bpj/vim-keymap/keymap/smartspace.vim as of Fri Mar 17 16:18:04 2017

```
" Maintainer:
               Benct Philip Jonsson <bpjonsson@gmail.com>
" Last Changed: 2001 Jan 1
" URL:
                <https://github.com/bpj/bpj-vim-keymaps>
" Someone asked on the vim use mailing list:
" > When we use Vim, what's the best way to add two spaces between
" > sentences automatically rather than manually press the space bar
" > twice? That's prevalent in the source code comment like the following:
" >
" >
      * comment comment.<space><more comment.</pre>
      * /
" >
" > Additionally by default, when we join two sentences with one on top of
" > the other, there would be two spaces between them after joining. What
" > do we need to do to achieve the similar effect with Vim automatically
" > when we are typing?
" I came up with a way to do this which actually works: a keymap and a
" couple of autocommands. (See :h mbyte-keymap and :h 'kmp').
" Don't use these when writing code! :-)
" The point of using a non-breaking space after abbreviations is so that
" you won't get linebreaks after abbreviations, which is considered bad
" style.
" The point of inserting a non-breaking space plus an ordinary space at
" the end of a sentence is so that the double space will be preserved when
" reformatting, but lines can break at the ordinary space anyway. You
" almost certainly want to define these autocommands (where '<nbsp>' and
" '<space>' mean that you should type '<C-K>NS' (digraph for non-breaking
" space) and an ordinary space respectively at those points):
      "Replace dot + nbsp + space with dot + space + space before writing Markdown files.
      :au BufWritePre *.md %s/\.<npsp><space>/.<space>/g
      " Replace dot + nbsp + newline with dot + newline before writing Markdown files.
      :au BufWritePre *.md %s/\.<npsp>$/./g
      " The reverse actions after writing a Markdown file.
      :au BufWritePost *.md %s/\.<space><space>/.<nbsp><space>/g
      :au BufWritePost *.md %s/\.$/.<nbsp>/g
```

```
" I use '*.md' in the examples because that fits my use case. You may want
" to substitute or add '*.txt' or something else.
" Note that the non-breaking spaces after abbreviations are preserved when
" writing the file. That is intentional!
" If you think autocommands are too much magic behind your back you can
" define them as ordinary commands instead and apply them manually:
      :com! NoSmartspace %s/\.<npsp><space>/.<space>/g | %s/\.<npsp>$/./g
      :com! Smartspace %s/\.<space><space>/.<nbsp><space>/g | %s/\.$/.<nbsp>/g
" short name for keymap
let b:keymap_name = "ssp"
highlight lCursor guibg=red guifg=NONE
scriptencoding utf-8
loadkeymap
" leave this line blank!
" Punctuation plus space
.<Space>
           .<char-0xa0><Space>
.'<Space> .'<char-0xa0><Space>
.'"<Space> .'"<char-0xa0><Space>
."'<Space> ."'<char-0xa0><Space>
."<Space> ."<char-0xa0><Space>
.')<Space> .')<char-0xa0><Space>
.'") < Space > .'") < char - 0 x a 0 > < Space >
.')"<Space> .')"<char-0xa0><Space>
."') < Space > ."') < char - 0 x a 0 > < Space >
.")'<Space> .")'<char-0xa0><Space>
.") < Space > .") < char - 0 x a 0 > < Space >
.)'<Space> .)'<char-0xa0><Space>
.)'"<Space> .)'"<char-0xa0><Space>
.)"'<Space> .)"'<char-0xa0><Space>
.)"<Space> .)"<char-0xa0><Space>
.)<Space> .)<char-0xa0><Space>
!<Space>
           !<char-0xa0><Space>
!'<Space> !'<char-0xa0><Space>
!'"<Space> !'"<char-0xa0><Space>
!"'<Space> !"'<char-0xa0><Space>
```

!"<Space>

!"<char-0xa0><Space>

```
!')<Space> !')<char-0xa0><Space>
!'") < Space > !'") < char - 0 x a 0 > < Space >
!')"<Space> !')"<char-0xa0><Space>
!"') < Space > !"') < char - 0 x a 0 > < Space >
!")'<Space> !")'<char-0xa0><Space>
!")<Space> !")<char-0xa0><Space>
!)'<Space> !)'<char-0xa0><Space>
!)'"<Space> !)'"<char-0xa0><Space>
!)"'<Space> !)"'<char-0xa0><Space>
!)"<Space> !)"<char-0xa0><Space>
!)<Space> !)<char-0xa0><Space>
          ?<char-0xa0><Space>
?<Space>
?'<Space> ?'<char-0xa0><Space>
?'"<Space> ?'"<char-0xa0><Space>
?"'<Space> ?"'<char-0xa0><Space>
?"<Space> ?"<char-0xa0><Space>
?') < Space > ?') < char - 0 x a 0 > < Space >
?'") < Space > ?'") < char - 0 x a 0 > < Space >
?')"<Space> ?')"<char-0xa0><Space>
?"') < Space > ?"') < char - 0 x a 0 > < Space >
?")'<Space> ?")'<char-0xa0><Space>
?")<Space> ?")<char-0xa0><Space>
?)'<Space> ?)'<char-0xa0><Space>
?)'"<Space> ?)'"<char-0xa0><Space>
?)"'<Space> ?)"'<char-0xa0><Space>
?)"<Space> ?)"<char-0xa0><Space>
?)<Space> ?)<char-0xa0><Space>
..<Space> .<char-0xa0> " dot plus non-breaking space
..<Space><Space> ..<Space> " dot plus ordinary space"
!!<Space>
          !<Space>
!!'<Space> !'<Space>
!!'"<Space> !'"<Space>
!!"'<Space> !"'<Space>
!!"<Space> !"<Space>
!!')<Space> !')<Space>
!!'") < Space > !'") < Space >
!!')"<Space> !')"<Space>
!!"')<Space>
                !"')<Space>
!!")'<Space>
              !")'<Space>
!!")<Space> !")<Space>
!!)'<Space> !)'<Space>
!!)'"<Space>
                !)'"<Space>
                !)"'<Space>
!!)"'<Space>
!!)"<Space> !)"<Space>
!!)<Space> !)<Space>
??<Space>
            ?<Space>
```

```
??'<Space> ?'<Space>
??''*<Space> ?''*<Space>
??''*<Space> ?''*<Space>
??''<Space> ?''*<Space>
??'')<Space> ?'')<Space>
??'')<Space> ?'')'<Space>
??'')'*<Space> ?'')'<Space>
??'')'<Space> ?'')'<Space>
??'')<Space> ?'')'<Space>
??'')<Space> ?'')'<Space>
??'','<Space> ?'','<Space>
??','<Space> ?'','<Space>
??','<<Space> ?'','<<Space>
??','<<Space> ?'','<<Space> ?'','<<Space>
??'','<<Space> ?'','<<Space> ?'','
```

" You will want to add more and/or those for your language

```
Mr.<Space> Mr.<char-0xa0>
Mrs.<Space> Mrs.<char-0xa0>
Dr.<Space> Dr.<char-0xa0>
e.g.<Space> e.g.<char-0xa0>
i.e.<Space> i.e.<char-0xa0>
viz.<Space> viz.<char-0xa0>
"more...
```

[&]quot; Abbreviations

[&]quot; Vim: set ft=vim noet ts=20 sts=20 list: