

I. FANCY PARSING OF LINKS FOR CITATIONS WITH PRE AND POST TEXT.

Some \LaTeX citations look like.[?] (post text) Here I explore parsing a link like.[?] (post text) Note you cannot use `[]` inside the link, as it breaks the link syntax. Also, these links must be wrapped in `[[[]]]` because of the parentheses and spaces in the parentheses. This is a very different approach than used here which used the description of the link to define the pre and post text. The disadvantage of that approach is that the key is hidden, whereas in this approach it is not; you can see the key and pre/post text.

The basic strategy will be to use a regexp to parse the link path. The regexp below is pretty hairy, but basically it looks for optional text in `[]` and uses numbered groups to store what is found. Then, we use what we found to construct the \LaTeX syntax. We redefine the function that gets the key for clicking, and we redefine the cite format function. The result is that we

```
1 (defun org-ref-parse-key (s)
2   "return pretext, posttext and bibtex key from a string like \"(pre text)(post text)bibtexkey\""
3   (string-match "\\(?:1:(\\(?:2:[^)]*\\))\\)?\\(?:3:(\\(?:4:[^]]*\\))\\)?\\(?:5:.*\\)" s)
4   ;; pretext posttext key
5   (list (match-string 2 s) (match-string 4 s) (match-string 5 s)))
6
7 (defun org-ref-get-bibtex-key-and-file (&optional key)
8   "returns the bibtex key and file that it is in. If no key is provided, get one under point"
9   (interactive)
10  (let ((org-ref-bibliography-files (org-ref-find-bibliography))
11        (file))
12    (unless key
13      ;; get the key
14      (setq key (nth 2 (org-ref-parse-key (org-ref-get-bibtex-key-under-cursor)))))
15    (setq file (catch 'result
16      (loop for file in org-ref-bibliography-files do
17        (if (org-ref-key-in-file-p key (file-truename file))
18          (throw 'result file))))))
19    (cons key file)))
20
21 (defun org-ref-format-cite (keyword desc format)
22   (cond
23     ((eq format 'latex)
24      (let* ((results (org-ref-parse-key keyword))
25             (pretext (nth 0 results))
26             (posttext (nth 1 results))
```

```

27         (key (nth 2 results)))
28     (concat "\\cite"
29         (when pretext (format "[%s]" pretext))
30         (when posttext (format "[%s]" posttext))
31         (format "{%s}" key))))))

```

org-ref-format-cite

```

1 (org-ref-format-cite "(pre text)(post text)key" nil 'latex)

```

\cite[pre text][post text]{key}

```

1 (org-ref-format-cite "(pre text)key" nil 'latex)

```

\cite[pre text]{key}

```

1 (org-ref-format-cite "key" nil 'latex)

```

\cite{key}

It looks like they all work! Let us test the links;¹¹ (pre text) and.² (post text) a multiple citation.^{1,3,4}

This seems to work from an export point of view. You can not mix multiple citations with this syntax.

¹ P. Mehta, P. A. Salvador, and J. R. Kitchin, ACS Applied Materials & Interfaces **0**, null (2014), <http://pubs.acs.org/doi/pdf/10.1021/am4059149>.

² N. Biškup, J. Salafranca, V. Mehta, M. P. Oxley, Y. Suzuki, S. J. Pennycook, S. T. Pantelides, and M. Varela, Phys. Rev. Lett. **112**, 087202 (2014).

³ R. L. Thompson, E. Albenze, W. Shi, D. Hopkinson, K. Damodaran, A. Lee, J. Kitchin, D. R. Luebke, and H. Nulwala, RSC Adv. **4**, 12748 (2014).

⁴ F. Calle-Vallejo, N. G. Inoglu, H.-Y. Su, J. I. Martinez, I. C. Man, M. T. M. Koper, J. R. Kitchin, and J. Rossmeisl, Chem. Sci. **4**, 1245 (2013).