OpTEXcount

Richard Hartmann

1 Functionality and Usage

OpTEXcount is a basic python utility that analyzes OpTEXsource code. It is inspired by already existing TEXcount for LATEX. The functionality is really lightweight and basic. It counts words and other elements of OpTEX document and sorts them out into individual categories. These categories are inspired by TEXcount.

1) **Regular text words count**
2) Header words count
3) Caption words count
4) Header count
5) Figures/Tables count
6) Inline formulae
7) Displayed math formulae

Regular text words are words that occur in the main text. Header words are introduced using these particular keywords: `\tit`, `\chap`, `\sec` and `\secc`. Caption words take into account captions of tables and figures, i.e., `\caption/t` and `\caption/f`, footnotes(`\fnote` or `\fnotetext`) and marginal notes(`\mnote`). Header count is the number of all headers in our document. Figures/Tables count counts all tables and figures which occur in the document. Inline formulae are introduced using `$...$` and displayed formulae are introduced using `$$...$$`.

If there is a verbatim text in some category, its words are counted too.

Alongside above mentioned statistics, all individual sections are printed separately with particular word counts.

To get this utility running, Python3 has to be installed. Then user can run this utility by command:

```
python3 optexcount.py optex-filename
```

User can also use `-verbose` specifier. The source code will be printed on the screen with colored words, so he can easily see, what is considered as text word, header word, caption word, keyword etc.

2 A Brief Implementation Description

OpTEXcount uses regular expressions to separate individual words, to identify keywords, brackets, math formulae etc. These preprocessed words are sequentially loaded into counter that decides how to treat with the word.

If some specific keyword occurs, that is not known by this utility, it is skipped. Problem may occur, if an unknown keyword has got some arguments, the utility could count these arguments as individual words. Therefore, most common keywords with arguments are listed in `src/keywords.py`, so the utility can load successfully these arguments. If there is some important keyword missing, feel free to modify this file, to add this keyword...

In case of some keyword, that is important for the utility to identify, for example `\sec` or `\fnote`, special handler routine is ran, to manage this particular keyword. The place, from which we process the word also determines the category of the word, in what color should be this word highlighted in `-verbose` mode etc.

3 Utility Limitations

The results are pretty precise when the most basic OpTEX constructions are used. As It has been already mentioned above, it is really basic and lightweight utility. When user uses some more advanced TEXconstructions, problems may occur, and the counts may be imprecise. It doesn't take into account macro expansion, self defined rules etc... When using this utility, a valid OpTEX source code should be passed. If the file contains some errors(for example unterminated inline verbatim), utility terminates the process and won't provide much information about the problem. This utility only works with one particular file so it doesn't count included files as well.

Beside above mentioned limitations of this utility, it could be very useful for the analysis of basic OpTEX documents(such as the exemplary document `tests/main_test.tex`).