We are choosing Keras tokenizer since it is a very reliable built-in tokenizer library.

Keras tokenizer allows to vectorize a text corpus, by turning each text into either a sequence of integers (each integer being the index of a token in a dictionary) or into a vector where the coefficient for each token could be binary, based on word count, based on tf-idf...

For hyper parameter, we use its default parameter which all punctuation is removed (filter regular expression: '!"#$%&()\*+,-./:;<=>?@[\\]^\_`{|}~\t\n'), turning the texts into space-separated sequences of words (words maybe include the ' character). These sequences are then split into lists of tokens. They will then be indexed or vectorized. 0 is a reserved index that won't be assigned to any word.

We use these Tokenizer function to convert our raw data set into useable input: *fit\_on\_text, word\_index, text\_to\_sequences*

Then we use **keras.preprocessing** library to truncate sequence with *pad\_sequence*

After the tokenizing process is completed, the trained tokenizer is saved using **pickle** library and ready to be reuse