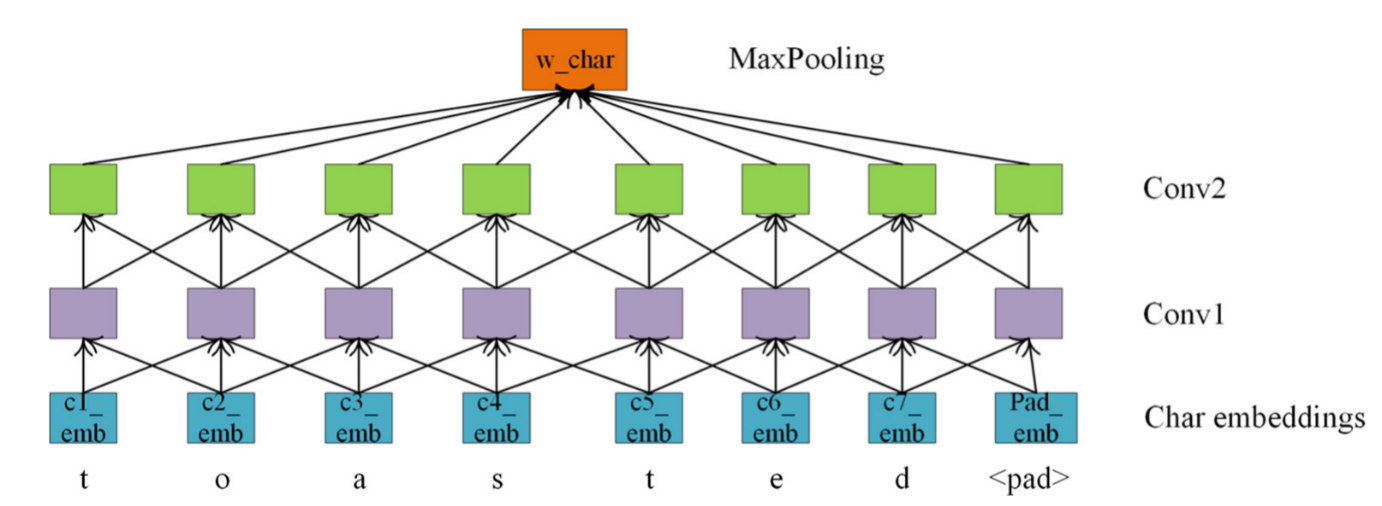
Convolutional Neural Network (CNN) are very powerful deep neural networks which uses various operations lke convolution, pooling etc with the help of padding, changing stride etc. Usually when we think of Convolutional Neural Network we think of them as something used in classification models in Computer Vision tasks to capture the locational invariance(not affected by the location of the object in the image) and compositionality(composing a local patch of low level features into high level representations) of an image. However in the field of NLP also CNN have proved to be of significant advantage. The CNN’s feature functions (i.e., the convolution filters) are applied to extract high-level features from adjacent words or n-grams regardless of their position, while taking local ordering patterns into account (Goldberg, 2017). The difference is just that instead of an image here the input will be in the form of a matrix which will either represent words or character features in the form of vectors. Another difference is that the filters used here will slide over full rows of the input matrix instead of sliding over some local patch.

We can use CNN in case of words as well as characters. One popular use of CNN’s in NLP is in the task of sentence classification (<https://arxiv.org/pdf/1510.03820.pdf>) where we represent the text as an array of vectors( each word mapped to a specific vector in a vector space composed of the entire vocabulary) that can be processed with the help of a CNN .When we are working with sequential data, like text, we work with one dimensional convolutions, but the idea and the application stays the same. We still want to pick up on patterns in the sequence which become more complex with each added convolutional layer.

Usually when we use bag of words or word2vec they only capture dependencies or structures at word level and cannot capture deep patterns/structures/dependencies. To overcome this problem we can use CNN at character level. And combine the word vectors generated with the CNN with the pre-trained word vectors trained on word level to combine the advantages of both.

<https://www.journals.elsevier.com/neurocomputing>

For example in this image we give the character vectors (pre-trained or randomized) as input along with the vectors for the padding (required as per the length of longest word) and then CNN is applied on these vectors and finally we use a max pooling layer to form the word vector for the same. This word vector generated has character level features in them and it can be combined with pre-trained word vectors to utilize the text features at maximum level.