than the layout of tables and the number of points on the graphs change without altering the text details in the graphs or its size and placement in the document. Now contrast this with the approach an average Microsoft Word user would take.

Various approaches exist for completing this task. The R statistical software [4] can interact with

LaTeX or a number of other formats for interweaving text and statistical output. R can be used to carry out the entire process from the text file containing the mixture of input commands and LaTeX commands all the way through to the creation of the PDF document for distribution using a method known as 'Sweave'[6]. The Sweave system [2] is most notably used in the creation of support documentation for R itself through numerous vignettes [3].

Sweave is not unique. A similar product exists for Matlab, called Matlab Report Generator, as well as two tools for many statistical applications. The first, that is still a work in progress, is called StatWeave[7]. The second is called Emacs Speaks Statistics[8], often abbreviated to ESS, is reliant on a more comprehensive system called Org-Mode[9]. Emacs and ESS/Org-Mode can work with all manner of mathematical and statistical applications and programming languages. As with Sweave, ESS and Org-Mode can interweave commands, output from those commands and any other text in one document. An Org-Mode document can be used to create end-user documents of many forms including PDF, HTML, XML, and even Microsoft Word, with or without the creation of a LaTeX document.

5 LaTeX is the link, not the starting point

Many academics assert that to be successful in mathematical or statistical subjects, an academic needs to be capable of working with LaTeX. More notable however, many experienced blind people who are working in or studying mathematics suggest that a blind student who is serious about studying mathematical subjects to a higher level also needs to be capable of writing documents using LaTeX.

For many users of LaTeX, it is the starting point of the document preparation process. There are many ways nowadays of using LaTeX without creating an entire document. Access to Microsoft Word documents including mathematical notation has been improved through the successful interaction of MathType[10] and LaTeX. Many mathematical software applications can export their output windows in LaTeX formatted text, including Maple[11] and Maxima[12] but this author can only vouch for the accessibility of Maxima.

An alternative to using MathType's ability to display LaTeX code for equations is to use a converter application to create a LaTeX source file, such as Word-to-Latex[13].

6 Not all PDFs are bad

For many years blind people have complained about the lack of accessibility of PDF documents. Aside from the obvious problems associated with those PDF files created as a scanned image (usually via a photocopier) there is often little 'wrong' with many PDF files created today. The PDF files created by Microsoft Word for example are usually quite readable using commercial screen readers such as JAWS[14]. In this context, 'readable' should be interpreted as a standard lower than perfect. Full access is not guaranteed in anything other than those PDF documents that have ordinary text and simple formatting; mathematical notation, alternative language fonts, and graphics of any kind remain barriers to accessibility.

Once solutions to the access issues for these objects within PDF documents are resolved, this document format should be as useful to the blind user as they are for the sighted. The opportunity to move around the PDF document using hyperlinks could give value. At present, blind people can convert PDF documents to other useful forms, but the benefits of working with a PDF document are generally lost in the conversion. For example, solutions such as pdf2txt[15] have provided a quick and dirty solution for the blind user wanting to get just the text out of a PDF document in a hurry. The speed of the admittedly limited access this solution offers makes it a valued tool, especially for large PDF documents which are slow to work with as a user of screen reading software.

LaTeX users have the option of creating the final PDF document via two pathways. Either using standard LaTeX which creates a DVI file, which is then processed into a post-script file and then into a PDF document; or, using PDFLaTeX directly which makes the PDF from the TEX source file. In either case, the resulting PDF file is an untagged document. Screen readers such as JAWS take some time to process the document before the blind person has access to the file's contents. In my experience, the best results from these untagged documents is obtained using the 'left to right top to bottom' option for the JAWS screen reader and documents prepared using LaTeX.

Ultimately, the only option available to screen reader users to gain access to the PDF documents that contain mathematical notation, foreign font symbols, and the like is by using the Infty Reader software as mentioned previously.

7 An example

The majority of comparative advantages in document preparation for the author (over his sighted colleagues) have come from being able to worry about the content without worrying about its formatting. To this end, LaTeX is a key element to the author's success as a practicing academic and statistician. LaTeX offers the blind author a comparatively easy way to create documents that include tables, figures, mathematical expressions, and characters and symbols from many fonts. The raw LaTeX is accessible, only being dependent on the user's choice of text editor and adaptive technology software.

Over the last three years I have been building a set of notes that help novice users of the R statistical software complete tasks they may need in an introductory statistics course. This set of notes now forms a volume that is distributed as a PDF document to students via a web download [1]. I have transferred the material into a series of webpages during 2011 for the benefit of my own students, and for the benefit of blind users worldwide. This means that I needed to create both HTML for immediate open distribution via:

`http://r-resources.massey.ac.nz/LURN/front.html`

and as a single archive for users wishing to have MathML functionality via:

`http://r-resources.massey.ac.nz/BlindR.exe`

Upon receipt of a request for additional material, or discovery of any typo, error, or opportunity for minor improvements, the raw text files with LaTeX code and R commands can be edited. Then the documentation is updated with the press of a single button. I can be confident that the various formats contain the same information, that is up to date and has functional R code for students to replicate at their leisure.

The complete process includes the following steps:

1. Write separate files for each chapter and an overall TEX file that brings them together in the correct order. This step is only required once. Note there are separate TEX files for creating the PDF and HTML versions but only one copy of the material for each chapter.

2. Sweave the raw text files, using R, to create the TEX files for each chapter.

3. Process the group of TEX files to create the PDF file. This includes processing the files with all standard LaTeXprocesses for the table of contents, bibliography and indexing.

4. Re-process the group of TEX files with the next header file to create the HTML files. As well as the processes used to create the PDF file in the previous step, this step has some additional commands for creating the HTML and is dependent on the TeX4ht package.

5. Re-process the set of TEX files for a third time, with the same requirements as in the previous step except for choosing XHTML instead of HTML output files. This collection of XHT files is then zipped up into a self-extracting archive using WinRAR[16].

6. Delete any files that do not need to be kept. The various processes described above create a number of log files that once reviewed can be discarded. This includes individual graphics files created by the statistical software as well as working files.

This process is carried out using the Windows operating system but is equally replicable under other operating systems. The process works well, but there is room for a number of improvements. For example, I found it quite difficult to alter the font used on the webpages to meet a request made for low-vision readers within the TEX4ht processing. A simple solution was forthcoming however and involved the addition of a single command to the CSS files that control the layout and formatting of the webpages. This is handled as an extra step in the above process.

8 Shortcomings, ToDo's, and challenges ahead

The list of tasks that could receive further attention is not short, and there are new tasks that will be added to the list because new technology will inspire new desires and create new opportunities.

Many of the tools used for creating my documents are based on open source or free software. The problem with such applications is the very real risk that the development of these projects grinds to a halt due to a wider lack of interest. The TEX4ht project has slowed in its development since the untimely death of the main contributor. The problems I find with this very useful application are both related to the graphics it creates. I have some anecdotal evidence that some images are slightly distorted (not too much of a problem for the blind reader), but there seems to be no scope for creating an ALT tag for the graphics in the HTML or XML versions of the document.

I have already mentioned that the PDF documents created by LATEX are untagged. This is at least, seen as a job to do by the community developing TEX and LATEX. If more of the documents created by LATEX users were made available in both PDF and HTML formats, this concern would be less relevant. The access a blind person has, either using a screen reader or a braille display to an HTML document is actually quite acceptable if all the necessary steps are taken to make it accessible. The same cannot be said of hard copy braille documents which attempt to replace the printed documents sighted people use.

Further development of two-dimensional tactile displays that can be used to view documents containing text, maths, and graphical material may be an important solution to improving access for the blind mathematician or scientist, but this will create extra demands on publishers to create their material in a usable form. This means successful conversion and interweaving of mathematical notation, graphics and standard text as would be the case for the printed form of the same material.

In this author's opinion, the task of creating good hardcopy braille documents that include text, mathematical material and graphics will continue to be put into the 'too hard' basket while the job is so dependent on the efforts of humans. Increased automation of the processes involved in this activity must contribute to the reduction of this barrier, as long as the results are reliable.

There is little efficiency to be expected in the production of sound recordings using human speech, but we should start to see greater use of synthetic speech recordings as this software improves. Newspaper and magazine articles are already recorded using synthetic speech through the National Federation of the Blind's[17] Newsline (as well as similar services offered by Vision Australia[18] and the Royal New Zealand Foundation of the Blind[19]) so we can expect there to be a need for the automation of conversion of mathematical and scientific material into sound using synthetic speech at some time.

Having mentioned both braille and sound solutions leads me to wonder how long it will be before we can create a full daisy title from a LaTeX file. It would need to include high quality synthesized speech reading the text and equations, access to all text information in braille, as well as a tactile representation of all images.

9 Conclusion

I have presented an example of a fully automated process for preparing a number of document formats from a single source. I hope this inspires discussion, which in turn could lead to further development of more automated processes.

I believe that publishers across all disciplines would expand their capacity to include creation of accessible documents for blind people if only they knew how easy it could be to do so. In closing I note that improved access for blind people would also create extra avenues for the sighted world to disseminate their scholarly and literary works.

Web resources noted in the text

[1]The Infty Project can be found at `http://www.inftyproject.org/en/`

[2]SAS is marketed by the SAS Institute, found at `http://www.sas.com/`

[3]Plastex, found at `http://plastex.sourceforge.net/`, is an open source project for converting LaTeX into HTML and other formats.

[4]More details about TeX4ht are available at `http://tug.org/applications/tex4ht/mn.html`

[5]OmniPage is marketed by Nuance, found at `www.nuance.com/`

[6]The Sweave homepage is at `http://www.stat.uni-muenchen.de/~leisch/Sweave/`

[7]StatWeave is available at `http://www.divms.uiowa.edu/~rlenth/StatWeave/`

[8]The Emacs Speaks Statistics homepage is at `http://ess.r-project.org/`

[9]The Org-Mode homepage is at `http://orgmode.org/`

[10]MathType is a product marketed by Design Science Inc., found at `http://www.dessci.com/en/`

[11]Maple is available at `http://www.maplesoft.com/`

[12]Maxima (and therefore wxMaxima) are available at `http://maxima.sourceforge.net/`

[13]Word-to-LaTeX products and on-line services are available at `http://www.wordtolatex.com/`

[14]JAWS is marketed by Freedom Scientific, found at `http://www.freedomscientific.com/`

[15]A free pdf2txt converter is available at `http://EmpowermentZone.com/p2tsetup.exe` although many other (mostly commercial) options do exist.

[16]WinRAR is available from `http://www.rarlab.com/`

[17]The NFB is found at `http://www.nfb.org/`

[18]Vision Australia's home page is at `http://www.visionaustralia.org.au/`

[19]The RNZFB's home page is `http://www.rnzfb.org.nz/`

References

[1] A. Jonathan R. Godfrey. *LURN: Let's Use R Now*. Institute of Fundamental Sciences, Massey University, Palmerston North, New Zealand, 2010. ISBN 978-0-473-17650-1, available from `http://r-resources.massey.ac.nz`.

[2] Friedrich Leisch. Sweave: Dynamic generation of statistical reports using literate data analysis. In Wolfgang Härdle and Bernd Rönz, editors, *Compstat 2002 — Proceedings in Computational Statistics*, pages 575–580. Physica Verlag, Heidelberg, 2002. ISBN 3-7908-1517-9.

[3] Friedrich Leisch. Sweave, part II: Package vignettes. *R News*, 3(2):21–24, October 2003.

[4] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.