Then, the sentence is turned into a vector by simply concatenating these integers. For instance, if the sentence is "To be or not to be." and the indices of the words are as follows:

- "to": 5
- "be": 8
- "or": 21
- "not": 3

Then the sentence gets encoded as the vector `[5,8,21,3,5,8]`.

Loading the data

The data comes preloaded in Keras, which means we don't need to open or read any files manually. The command to load it is the following, which will actually split the words into training and testing sets and labels!:

```
from keras.datasets import imdb
(x_train, y_train), (x_test, y_test) = imdb.load_data(path="imdb.npz",
                                                      num_words=None,
                                                      skip_top=0,
                                                      maxlen=None,
                                                      seed=113,
                                                      start_char=1,
                                                      oov_char=2,
                                                      index_from=3)
```

The meanings of all of these arguments are here. But in a nutshell, the most important ones are:

- **num_words**: Top most frequent words to consider. This is useful if you don't want to consider very obscure words such as "Ultracrepidarian."
- **skip_top**: Top words to ignore. This is useful if you don't want to consider the most common words. For example, the word "the" would add no information to the review, so we can skip it by setting `skip_top` to 2 or higher.

Pre-processing the data

We first prepare the data by one-hot encoding it into (0,1)-vectors as follows: If, for