## TeX and copyediting

SK Venkatesan and CV Rajagopal

### Abstract

Copyediting of a manuscript involves bringing consistency at many levels, with many kinds of local and non-local changes. The LaTeX macros of the proposed copyediting package offer an excellent way to create markup that can handle the types of changes made by a copyeditor in a consistent way.

The English language also has certain localization requirements that could be handled through language switches in the spirit of the Babel package. Localization can be achieved through macros such as \vara{color} that take care of variant spellings. We also propose a family of macros for other copyediting requirements such as parenthetical commas, serial commas, Latin abbreviations, firstly, secondly, thirdly usages, juxtaposing and the use of appropriate synonyms. This copyediting package will streamline the task of copyediting and bring a higher level of quality, visibility and control to the final output.

### 1  Introduction

There can be many a slip between the cup and the lip in the publishing process. The manuscript that arrives in a modern publisher's office, usually as a LaTeX or an MS Word file, gets transformed bit by bit into a common XML form and then it is typeset into its final PDF form. It is a bit like smelting and purifying iron from its raw form and pouring it into an XML mold, using a DTD as the sieve, to produce the final finished products. Copyediting is a crucial step in the process and is receiving increasing attention now, as copyediting changes are being clearly indicated in the proofing process to the author.

Copyediting involves a broad range of activity: the accurate conversion of the initial input to XML, ensuring consistency of usage within the manuscript, correcting basic language and grammar, applying the finer aspects of the publisher's style, and placing XML hooks to ensure finer typographic aspects are taken care. The XML keeps the link alive between the present print-led world and future worlds such as HTML5. Copyeditors and XML form the bridge between these two worlds. Although TeX can be misused in many ways to make life difficult for a copyeditor [2], we have come a long way from the earlier days when the technology was still underdeveloped [1]. LaTeX's own secret little macros and TeX4ht have also made it easier to form this bridge between the two worlds.

Just as in all professions, copyeditors come from a long lineage of tradition. Copyediting tries to filter out what are deemed imperfections and inconsistencies in the manuscript, and ensure that author–reader communication is improved. Each publisher has an in-house style guide that has been refined over many years and forms the basis for copyediting. Our experience with different publishers has shown that it is possible to design a generic set of TeX macros that can be used in the spirit of BibLaTeX macros.

It should be mentioned here that these macros are not designed to replace copyeditors but to make it easier for them to take care of mundane aspects of copyediting in a systematic way, so that they will be able to concentrate on improving what's crucial, the author–reader communication. Despite market trends, the role of copyediting has never been more important in the present world with varied rendering devices, with different aspect ratios and modern semantic capabilities.

### 2  Copyediting macros

Copyediting involves quite a broad spectrum of activity. At one end of the spectrum it improves semantic communication between the author and the reader. At the other end of the spectrum it reinforces certain stylistic and typographic conventions of the publisher. Semantic aspects are much beyond the capability of ordinary TeX macros, so it is at the latter end of the spectrum that most of this effort will be focussed.

We break down the copyediting process into various modular components:

1. Localization (`loc`)
2. Close-up, Hyphenation, and Spaced words (`chs`)
3. Latin abbreviations (`lat`)
4. Acronyms and Abbreviations (`abr`)
5. Itemization, nonlocal lists and labels (`itm`)
6. Parenthetical and serial commas (`pc`)
7. Non-local tokenization (`nlt`)
8. Genus-species identification (`gsp`)
9. Juxtaposing (`jxt`)
10. Synonyms (`syn`)

A macro \delins is used to indicate the copyediting text changes as in:

```
\delins[opt]{deleted text}{inserted text}
[comments about the change]
```

The option (`opt`) within the square brackets indicate the category to which this change belongs:

```
opt=loc,chs,lat,abr,itm,pc,nlt,gsp,jxt,syn
```

However, these macros do not expose the detailed information about the copyediting categories, so we have created specific macros for each of the copyediting categories.

## 3  Localization —
## British-American-Australian-Canadian

There are many sub-categories in British-American-Australian-Canadian variations:

■ DG (Am) vs. DGE (Au, Br, Ca). In American spelling, Acknowledgement, Judgement (et al.) lose the e to become Acknowledgment, Judgment.

■ Z (Am, Ca) vs. S (Br, Au). American and Canadian spelling prefers ize, while Australian and British use ise in words like apologize/apologise and authorize/authorise. However, the rule is different for yze/yse patterns, in words like analyze/analyse, where American prefers z and the rest use s.

■ S (Am, Ca) vs. C (Br, Au). In words like defense/defence, offense/offence, American and Canadian prefer s instead of c.

■ G (Am) vs. GUE (Br, Au, Ca). In words like dialog/dialogue, catalog/catalogue American prefers to drop the ue.

■ OR (Am) vs. OUR (Br, Au, Ca). In words like color/colour, favor/favour American omits the u.

■ ER (Am) vs. RE (Br, Au, Ca). In words like center/centre, caliber/calibre American prefers the er spelling.

■ L (Am) vs. LL (Br, Au, Ca). In words like canceled/cancelled, modeled/modelled American prefers the single l spelling while the rest prefer double l.

■ Others. Many other differences don't fall into any regular pattern like the above, and so can be handled only by a word list with their corresponding language mapping table.

We use one macro to care of all of this complexity: `\vara{color}`. The switch to a particular language spelling is done in the preamble, e.g.:

```
\usepackage[lang=uk]{copyediting}
```

Both `\vara{color}` and `\vara{colour}` produce the same output, in this case colour, so the author's original text need not be changed. The other options for language switch in this context are am,ca,au. The default language is the British spelling. In exceptional cases when one wants to force a particular use in a particular instance one can use `\vara*{analog}`, which will leave the input unchanged.

## 4  Close-up, hyphenation, and spaced words

Although American spellings use fewer hyphens, that modern preference for closed prefixes has exceptions:

1. if the root word is a proper noun or a number (post-Depression, pre-2001)
2. for double prefix (non-self-governing)
3. if the prefix precedes a proper open compound then en-dash is used (pre–Civil War)

4. if two instances of the letter i or the letter a are adjacent (anti-intellectual, extra-action), or other combinations of letters that could hamper reading (pro-labor)
5. for a double prefix (anti-antibody)
6. for a repeated prefix with implicit use (over- and understimulation)

We use the macro `\hyp{{anti}{body}}` to hyphenate a compound word. For a closed-up word we use `\closeup{{anti}-{body}}`. For compound words that occur as two separate words we use `\sword{{Civil} {War}}`. You might wonder what use are such complex verbose macros in a LaTeX file? They give visibility to the corrections the copyeditor makes and offer hooks to produce a global inventory of various changes while at the same time making it feasible to make switches on a global scale.

## 5  Latin abbreviations

Latin abbreviations such as:

| cf. | compare | et al. | and others |
|-----|---------|--------|------------|
| etc. | and so forth | e.g. | for example |
| i.e. | that is | NB | note |

are straightforward to handle with a macro, as in `\lat{et al.}`, where the stylistic aspect is taken care of by global switches:

```
\usepackage[lat=0,abbr=italic]{copyediting}
```

The default `lat=0` leaves the text as is, and `italic` defines the font used. Using option `lat=1` removes all the dots, and `lat=2` changes the text to its English equivalent shown above.

## 6  Acronyms and abbreviations

If the initial letter abbreviations are spoken together as a word, as in Acquired Immune Deficiency Syndrome (AIDS), the term "acronym" is used — but we will not make this distinction here and treat them as one and the same. A simple macro, `\ac{AIDS}`, is good enough and the default global switch ensures that it is expanded correctly the first time within parenthesis. The mapping between acronym and expansion is declared using:

```
\newacro{AIDS}
  {Acquired Immune Deficiency Syndrome}
```

Many standard acronyms are available by default from the package, and only new acronyms need to be added this way. This can be checked during compilation. To avoid expansion of trite acronyms the first time, one can use the starred form, `\ac*{UK}`.

## 7  Itemizations, nonlocal lists, labels

A list with only a few items may be written like this:

- Firstly, this is an endangered species;

- Secondly, humans find them delicious;
- Thirdly, they are only found on this island.

In this example we could have as well have used 'first', 'second', 'third', instead of '...ly', making that a global option. It is also possible that the items may be changed to use standard arabic numeral: 1, 2, 3, ... In order to make such changes possible with a simple switch, one can use macros:

```
\begin{eitem}
  \item this is a endangered species;
  \item humans find them delicious;
  \item they are only found on this island.
\end{eitem}
```

Given that we run LATEX at least three times, we can have an option to change the last item in the list to lastly, as a global switch:

```
\usepackage[eitem=0,last]{copyediting}
```

where `eitem=0` is the (default) option that causes firstly, secondly ... and `last` indicates that the last item should be lastly. With `eitem=1` set, ly drops out, and with `eitem=2,3,..`, a standard list (enumerated, bulleted, etc.) list is output.

## 8   Parenthetical and serial commas

Many long sentences are difficult to read and can be communicated better with parenthetical constructs or footnotes rather than commas. It would be nice to have switches that can make this change, as in:

> The enthusiastic young ducks flying in front of the group\pc{led by the sagacious older ones at the back, make a lot of noise and turbulence} which are used by older ones at the back to warm their heart and the wings.

which outputs to some variation of:

> The enthusiastic young ducks flying in front of the group, led by the sagacious older ones at the back, make a lot of noise and turbulence, which are used by older ones at the back to warm their heart and the wings.

Depending on the global switch `pc=0,1,2,3,4` we can choose a parenthetical comma, parenthesis, em-dash, footnote or sidenote.

## 9   Elist (Oxford comma)

For a list of items as in this sentence:

> Suddenly warblers, tits and wrens started singing in chorus ...

we change the source to:

```
Suddenly \elist{warblers,tits,wrens}
started singing in chorus...
```

which can then be transformed into:

> Suddenly warblers, tits, and wrens, started singing in chorus ...

This macro helps bring consistency across the document regarding the placement of comma before and in the last item and in ensuring proper white-space after the comma for each item. The comma before and is known as the Oxford comma and can be triggered by a global switch, `oxfordcomma`.

## 10   Non-local tokenization

In a sequence of minimization operation, in a typical newspaper article the copyeditor encounters:

> His Holiness, the Prince of Mangoistan, addressed a gathering of ordinary mangoes in the capital New Mango. The Prince of Mangoistan pointed out the serious threat of foreign insects in the country. He further pointed out ...

His Holiness the Prince of Mangoistan shrinks to The Prince of Mangoistan and then finally to He. This copyediting operation can be denoted using:

```
\definetoken{mango}
  {His Holiness, the Prince of Mangoistan}
  {The Prince of Mangoistan}
  {He}
```

at the first instance and then `\Token{mango}` at the later instances. The `\Token{mango}` macro can thus be useful simply to indicate to which nouns the important pronouns link in a paragraph. (However, not all pronouns in English language have corresponding original objects as in the case of 'It' in 'It is raining!'.)

## 11   Genus-species identification

The Genus species formatting is similar to Latin abbreviations in many ways but has its own conventions as well. The macro:

```
\gensp{E. coli}
```

italicizes all instances and expands the abbreviations at the first instance. Like the Latin abbreviation macro `\lat`, this macro also allows the embedding of new undefined genus species entities, as in:

```
\definegensp{E. coli}{Escherichia coli}
```

## 12   Juxtaposing

Consider the sentence:

> It is another politician that we can't trust in the White House.

This can be changed to:

> It is another politician \pull that we can't trust\push{ in the White House}.

SK Venkatesan and CV Rajagopal

The output of this being:

> It is another politician in the White House that we can't trust.

If this is the original and we want it the other way around, we can similarly write:

> It is another politician \push{in the White House} that we can't trust \pop.

Juxtaposing has wide application, especially for copyediting misplaced modifiers. We use three macros \pull, \push and \pop to achieve juxtaposing.

## 13   Synonyms

Sometimes we also need to use the appropriate synonyms (e.g., due to redundancy in the original), as in:

> The temperature of the water was raised to 80°C to see if some bacteria . . .

This could be changed to:

> The temperature of the water was \syn{raised}{increased} to 80°C to see if some bacteria . . .

producing the output:

> The temperature of the water was increased to 80°C to see if some bacteria . . .

The \syn macro can thus be used for fixing inappropriate usage with an appropriate equivalent.

## 14   Interactive proofing

The above set of macros bring a certain level of transparency and consistency to the copyediting process. Using additional macros, this also has the potential to convey further the key aspects of copyediting to the author using menus and dashboards, bringing an interactive aspect to the proofing process.

## 15   Conclusion

We have made an attempt at bringing together many copyediting aspects as LaTeX macros. This involves some amount of drastic simplification and abstraction that cannot suffice in all cases. The starred macros can be used in those exceptional cases where one needs to escape the global switch. The non-local linkages work just as in the case of bibliography links by multiple compilation of LaTeX that passes information through auxiliary files.

All this is only a small step towards the Himalayan task of climbing the semantic hill through LaTeX macros as envisaged by SenseTeX [5].

## 16   Related work

There are some CTAN packages which we would like mention in relationship to our copyediting macros. The abbrevs package [4] has an interesting set of macros but the spirit of the macros here is quite different, in that we would like to keep the original text of the author in some form or another, so that changes made can be shown in a transparent manner. The acronym package [3] is very close in concept to our acronym macros and has some additional interesting features that needs to be considered in our copyediting macros.

## Acknowledgements

## References

[1] P. Flynn (1993). TeX and SGML: A Recipe for Disaster? *TUGboat* **14**(3), 227–230. http://tug.org/TUGboat/tb14-3/tb40flynn.pdf

[2] E. Gregorio (2005). Horrors in LaTeX: How to misuse LaTeX and make a copy editor unhappy, *TUGboat* **26**(3), 273–279. http://tug.org/TUGboat/tb26-3/tb84gregorio.pdf

[3] Tobias Oetiker (2012). An Acronym Environment for LaTeX 2ε. http://ctan.org/pkg/acronym

[4] Matt Swift (2001). The abbrevs LaTeX package. http://ctan.org/pkg/abbrevs

[5] S.K. Venkatesan (2005). Moving from bytes to words to semantics. *TUGboat* **26**(2), 165–168. http://tug.org/TUGboat/tb26-2/venkat.pdf

⋄ SK Venkatesan
  TNQ Books and Journals Pvt Ltd,
  Dr Vikram Sarabhai Instronic
    Estate, Kottivakkam,
  Chennai 600041, India
  skvenkat (at) tnq dot co dot in

⋄ CV Rajagopal
  River Valley Technologies,
  JWRA 34, Jagathy,
  Trivandrum 695014, India
  cvr3 (at) river-valley dot org