# The `movement-arrows` package

Alan Munn

amunn@msu.edu

Version 1.0

May 21, 2023

**Abstract**

The `movement-arrows` package supplies simple support for drawing movement arrows on example sentences. It automatically adjusts spacing between examples or gloss lines to make room for the arrows. Arrows can also be annotated with labels. The package uses TikZ as a base, and various properties of the arrows can be adjusted using TikZ styles. The package has been tested with the `gb4e`, `linguex`, and `ExPex` example packages.

## 1 Package commands

| | |
|---|---|
| `\markwd[<name>]{<word>}` | marks a word or phrase for arrow placement; if word is a single word with no formatting, name will be set to word. If word is a phrase or contains formatting, a name must be given. |
| `\arrow[<options>]{<start>}{<end>}` | add an arrow between start and end nodes defined with `\markwd`. `\arrow*` draws an arrow above the words. |
| `\arrowheight` | length for the depth that the vertical line of the arrow drops |
| `\extraexheight` | length to add more space after a line with an arrow |

Table 1: Package commands

## 2 Basic usage

The way the package works is that you mark words in your example sentence using the `\mkword` macro and then use the `\arrow` macro to connect the words. So with the following code (using `gb4e` as our example package), we can get the result in (1). Note that you must compile the document twice to place the arrow correctly, because the package uses the TikZ `[remember picture]` function.

```
\begin{exe}
\ex  \mkword{Where} did this move from \mkword{t}?
\arrow{t}{Where}
\end{exe}
```

(1)   Where  did this move from t ?

This is the simplest use of the `\mkword` macro: it automatically creates a node with the same name as the word. This only works, however, if the word is a single word with no other formatting.

For example, if instead of representing traces with $t$ we use copies or a subscripted trace, we need to explicitly state the node name in the optional argument of `\mkword`. Here's a revised example:

```
\begin{exe}
\ex  \mkword[Where]{Where\textsubscript{i}} did
this move from \mkword[t]{t\textsubscript{i}}?
\arrow{t}{Where}
\end{exe}
```

(2)   Where$_i$  did this move from t$_i$ ?

## 2.1 Adding labels

The optional argument can be used to add a label to an arrow. There is one built-in label style called `circ`. Other styles can be created by the user if needed. The `circ` style is defined as follows:

```
\tikzset{
      circ/.style = {
              draw,circle,solid,
              node contents=#1,
              fill=white,
              inner xsep=.2em,
              inner ysep=0pt,
}}
```
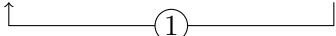
So to add a circled number "1" to the previous example we add `circ=1`:

```
\begin{exe}
\ex  \mkword[Where]{Where\textsubscript{i}} did
this move from \mkword[t]{t\textsubscript{i}}?
\arrow[circ=1]{t}{Where}
\end{exe}
```
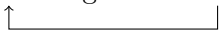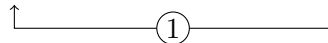
(3)   Where$_i$  did this move from t$_i$ ?
      ①

## 2.2 Glossed examples

The package automatically adjusts the vertical spacing between gloss elements and the first line to accommodate the arrows. Here's a glossed example. In this example, because we want the arrow to point in the middle of *een auto*, since it's a phrase that has moved, we need to manually adjust the spacing between *a* and *car* in the gloss line. Alternatively we could just use either *een* or *auto* as the marked node and then no adjustment would be needed. This code also makes use of the `\extraexheight` length to increase the spacing after the second arrow in the example set.

```
\begin{exe}
\ex
\begin{xlist}
\ex{\gll\ldots dat Jan \mkword[een]{een auto} gisteren \mkword{t}  gekregen
    heeft.\\
\ldots that John {a \hspace*{1em} car}  yesterday t gotten has \\
\ldots that John a car yesterday.}
\arrow{t}{een}
\addtolength{\extraexheight}{1ex}
\ex  \mkword{Where} did this move from \mkword{t}?
\arrow[circ=1]{t}{Where}
\ex Another example.
\end{xlist}
\end{exe}
```

(4)  a.  ...dat  Jan   een auto  gisteren   t  gekregen heeft.

...that John a    car   yesterday t  gotten    has
...that John a car yesterday.

b.  Where  did this move from t ?

c.  Another example.

## 2.3  Adding colour or changing the line style

The package uses TikZ styles to style the arrow and the label. The label style `circ` can be modified by using adding parameters directly. The arrow style can be modified by using `\tikzset{arrow/.append style={...}`. Here is some examples of both usages.

```
\begin{exe}
\tikzset{arrow/.append style={{red,dashed,very thick}}}
\ex  \mkword[Where]{Where\textsubscript{i}} did
this move from \mkword[t]{t\textsubscript{i}}?
\arrow[cyan,circ=1]{t}{Where}
\end{exe}
```

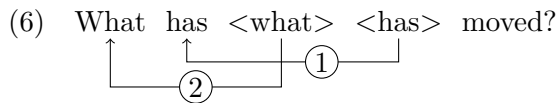(5)  Where$_i$  did this move from t$_i$ ?

## 2.4  Multiple arrows on one example

If we want to have more than arrow on a single example, it may make sense to have them offset from one another. The position of the labels may also need to be manually adjusted. This is quite easy to do. The length of the vertical line in the arrow is set by the length `\arrowheight`. The position of the label can be moved by adding a `xshift` value. Here's an example:

```
\begin{exe}
\ex \mkword{What} \mkword{has} \mkword[twh]{<what>} \mkword[thas]{<has>}
    moved?
\arrow[circ=1,xshift=1em]{thas}{has}
\setlength{\arrowheight}{4.5ex}
\arrow[circ=2]{twh}{What}
\end{exe}
```

(6)   What  has  &lt;what&gt;  &lt;has&gt;  moved?

For examples like this, some manual vertical spacing may be needed to prevent subsequent lines of text from overlapping the arrow.
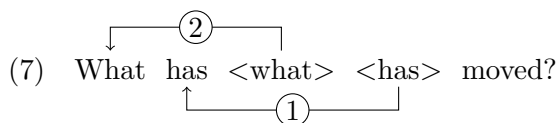
## 2.5  Arrows on top of words

The starred version of \arrow, \arrow* will draw the arrow above the words rather than the default position below the words. At the moment, no automatic spacing is done for arrows above, which means you will need to add vertical space yourself. I may consider adding automatic spacing here too, but at the moment this is not a priority, especially since at least in gb4e the \sn version of \ex will produce a blank line very simply. Here's the previous example using a combination of over- and under-arrows:

```
\begin{exe}
\sn
\ex \mkword{What} \mkword{has} \mkword[twh]{<what>} \mkword[thas]{<has>}
moved?
\arrow[circ=1]{thas}{has}
\arrow*[circ=2]{twh}{What}
\end{exe}
```

(7)   What  has  &lt;what&gt;  &lt;has&gt;  moved?

## 3  Version history

The first version of this package was written in 2017, but never released publicly. The use of the tikz-extra package paths.ortho allowed for the current code to be massively simplified. Bug reports and feature requests are welcome at the GitHub bug tracker.

## 4  Acknowledgements

Thanks to the TeX Stackexchange community, from whom I have learned many things, especially about TikZ. The earliest version of the package used code created by user Jake in response to a question I asked. This has now be superseded by the paths.ortho methods, but the style described there is quite useful in and of itself. I've also benefitted from discussions and suggestions with user percusse.