text_to_word_sequence

```
keras.preprocessing.text.text_to_word_sequence(text,
    filters=base_filter(), lower=True, split=" ")
```

Split a sentence into a list of words.

- **Return**: List of words (str).
- **Arguments**:
    - **text**: str.
    - **filters**: list (or concatenation) of characters to filter out, such as punctuation. Default: base_filter(), includes basic punctuation, tabs, and newlines.
    - **lower**: boolean. Whether to set the text to lowercase.
    - **split**: str. Separator for word splitting.

one_hot

```
keras.preprocessing.text.one_hot(text, n,
    filters=base_filter(), lower=True, split=" ")
```

One-hot encode a text into a list of word indexes in a vocabulary of size n.

- **Return**: List of integers in [1, n]. Each integer encodes a word (unicity non-guaranteed).
- **Arguments**: Same as `text_to_word_sequence` above.
    - **n**: int. Size of vocabulary.

Tokenizer

```
keras.preprocessing.text.Tokenizer(nb_words=None, filters=base_filter(),
    lower=True, split=" ")
```

Class for vectorizing texts, or/and turning texts into sequences (=list of word indexes, where the word of rank i in the dataset (starting at 1) has index i).

- **Arguments**: Same as `text_to_word_sequence` above.
    - **nb_words**: None or int. Maximum number of words to work with (if set, tokenization will be restricted to the top nb_words most common words in the dataset).
- **Methods**:

- **fit_on_texts(texts)**:
  - **Arguments**:
    - **texts**: list of texts to train on.
- **texts_to_sequences(texts)**
  - **Arguments**:
    - **texts**: list of texts to turn to sequences.
  - **Return**: list of sequences (one per text input).
- **texts_to_sequences_generator(texts)**: generator version of the above.
  - **Return**: yield one sequence per input text.
- **texts_to_matrix(texts)**:
  - **Return**: numpy array of shape `(len(texts), nb_words)`.
  - **Arguments**:
    - **texts**: list of texts to vectorize.
    - **mode**: one of "binary", "count", "tfidf", "freq" (default: "binary").
- **fit_on_sequences(sequences)**:
  - **Arguments**:
    - **sequences**: list of sequences to train on.
- **sequences_to_matrix(sequences)**:
  - **Return**: numpy array of shape `(len(sequences), nb_words)`.
  - **Arguments**:
    - **sequences**: list of sequences to vectorize.
    - **mode**: one of "binary", "count", "tfidf", "freq" (default: "binary").

- **Attributes**:
  - **word_counts**: dictionary mapping words (str) to the number of times they appeared on during fit. Only set after fit_on_texts was called.
  - **word_docs**: dictionary mapping words (str) to the number of documents/texts they appeared on during fit. Only set after fit_on_texts was called.
  - **word_index**: dictionary mapping words (str) to their rank/index (int). Only set after fit_on_texts was called.
  - **document_count**: int. Number of documents (texts/sequences) the tokenizer was trained on. Only set after fit_on_texts or fit_on_sequences was called.