NLP Day 6

A WEBINAR BY
DATA SCIENCE COMMUNITY SRM

Topics For This Session

VECTOR SPACE MODELS
WHAT ARE VECTOR SPACES AND WHY ARE THEY SO IMPORTANT ?

EMBEDDINGS
HUMAN INTUITION QUANTIFIED + HANDS ON

WORD2VEC EMBEDDINGS
MORE EMBEDDINGS + HANDS-ON

RNN BASED LSTMS AND CNN WITH NLP
MAKE THAT MODEL STRONGER WITH RNN BASED LSTMS AND CNN + HANDS-ON

COMPREHENSIVE ANALYSIS OF ALL MODELS AND FINAL CONCLUSION

Your Speakers

TARUSHI PATHAK

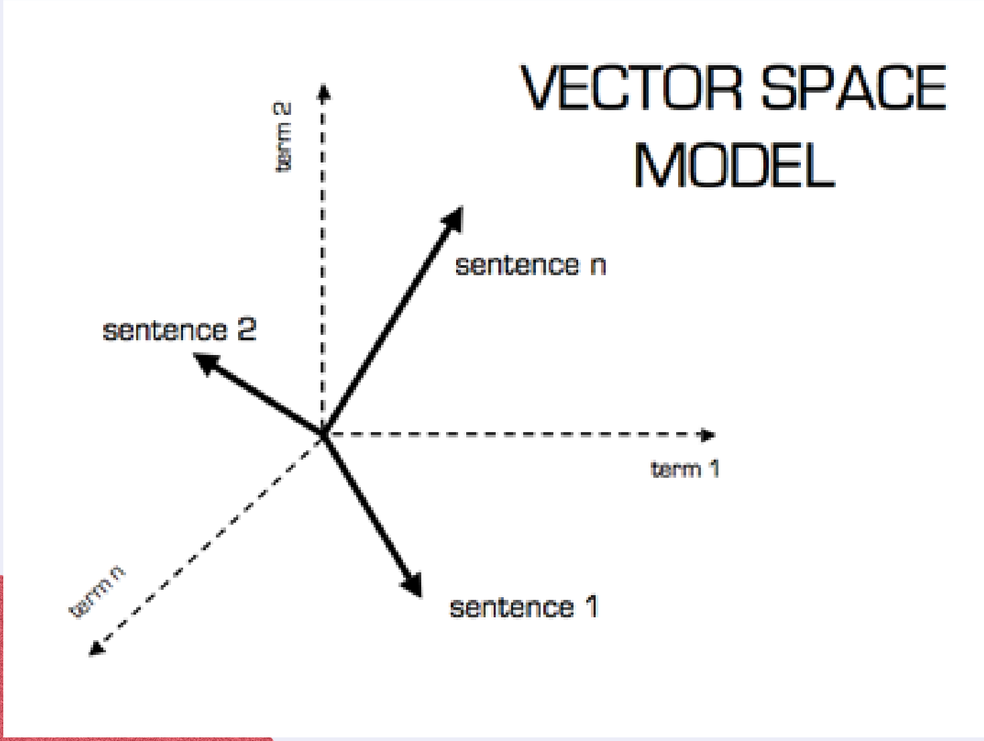
HARSH SHARMA

WARM-UP EXERCISE

- Load the Dataset
- Check the shape and data types present in the data.
- Make a list of steps that need to be performed in order to clean the text. (Hint: There are 8 things in total)

VECTOR SPACE MODELS: INTRODUCTION

Vector Space Models



The Vector-Space Model (VSM) for Information Retrieval represents documents and queries as vectors of weights.Each weight is a measure of the importance of an index term in a document or a query, respectively.

WHY ARE VECTOR SPACE MODELS IMPORTANT?

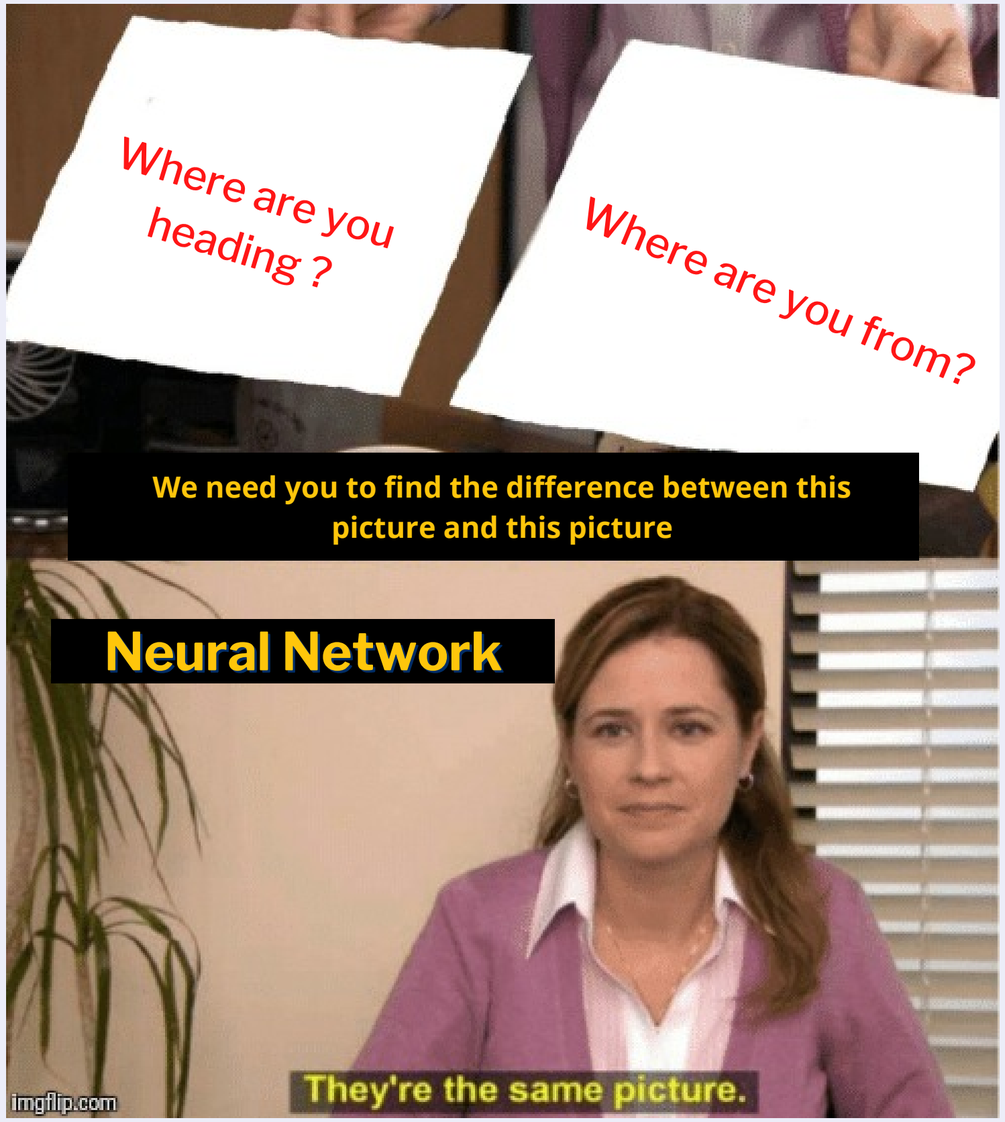
Consider the following two statements :

1) Where are you heading ?

2) Where are you from ?

Different Meaning

FOR A NEURAL NETWORK TRAINED ON ONE HOT ENCODED VECTORS



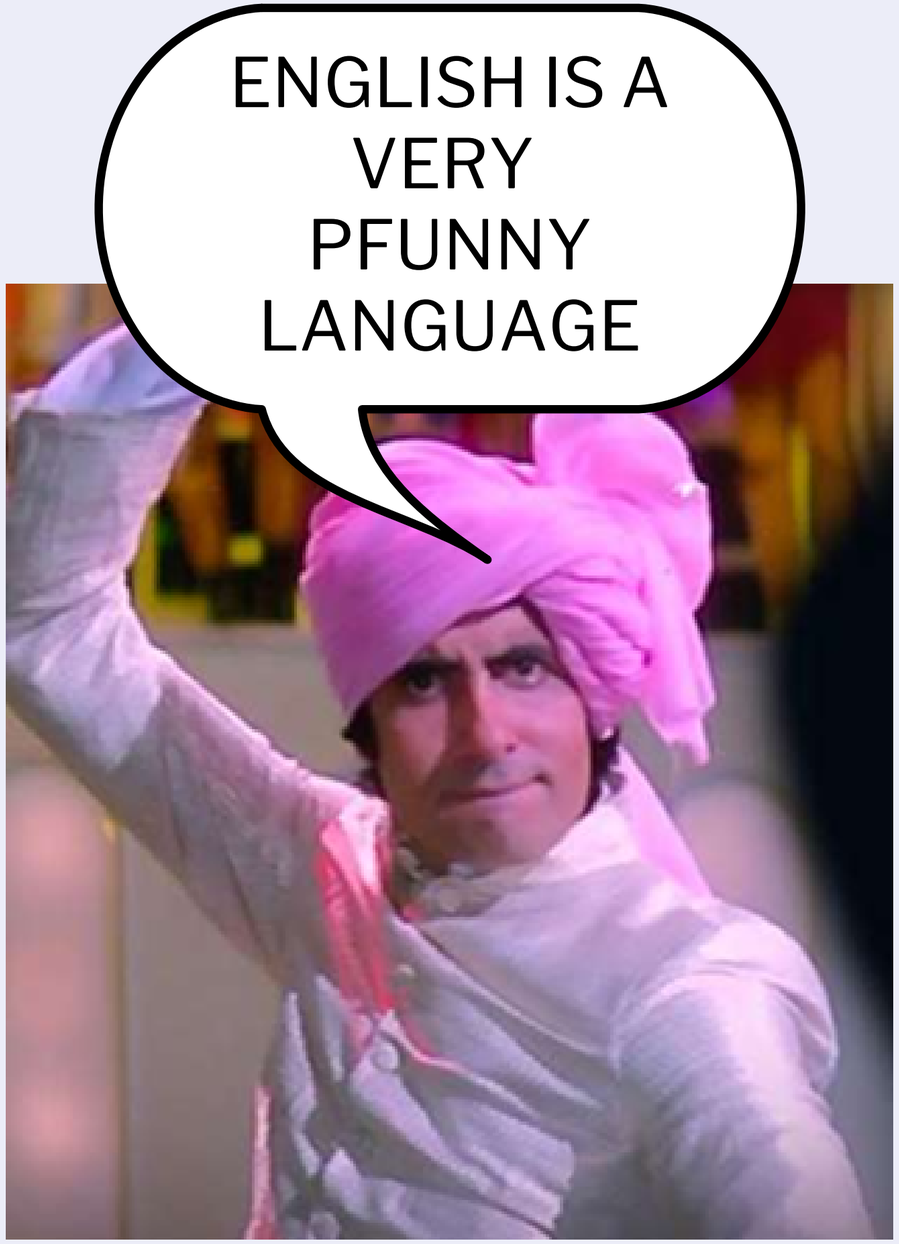
WHY ARE VECTOR SPACE MODELS IMPORTANT?

Consider the below statements :

1. What is your age ?
2. How old are you ?

Same Meaning

A model trained on one hot encodings will not be able to identify these relationships. Why ?

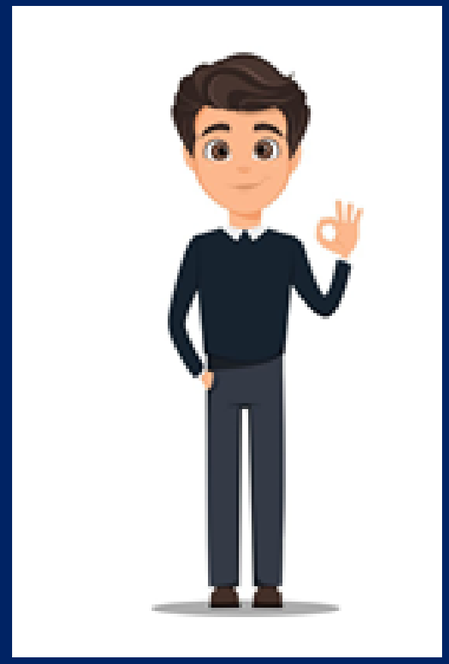


Or in other words, our languages are too complex for any machine in the first place , that understanding such semantic relations was going to be a great feat. This meant scientists needed a breakthrough and it came in the form of **Embeddings**

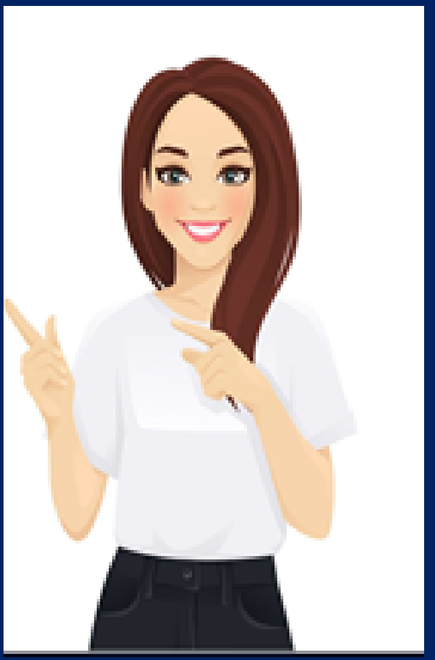
EMBEDDINGS

A word embedding is a learned representation for text where words that have the same meaning have a similar representation. It is this approach to representing words and documents that may be considered one of the key breakthroughs of deep learning on challenging natural language processing problems.

Here’s a word analogy for you to crack:



:



::

:

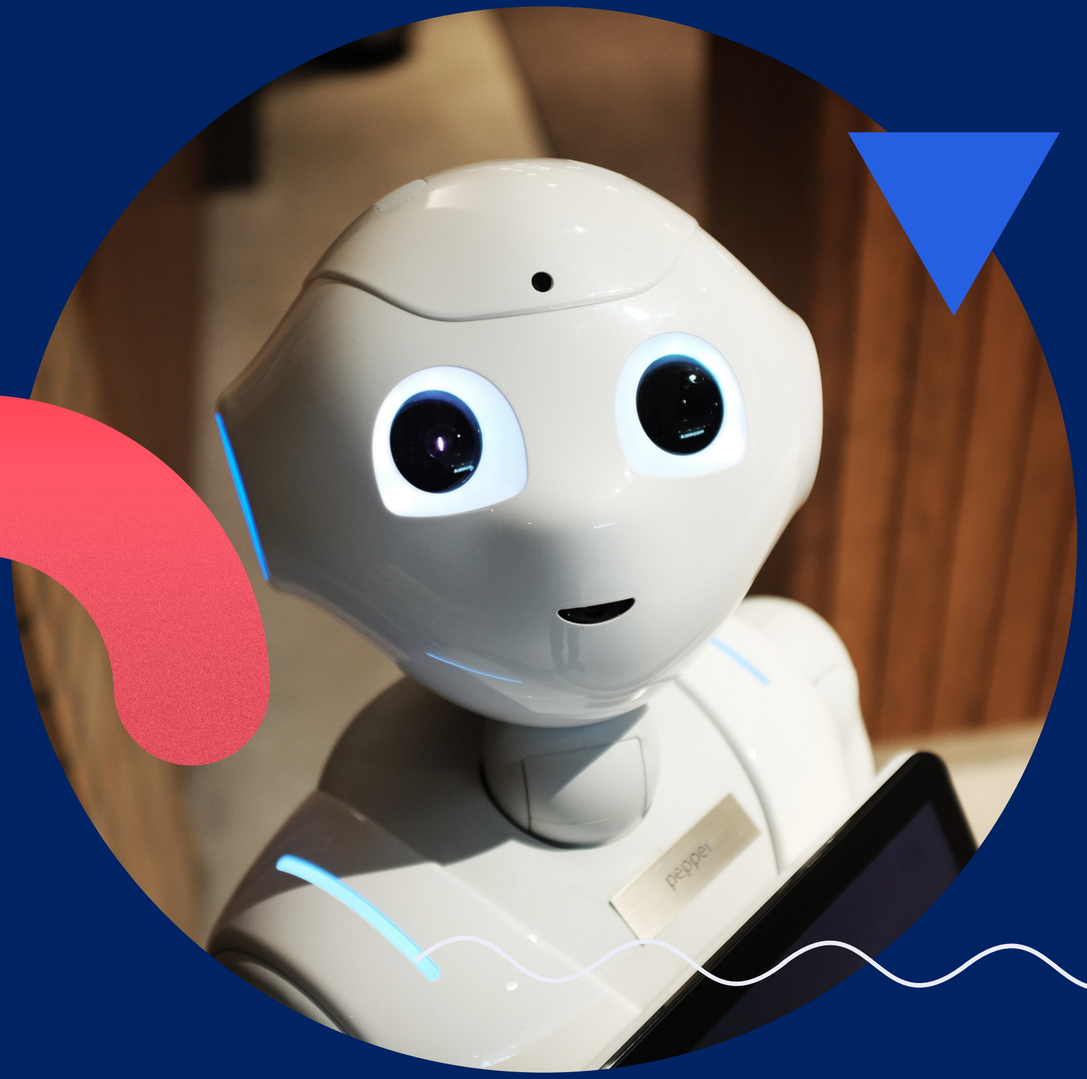
?

Yas Kween!

EMBEDDINGS- CONTINUED

Features	Man	Woman	King	Queen	Apple	Orange
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97
. . 300 features	--	---	---	---	---	---

STILL HAVING A TOUGH TIME GRASPING THE CONCEPT ? DON'T WORRY !
WE GOT YOU ☺



IT'S HANDS-ON TIME !

WORD2VEC EMBEDDINGS (SKIP GRAM MODEL)

Word2vec is a two-layer neural net that processes text by “vectorizing” words. Its input is a text corpus and its output is a set of vectors: feature vectors that represent words in that corpus. While Word2vec is not a deep neural network, it turns text into a numerical form that deep neural networks can understand.

Word2Vec comes in two flavours :

1. Skip Gram Model
2. Continuous Bag of Words Model

WORD2VEC EMBEDDINGS - SKIP GRAM MODEL

Consider the sentence :
I want a glass of orange juice to go along with my cereal.

Context	Target
Orange	Juice
Orange	Glass

Here the embedding attempts to learn the mapping between the context word and the target word. The target word could be ± 5 or ± 10 words near the context word.

Hence, if given a context word , the probability that the word within ± certain distance is the target word, is how the weights are designed or updated.

WORD2VEC EMBEDDINGS - MAKING OF EMBEDDING MATRIX EXPLAINED

Review [0] : The movie was really good.

The :	0.1	0.253	0.85	0.3	Goes on till 300 ..
movie:	0.2	0.4	0.5	0.6	Goes on ..

In this each word is represented by an embedding vector of dimension 300. Adding all those vectors would give us the vector representation as a sentence.

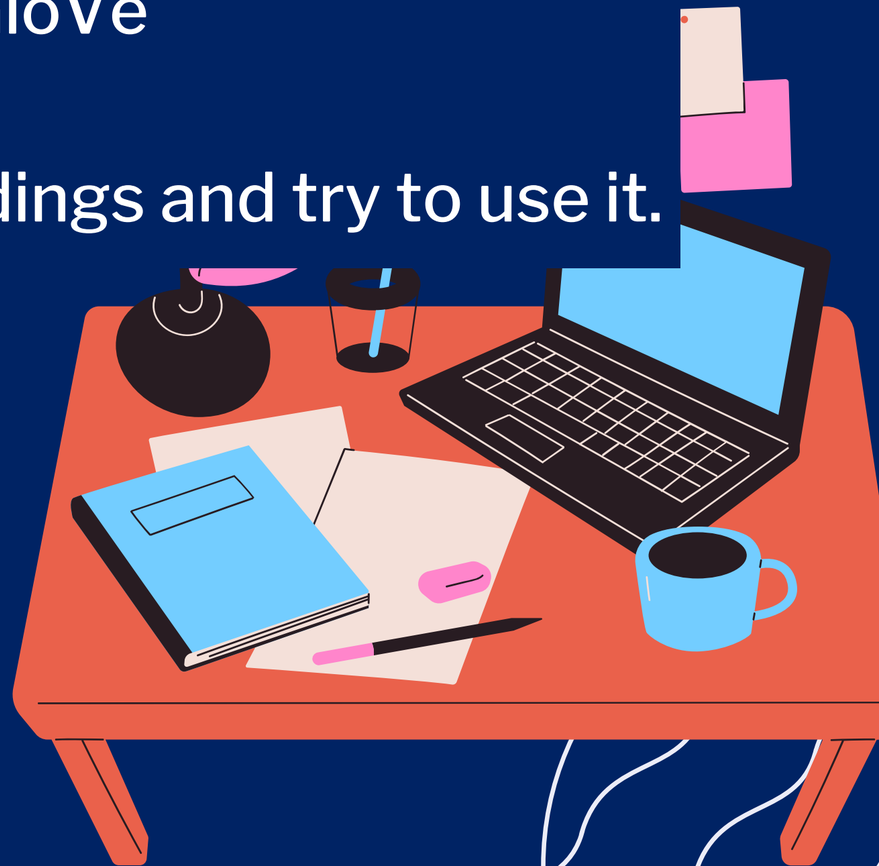
Review[0]:	0.8	0.55	0.235	0.9	Goes on ..

This review[0] is then divided by total number of words to normalize it and added to the embedding matrix. The embedding matrix at index 0 would hold the vector representation of review 0 and so on.

The major drawback of this model is the denominator function while calculating the probability, as it is computationally expensive.

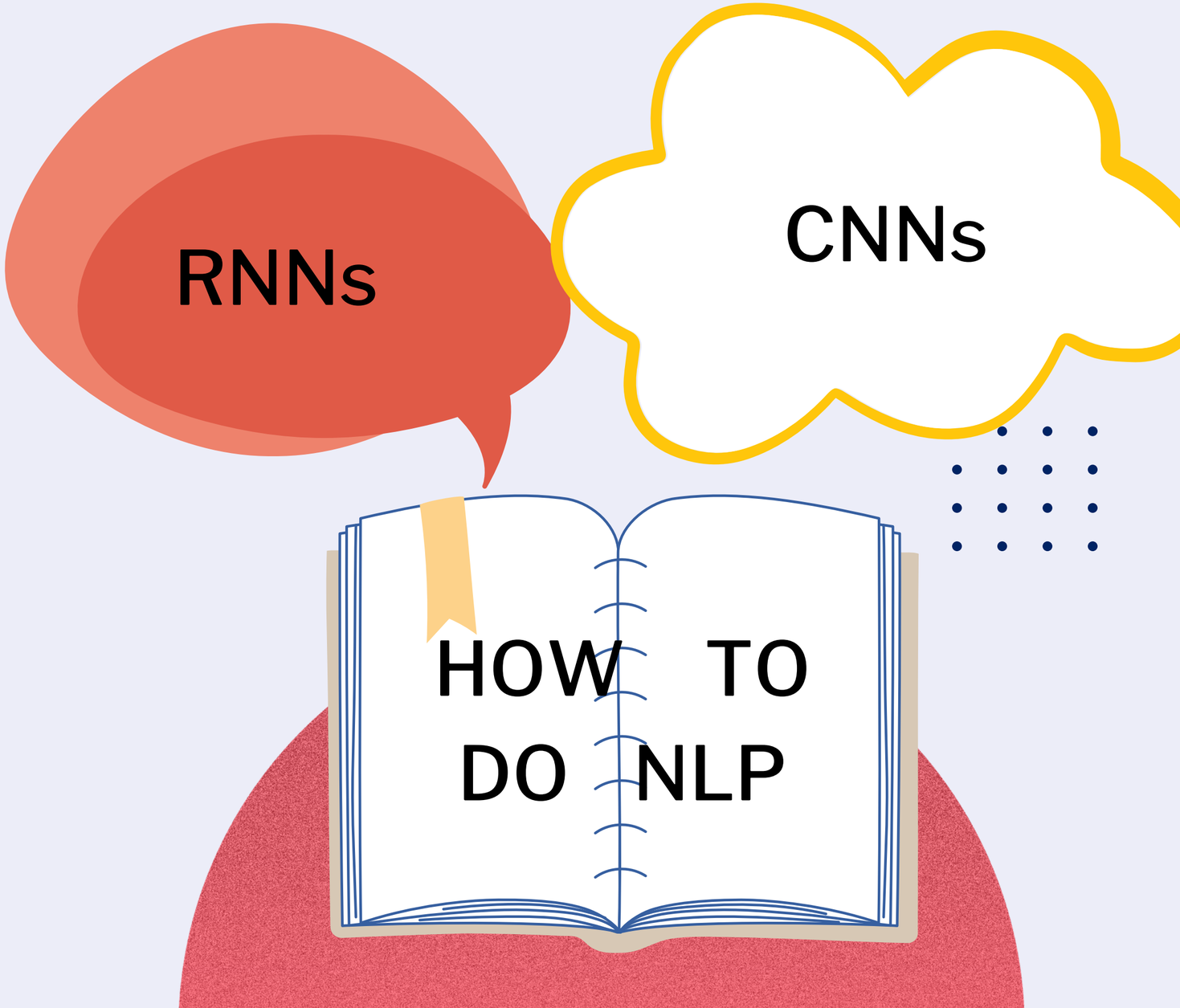
Homework Exercise

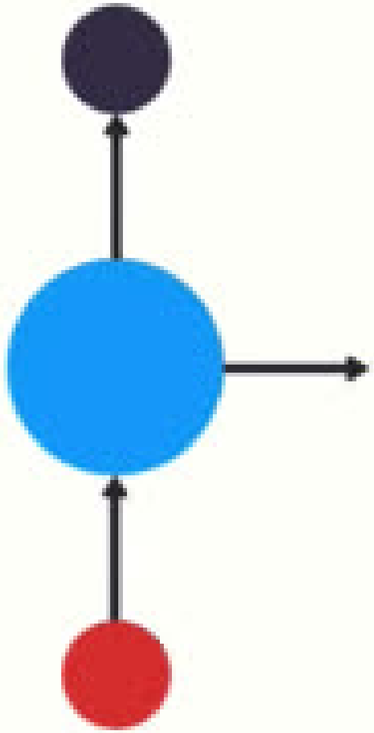
- Read about GloVe Embeddings
- Check out the code given in the Notebook for GloVe Embeddings and see if you can improve it.
- Check out Hugging Face Transformers Embeddings and try to use it.



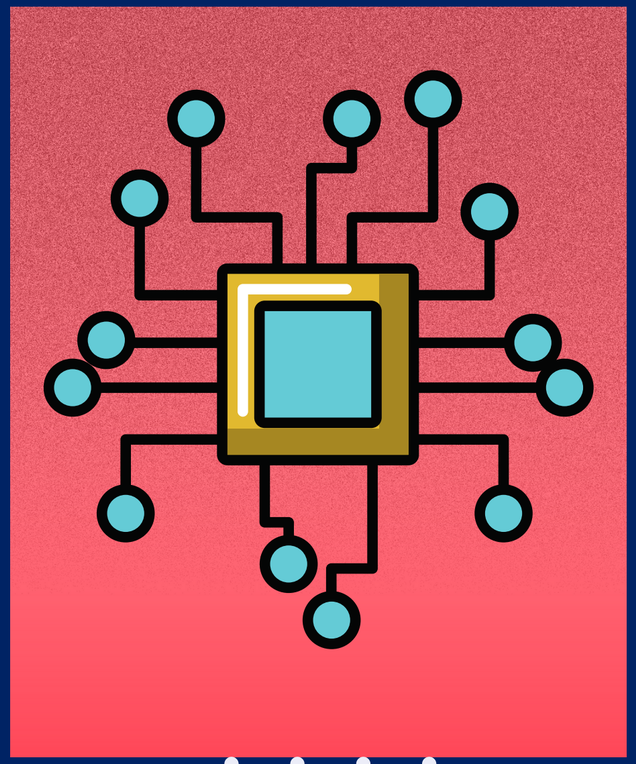

yahan lenge ek chota sa break

NLP with RNN and CNN





NLP WITH RNNS

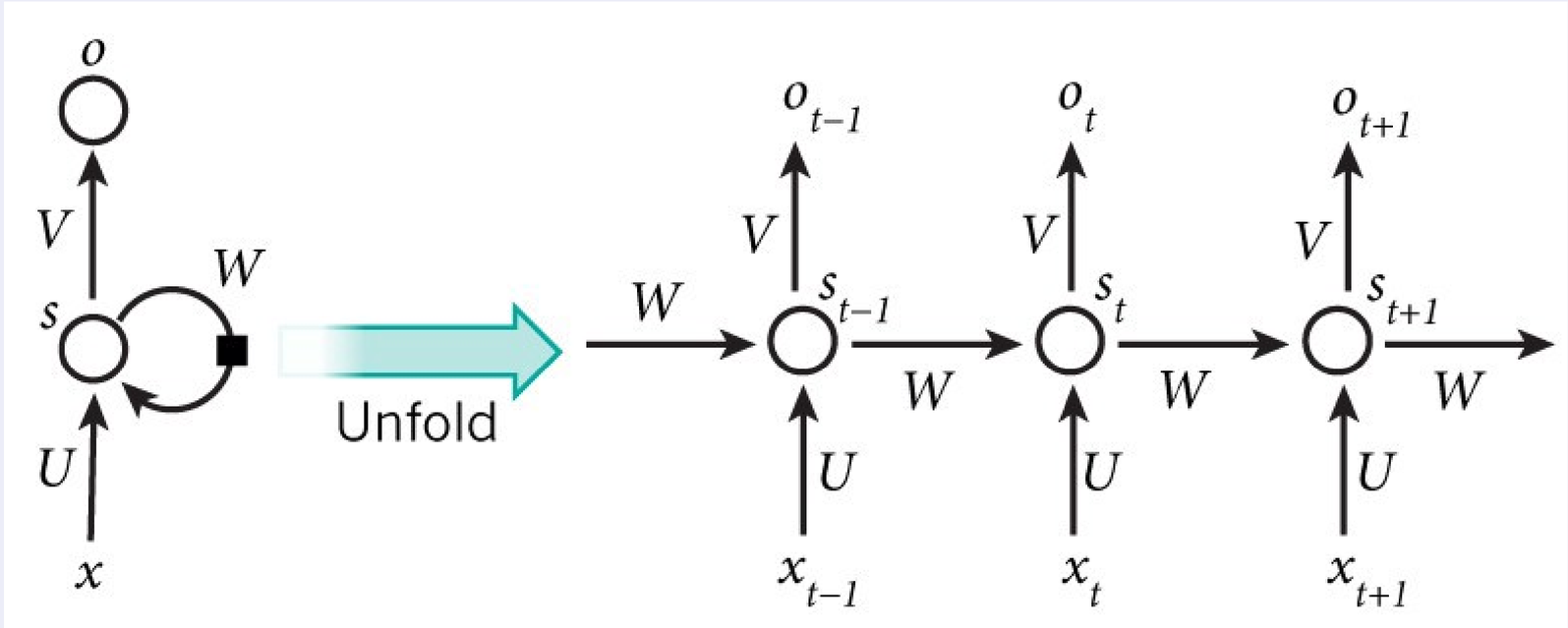


NLP WITH RECURRENT NEURAL NETWORKS

- Recurrent Neural Networks are a very important variant of neural networks heavily used in Natural Language Processing.

- Conceptually they differ from a standard neural network as the standard input in an RNN is a word instead of the entire sample as in the case of a standard neural network.

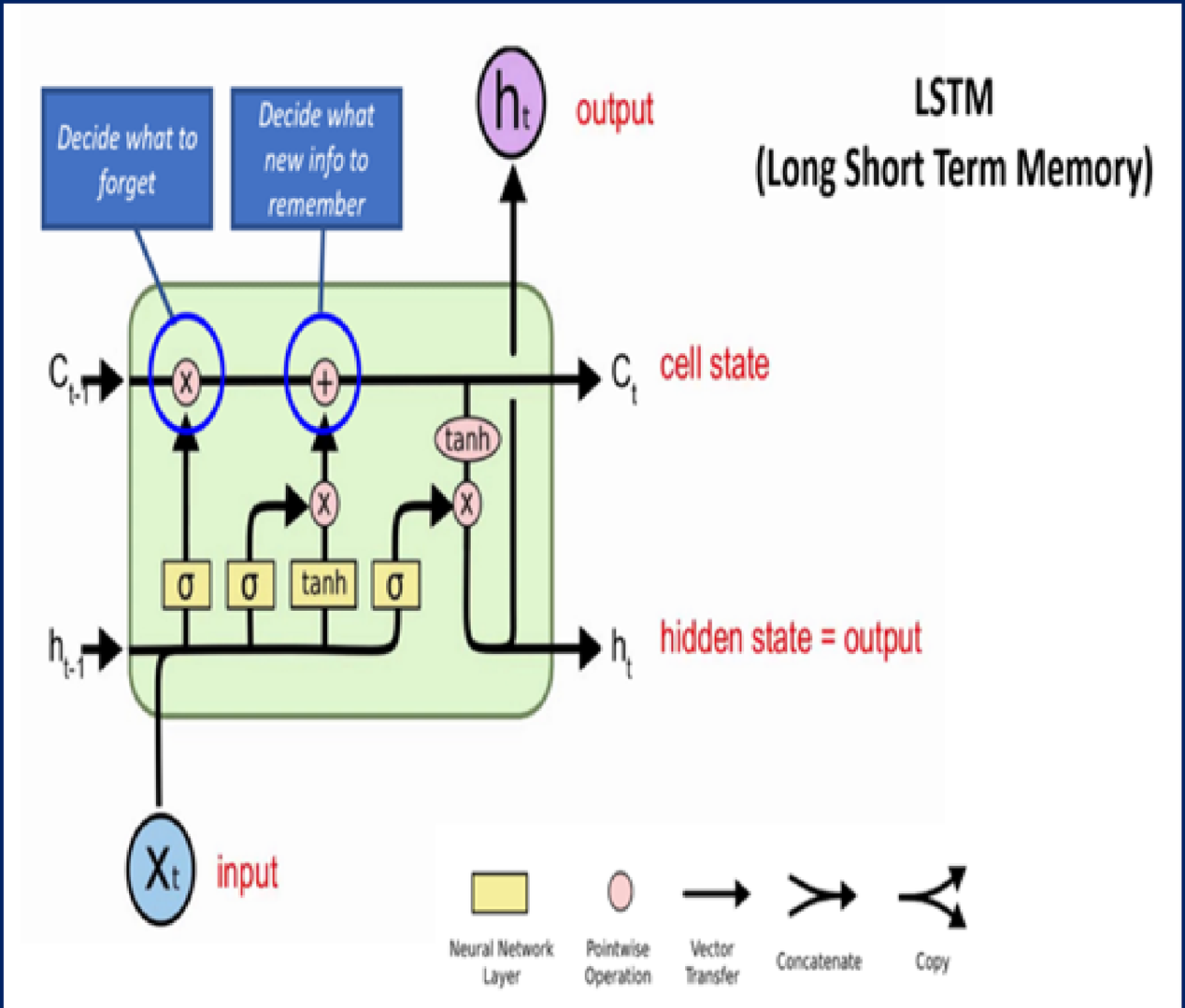
- This gives the flexibility for the network to work with varying lengths of sentences, something which cannot be achieved in a standard neural network due to its fixed structure.



The above diagram shows an RNN being unrolled (or unfolded) into a full network. By unrolling we simply mean that we write out the network for the complete sequence.

For example, if the sequence we care about is a sentence of 5 words, the network would be unrolled into a 5-layer neural network, one layer for each word.

LSTMS - BETTER THAN RNN



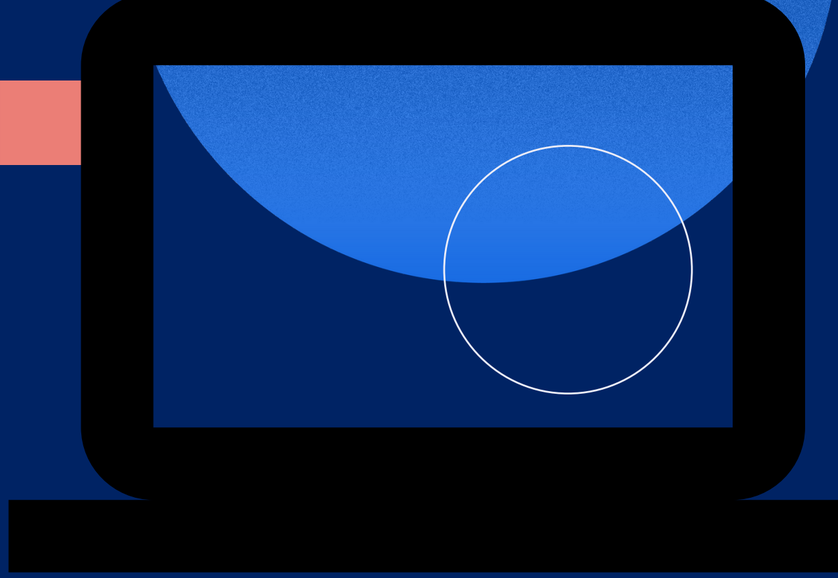
- Although an RNN can learn dependencies, however, it can only learn about recent information.
- Long-Short Term Memory networks or LSTMs are a variant of RNN that solve the Long term memory problem of the former. They have a more complex cell structure than a normal recurrent neuron, which allows them to better regulate how to learn or forget from the different input sources.
- LSTMs also mitigate the problems of exploding and vanishing gradients

- The cell state is updated twice with few computations that resulting stabilize gradients.

REMEMBER!

DON'T FORGET

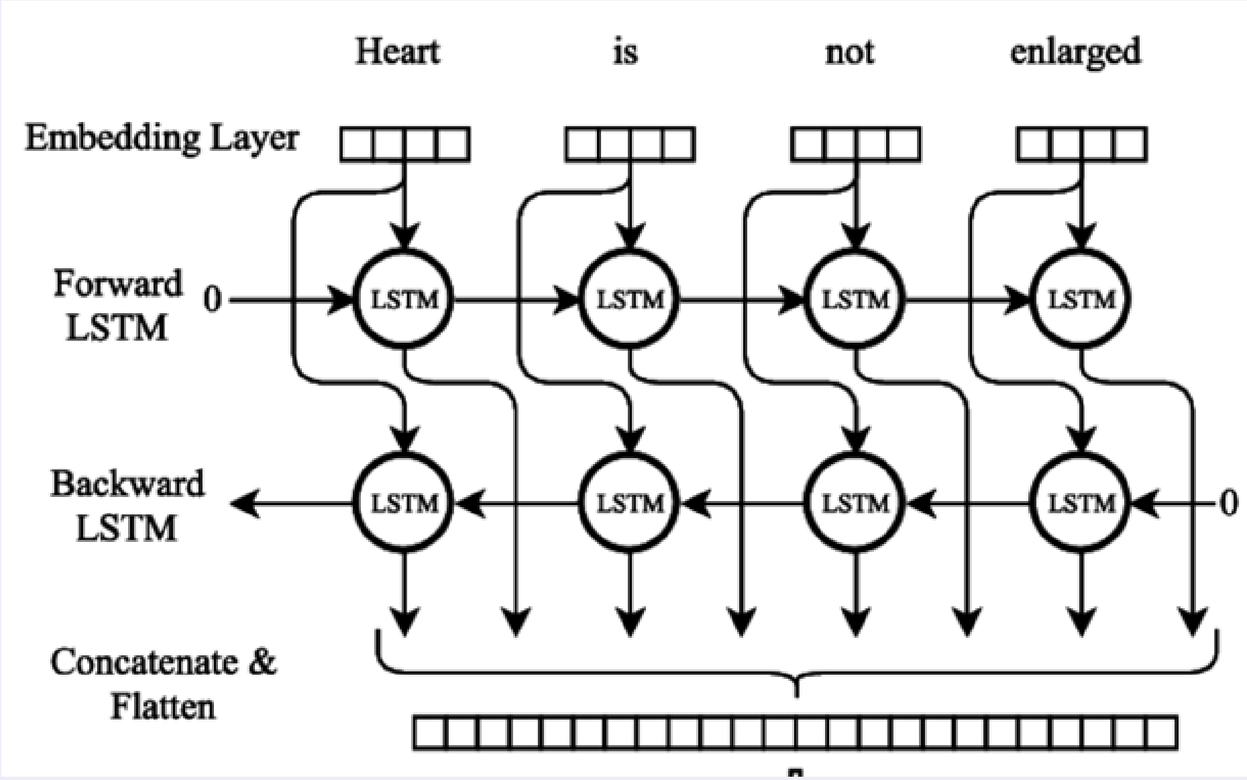
HOW LSTMS WORK WITH NLP :



- Take a sequence of words as the input.
- For each word, look up word embeddings to convert words into vectors.
- For each step (word embedding), we choose what information from the previous state to forget (e.g. the sentence before is in past tense and the current sentence is in present tense, then we “forget” the fact that previous sentence is in past tense as it is not relevant in this sentence), what information from the current state to remember.
- All this information is stored in a new state called the cell state, and the role of the cell state is to bring along information from the past that should be remembered.
- For each step, we have the hidden state which takes in information from the cell state to decides what to output for each word.

HOW BIDIRECTIONAL LSTMS WORKS WITH NLP

- Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems.
- Generally, where all timesteps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTMs on the input sequence.
- The first on the input sequence as-is and the second on a reversed copy of the input sequence. This can provide additional context to the network and result in faster learning on the problem. So , It usually has both ‘forward’ and ‘backward’ directional pass of data.
- The use of providing the sequence bi-directionally was initially justified in the domain of speech recognition and NLP Based applications because of its ability to interpret somehow strong contextual relations.



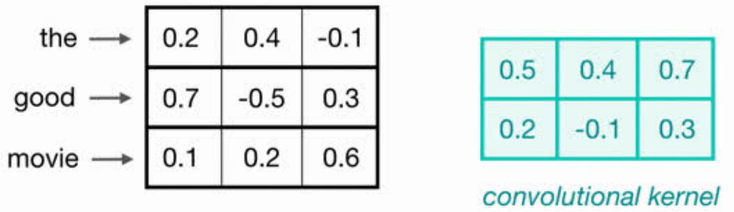
CONVOLUTIONAL NEURAL NETWORKS - CONV1D

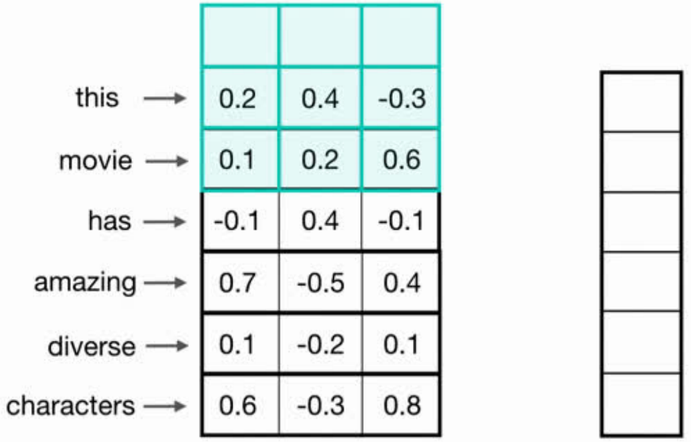
CNNs are often used in image processing, but Conv1D architecture has since been proven to be successful in solving NLP problems, especially in text classification.
Conv1D denotes ‘Convolution 1 Dimensional , so it can help to extract features from 1-dimensional plain data , like Text in case of NLP ,after being pre processed.

- It will extract and convert words into vectors using word embedding and Concatenate batches of word vectors into matrices.
- Then by putting word matrices into a layer similar to neural networks’ hidden layer, we use filters instead of weights.
- Each filter will represent a word category, e.g. categories for food, politeness, etc, so that after this layer each word matrix will have a score on each word category.
- For each word category, we choose the highest score out of all the layers.
- Finally , we will use a ‘softmax’ function to classify the text , given we are performing multi class classification.

CNN-1D

For example, in the sentence “I love this movie”, the score for “positive sentiment” would be very high because we have the word “love”. Even though “this” or “movie” may not be particularly positive, the CNN can identify this sentence as a positive sentence





BASIC PIPELINE FOR NLP WORKFLOW:

Let us Assume we are trying to perform multi-class classification and trying to build up Neural Network :

- Getting The RAW Data and Initial Preprocessing Of Text with its associated Labels.
- Creation of Training and validation data.
- Tokenizing, Text to Sequences, and Padding Conversion of raw Text.
- Using Labels Tokenizer for Converting Target labels to integers.
- Setting Up Initial Hyperparameters for Model Architecture.

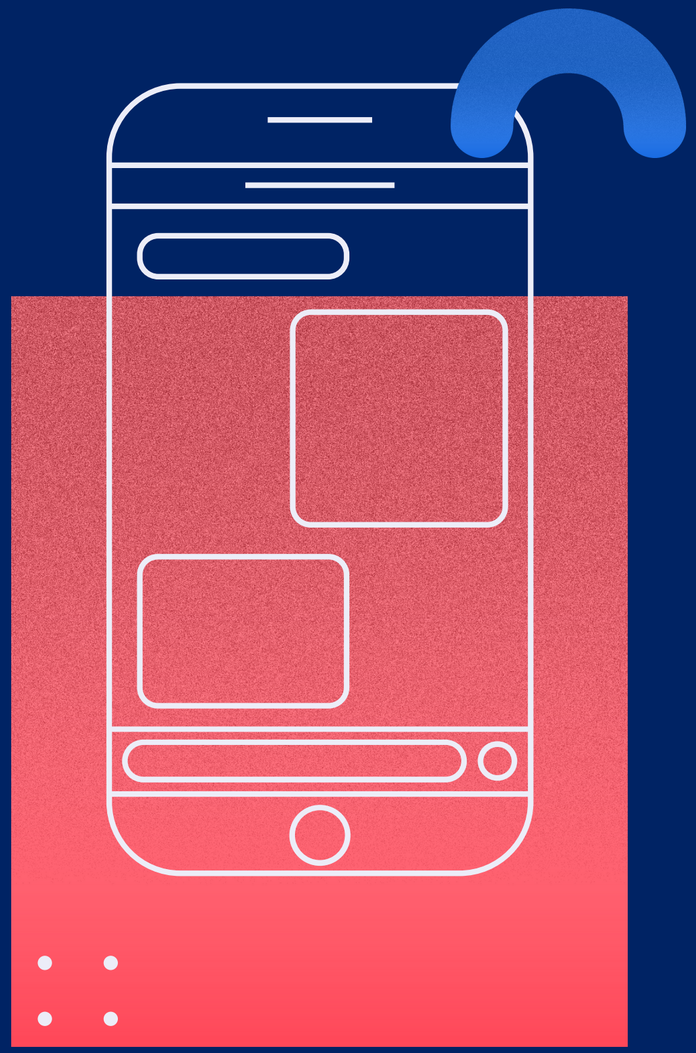
INSIDE MODEL :

- Embedding Layer The model begins with an embedding layer which turns the input integer indices into the corresponding word vectors. Word embedding is a way to represent a word as a vector. Word embeddings allow the value of the vector’s element to be trained. After training, words with similar meanings often have the similar vectors
- Neural Net layer: A neural Network Layer For Extracting Features and evaluating scores of input vs Target Text Sequences.For Example - LSTM or CNN-1D Layer.
- Dense Layer This the final layer, a Dense layer with softmax activation for the multi-class classification.

Model Ready to Train!!!

IT'S HANDS ON TIME

LET'S START CODING



We are going to perform a multi class classification on BBC News articles and its category. Based on the given text as an input, we will predict what would be the category. We have five type of categories: business, entertainment, politics, sport and tech.

In order to perform multi class classification , we will be taking a look on the following models :

Model 1 - CNN-1D
Model 2 - LSTM
Model 3 - Bidirectional LSTM
Model 4 - CNN-1D + LSTM
Model 5 - CNN-1D + Bidirectional LSTM

FINAL PROS AND CONS OF VARIOUS LSTMS USED:

Pros :

1. LSTM is able to preserve information from older steps, and hence is a solution to RNNs' vanishing gradient problem.
2. Due to its usefulness in comprehending the context, it is very often to see bidirectional LSTM in papers and applications.

Cons :

1.Unfortunately, LSTM does not solve RNN's parallelization problem, as each hidden state and cell state has to be computed before the next hidden state and cell state can be computed. This also means that LSTMs take a longer time to train, and require more memories.
2.As both RNN and LSTM described so far goes on one direction: left to right. This means that for tasks such as machine translation, where we are missing context from the words we have not yet seen.

FINAL PROS AND CONS OF CNN 1-D USED:

Pros:

1. Good performances for classification: as noted in the paper, although the simple CNN-static model only had little fine-tuning for its parameters, it did well even when compared to more sophisticated deep learning models, including some RNN models.

2.Parallelization is possible, therefore more efficient and versatile than RNNs.

Problems:

1.Padding is required: in order for the model to take in all the words, padding is required before the first word and after the last word, which might increase the sparsity of matrices while computing.

2.1D Based CNN Models might be able to pick a few and light ‘contextual’ words, but not a full-fledged contextual relation in a single sentence.

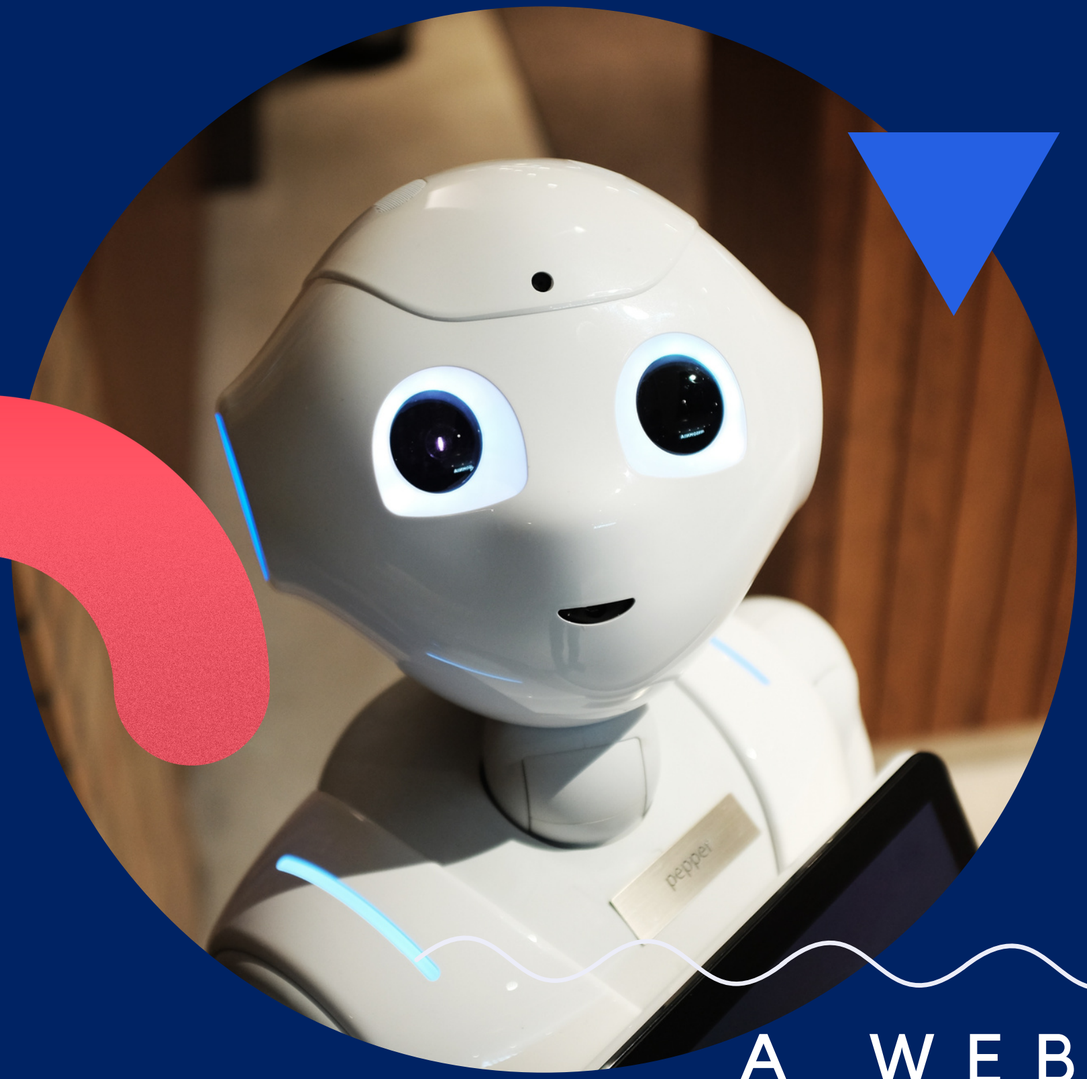
FINAL TIPS THAT YOU CAN TRY :

●Natural Language Processing Has a lot of wide applications and varieties. Therefore relying on one single model might not be that efficient. Hence, you can always try a hybrid version,i.e, LSTM+CNN, to obtain good results. Combining multiple models into an ensemble by averaging their predictions is a proven strategy to improve model performance

●Instead of Using Latent Embeddings, you can also use pre-trained word embeddings for obtaining better-featurised representations and hence better accuracy. Example -> Google News or Wikipedia-based Pre Trained Word embeddings.

●Generally Deeper Neural layers with optimized regularisers like dropout layers can help to prevent overfitting of textual-based sequences of data.

●NLP-based tasks usually require heavy computations, therefore, different optimizers can be used with different learning rates. Usually, Adam generally outperforms SGD. However, while it converges much faster than SGD, it has been observed that SGD with learning rate and tuned momentum slightly outperforms Adam but with a slow pace.



THANKS A LOT

HOPE YOU ENJOYED IT

A WEBINAR BY

DATA SCIENCE COMMUNITY SRM