Documentation for TeXworks autocompleter

Henrik Skov Midtiby

November 29, 2012

Abstract

TeXworks is a powerfull latex editor which can be extended through scripts written in javascript. This script is an attempt at adding autocompletion capability to the TeXworks editor. Currently it can complete references, citations, filenames, words present in the open files and items in a list of templates.

1 What can the script do?

The script can do context based completions in latex documents. By looking at open files (*.tex and *.bib) and contents of the local directory, the script can complete

- references, defined with label in open files
- citations, used in opened files and in open bibtex files
- filenames, based on directory content

All interaction with the script consists of activating it using its shortcut crtl + m. In a document with the content shown in figure 1, the script will generate the following completions. In the first line, the cursor location is marked with|, in the following lines the selected text is marked as this.

hiera|	h|	\ref{h|
hierarchy|	hierarchy	\ref{helloWorld
	hieroglyphs	\ref{herons
		\ref{helloWorld

If there is a single completion candidate, the candidate is inserted and you can continue writing. If multiple candidates exists, the first candidate is inserted with the last part of it selected; when the script is activated again the next completion candidate is inserted, this is continued until all candidates have been shown and the first is inserted again.

1.1 See the script in action

The following two videos demonstrates how the script can be used.

- Demonstration of context aware autocompletion for texworks
  http://www.youtube.com/watch?v=wTjyiik9xm0

```
\section{Hello world}
\label{helloWorld}

Some text about hierarchy and hieroglyphs.

\begin{align}
\label{herons}
A
& = \sqrt{s \cdot (s - a) \cdot (s - b) \cdot (s - c)}
\end{align}
```

Figure 1: Example document containing references and labels used in section 1.

- Autocompleter for TeXworks
  http://www.youtube.com/watch?v=gRSq_OSxPec

1.2 How to install the script

Download `autocomplete.js` and `autocompleteFunctions.js`[1] and place them in a subfolder of your TeXWorks scripts dir.

2 How does it work

2.1 State representation

The script relies on a state representation that contains information about what should be completed and the last suggestion presented to the user. What should be completed is in front of the cursor and the last suggestion is stored in the selected area. As an example take the following string

eqnPTesting

The representation is given as two parts *eqnP* and *Testing*. The last suggestion is found by combining the two parts of the representation, which gives *eqnPTesting*. From the first part (*eqnP*) can the list of completion candidates be determined and with the last part (*Testing*) it is possible to determine the next element to suggest.

2.2 Context recognition

2.2.1 Close environment

Open environments can be closed by placing the cursor first in the line where the environment should be closed. The script can handle nested environments by only closing previously unclosed environments.

Before	After
\begin{center}	\begin{center}
	\end{center}

[1] https://github.com/henrikmidtiby/TeXworks-scripts/tree/master/autocomplete

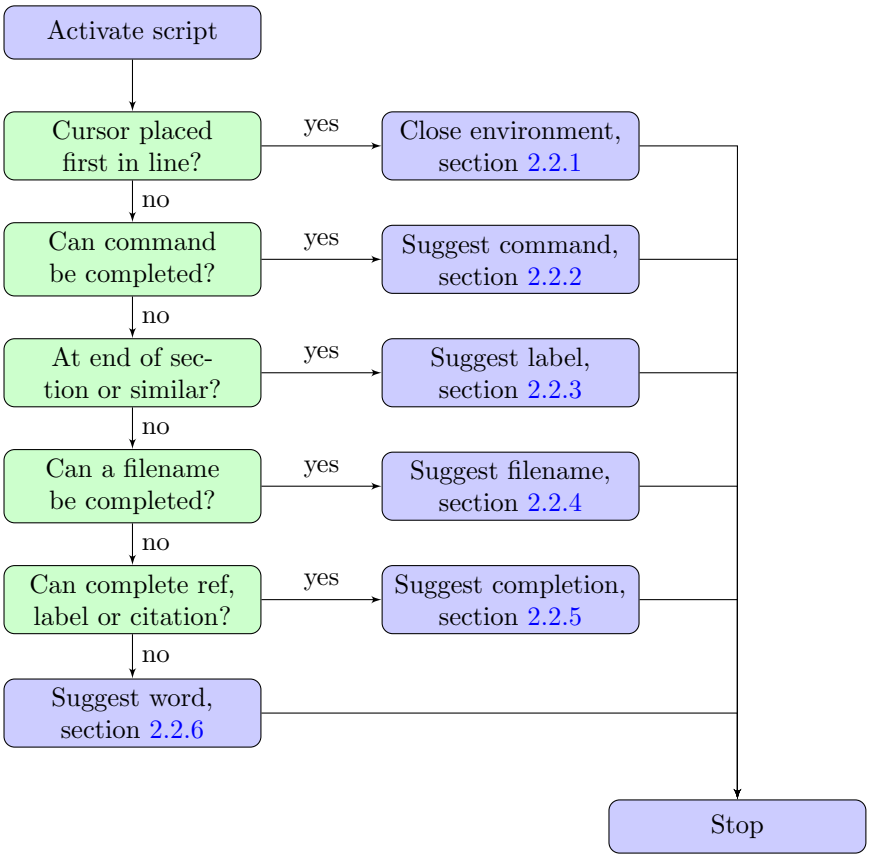

Figure 2: Flow chart.

Before
```
\begin{center}
\begin{itemize}
\item
```

After
```
\begin{center}
\begin{itemize}
\item
\end{itemize}
\end{center}
```

2.2.2 Suggest command or template

```
\include
\includegraphics
\includeonly
\includegraphics{filename}
```

2.2.3 Suggest label

Place the cursor at the end of a line containing a section, subsection or similar, activate the script and a label for the section / subsection will be suggested.

Before
```
\subsubsection{Suggest label}
```

After
```
\subsubsection{Suggest label}
\label{sssecSuggestLabel}
```

It also works inside figure and table environments after the caption command as shown below.

Before
```
\begin{figure}
\centering
\includegraphics[width=6cm]{}
\caption{Testing}
\end{figure}
```

After
```
\begin{figure}
\centering
\includegraphics[width=6cm]{}
\caption{Testing}
\label{figTesting}
\end{figure}
```

2.2.4 Suggest filename

Arguments to macros like include, input and includegraphics that accept filenames can be completed based on the files in the current directory structure. The following completion sequences can be observed in a directory structure as shown in figure 3.

```
\include{
\include{autocomplete.aux
\include{autocomplete.tex
```

```
\includegraphics{pic/
\includegraphics{pic/flowdiagram.tikz
```

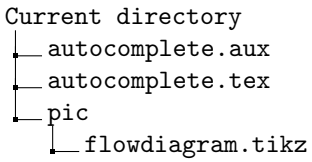

Figure 3: Example directory structure used in section 2.2.4.

2.2.5 Suggest ref, label or citation

References completed based on observed arguments to macros like label, ref, pageref and eqref in all files currently opened by TeXworks. A similar approach is used for completing arguments to label and cite commands. When completing cite commands the script also searches for candidates in open bibtex files.

\ref{eq
\ref{eqnMaxHeight
\ref{eqnSurf
\ref{eqnVol

2.2.6 Suggest word

2.3 Insert common text present in all candidates

3 Feedback and suggestions

If you have any feedback or suggestions about the script, feel free to contact me on `henrikmidtiby@gmail.com`.