# The movement-arrows package

Alan Munn
amunn@msu.edu

Version 2.0
June 2, 2023

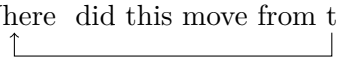
**Abstract**

The movement-arrows package supplies simple support for drawing arrows between elements in text or math. Its origin comes from linguistics, where drawing arrows between words is often used as an indication that the word or phrase has moved, hence the package name. Because of this origin, the package is aware of the main linguistics packages for example sentences and can be used with those packages, but does not require you to use of any of them. 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, covington and ExPex example packages.

## 1 Basic usage

The way the package works is that you mark words in text using the \mkword macro and then use the \mvarrow or \mvlink macro to connect the words. Here's a simple example:

```
\documentclass{article}
\usepackage{movement-arrows}
\mkword{Where} did this move from \mkword{t}?
\mvarrow{t}{Where}
```




## 2 Package commands

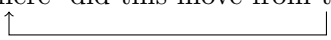
| | |
|---|---|
| \mkword[<name>]{<word>} | marks a word or phrase for arrow placement; if word is a single word with no formatting, name will be set to word. If word is a phrase or contains formatting, a name must be given. |
| \mvarrow[<options>]{<start>}{<end>} | add an arrow between start and end nodes defined with \mkword. \mvarrow* draws an arrow above the words. |
| \mvlink[<options>]{<start>}{<end>} | add a link (no arrows) between start and end nodes. \mvlink* draws an link above the words. |
| \setarrowstyle{<style>} | sets the arrow style; any regular TikZ arrow can be used. |
| \arrowheight | length for the depth that the vertical line of the arrow drops/raises. |
| \extraexheight | length to add more space after a line with an arrow. |
| \glarrowheight | length = sum of previous two lenghts |
| \arrowstrut | adds vertical space so that upwards links don't encroach on the previous line. |
| \arrowgloss | tells the package that an arrow appears in a glossed example. |

Table 1: Package commands

## 3 More examples

The way the package works is that you mark words in your example sentence using the `\mkword` macro and then use the `\mvarrow` 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.

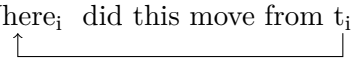
```
\begin{exe}
\ex  \mkword{Where} did this move from \mkword{t}?
\mvarrow{t}{Where}
\end{exe}
```

(1) 


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}}?
\mvarrow{t}{Where}
\end{exe}
```

(2) 

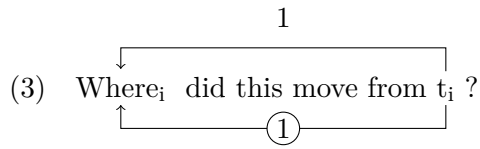

### 3.1 Adding labels

The optional argument can be used to add a label to an arrow. There are three built-in label styles: `circ` produces a circled label centred on the line itself. `above` and `below` place an uncircled label above or below the line respectively. 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`. To add an uncircled label above the arrow we can use `above=1`.

```
\begin{exe}
\ex  \mkword[Where]{Where\textsubscript{i}} did
this move from \mkword[t]{t\textsubscript{i}}?
\mvarrow[circ=1]{t}{Where}
\mvarrow*[above=1]{t}{Where}
\end{exe}
```




### 3.2 Glossed linguistic examples

Because of the way glossing macros work, arrows that are used in glossed examples using any of the standard linguistic glossing packages must be specially marked. We do this using the `\arrowgloss` command. This should appear before the glossed example that contains the arrow. 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.

For sample documents using each of the four linguistics example packages, please visit the GitHub repository.

```
\begin{exe}
\ex
\begin{xlist}
\ex{\arrowgloss
\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.}
\mvarrow{t}{een}
\ex  \mkword{Where} did this move from \mkword{t}?
\mvarrow[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 ?
1

c. Another example.

### 3.3 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{link/.append style={...}`. Here is some examples of both usages.

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

```
\end{exe}
```

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

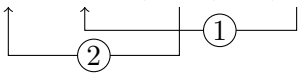

The arrow style itself can be changed using the `\setarrowstyle` command. The default arrow style is `->`; to use the `-latex` arrow style, you can use `\setarrowstyle{-latex}`. Any arrow style defined by TikZ is a useable style.

### 3.4 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?
\mvarrow[circ=1,xshift=1em]{thas}{has}
\setlength{\arrowheight}{4.5ex}
\mvarrow[circ=2]{twh}{What}
\end{exe}
```

(6) What has <what> <has> moved?



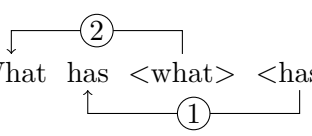

For examples like this, some manual vertical spacing may be needed to prevent subsequent lines of text from overlapping the arrow.

### 3.5 Arrows on top of words

The starred version of `\mvarrow`, `\mvarrow*` will draw the arrow above the words rather than the default position below the words. Here's the previous example using a combination of over- and under-arrows. The `\arrowstrut` command before the example add some vertical space so that the upwards arrow doesn't overlap the preceding text.

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

(7) What has <what> <has> moved?



### 3.6 Using with ExPex

The ExPex glossing macros work differently from gb4e, linguex, and covington, and so certain things need to be done slightly differently. Here's an example using ExPex:

```
\pex
\a\arrowgloss
\begingl[aboveglbskip=\glarrowheight]
\gla        \ldots dat Jan \mkword[een]{een auto} gisteren \mkword{t}
   gekregen heeft.//
\glb        \ldots that John {a\hspace*{1em} car}  yesterday t  gotten has
   //
\glft       \ldots that John got a car yesterday.
\mvarrow[circ=1]{t}{een}//
\endgl
```

There are two important things to note here. First, in the gloss that contains the arrow you need to set aboveglbskip=\glarrowheight to leave enough room for the arrow. Secondly, the \mvarrow command itself should appear *inside* the \glft line rather than outside of that line, i.e., it should appear before the // delimiter of the line.

### 3.7 Using with covington

When using the package with the covington glossing macros, you need to make sure that the \mvarrow command is *inside* the last line of gloss set:

```
\begin{subexamples}
        \item\arrowgloss\digloss
            {\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.\mvarrow{t}{een}}
\end{subexamples}
```

## 4 Links without arrows

All of the \mvarrow commands have an alternative version \mvlink which simply links the words with no arrows.

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

(8) What has <what> <has> moved?

## 5 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 and allowed version 1.0 to be released in May 2023, a little too fast. Version 2.0 changed the basic command name from `\arrow` to `\mvarrow` to avoid conflicts with other packages that define an `\arrow` command, and added the `\arrowgloss` and `\arrowstrut` macros for better spacing options. Bug reports and feature requests are welcome at the GitHub bug tracker.

## 6 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.