```
from nltk.stem import PorterStemmer
```

Stemming

-->stemming is nothing but chop-off approach whicn means cut's off word ending to get the root form.

Nature:Rule Based

usecase:when Performance matters more than linguistic accuracy(eg.search engines)

```
ps=PorterStemmer()

print(ps.stem("studies"), ps.stem("studying"), ps.stem("better"))

studi studi better
```

Lemmatization

-->Convert a word into its base dictionnary form (lemma) using vocabulery +Grammer

Nature:Lingustic+POS aware

Output:Valid Word

Example:NLP tasks needing meaning presentations(chatbots,text analytics,MLModels)

```
# Lemmatization
from nltk.stem import WordNetLemmatizer
lem = WordNetLemmatizer()
print(lem.lemmatize("studies"))
print(lem.lemmatize("studying", pos="v"))
print(lem.lemmatize("better", pos="a"))

study
study
good
```

Embedding

Embedding means converting text(or any discrete data like words ,sentences,images) into numerical vector

```
# Embeddings (Feature Extraction)

from transformers import AutoTokenizer, AutoModel
import torch

model_name = "sentence-transformers/all-MiniLM-L6-v2"
tokenizer = AutoTokenizer.from_pretrained(model_name)
```

```python
model = AutoModel.from_pretrained(model_name)

sentence = "Generative AI is powerful."

# Tokenize
inputs = tokenizer(sentence, return_tensors="pt")

# Get embeddings
outputs = model(**inputs)

# Convert to sentence embedding (mean pooling)
embedding = outputs.last_hidden_state.mean(dim=1)

print("Sentence Embedding Size:", embedding.shape)
print(embedding)
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/
_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your
settings tab (https://huggingface.co/settings/tokens), set it as
secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to
access public models or datasets.
  warnings.warn(
```

{"model_id":"f974db03ff474331b0fa37e90f19dab9","version_major":2,"vers
ion_minor":0}

{"model_id":"94a95217f5684aaa9a631040260f2672","version_major":2,"vers
ion_minor":0}

{"model_id":"2a452b55191146e984e689cf2670adb2","version_major":2,"vers
ion_minor":0}

{"model_id":"1bab67c71378483aa4abe77f874157fa","version_major":2,"vers
ion_minor":0}

{"model_id":"82d2dfb048a64b4c926627268b8de3b5","version_major":2,"vers
ion_minor":0}

```
WARNING:torchao.kernel.intmm:Warning: Detected no triton, on systems
without Triton certain kernels will not work
```

{"model_id":"8f21248d277a4634a71a062e9668385b","version_major":2,"vers
ion_minor":0}

```
Sentence Embedding Size: torch.Size([1, 384])
tensor([[-1.7787e-01, -5.9587e-01,  2.8073e-01, -3.8739e-02, -3.0393e-
01,
          8.9821e-02, -5.5097e-03, -1.0635e-01,  4.7133e-02, -3.7137e-
```

```
02,
         -4.6230e-01, -5.1503e-02,  2.2352e-01, -2.2739e-01, -1.1960e-
01,
          3.1323e-01,  6.0644e-01,  2.6610e-01, -5.1660e-01, -5.1871e-
01,
          1.3975e-01,  2.1718e-01,  3.1646e-01, -4.1030e-01,  2.4925e-
01,
          4.3116e-01, -1.5631e-01, -4.3562e-01,  6.7199e-01, -5.4777e-
01,
          2.9238e-01,  4.0740e-01, -1.8768e-01,  8.7738e-02, -8.4422e-
01,
          6.1939e-01, -6.3491e-01,  3.8403e-01,  4.8624e-01, -9.5839e-
02,
         -9.3169e-02,  9.6886e-02,  1.7126e-01, -6.3445e-01,  5.1117e-
01,
          4.4403e-03, -2.4051e-01, -1.9318e-03,  9.2241e-02,  1.1089e-
01,
         -8.5022e-01, -3.4584e-01, -2.8821e-01,  1.4890e-01,  1.5668e-
01,
          2.7837e-01,  2.6364e-01, -2.4727e-01,  1.6780e-01, -6.9004e-
03,
         -3.2883e-01, -4.5616e-02, -1.5443e-02, -2.4613e-01,  7.1361e-
01,
         -3.3362e-01,  2.2321e-01,  2.1636e-02, -2.1512e-02, -1.2244e-
01,
          4.9477e-01,  6.4926e-01, -1.3440e-01,  2.3271e-01,  4.6884e-
01,
          2.4687e-01,  2.4129e-02, -5.0732e-01,  6.3195e-01, -2.1741e-
01,
          1.8448e-01, -2.6944e-01,  4.2092e-02,  2.4728e-01,  3.9161e-
01,
         -1.5202e-01, -1.5278e-02,  1.6936e-01, -5.2004e-02,  4.1930e-
01,
         -1.8002e-01,  6.0778e-02,  1.4630e-01, -1.3125e-01,  9.9939e-
02,
          6.7936e-02,  4.1131e-02, -5.7219e-01, -3.2126e-01,  3.2507e-
01,
         -3.3564e-01,  3.3752e-01, -5.1899e-02, -3.0968e-01,  7.8358e-
02,
         -3.9308e-01,  6.2163e-01, -2.2338e-02,  1.7950e-01, -3.6619e-
01,
         -8.3551e-02, -1.3800e-01,  1.1586e-01, -4.4489e-01, -4.5895e-
01,
         -1.3034e-02,  2.4253e-01,  6.9445e-01, -1.0675e-01,  2.6828e-
01,
          1.4058e-01, -1.0135e-02, -2.9628e-01,  6.8591e-01,  5.3173e-
03,
         -7.2471e-01,  1.9597e-01, -2.1549e-32, -8.3879e-02,  1.4617e-
02,
         3.2882e-01,  8.6450e-01,  6.7117e-01,  9.7164e-02, -2.0261e-
01,
        -2.1832e-01, -1.8932e-01, -5.5703e-01, -5.2404e-01,  1.0268e-
01,
        -5.6742e-01,  5.1509e-01,  5.8889e-01, -2.7184e-01, -1.2362e-
01,
         2.9386e-01, -1.0636e-01, -2.0935e-01,  2.9733e-01, -6.7593e-
02,
        -3.7769e-02, -1.3770e-01, -2.8760e-02,  2.8602e-01,  6.5637e-
01,
        -1.4425e-01,  1.7235e-01,  5.5331e-02, -4.7078e-01, -3.5283e-
02,
        -3.4645e-01,  2.7858e-01, -1.8616e-01,  3.9540e-02, -3.5073e-
01,
        -1.8242e-01,  1.0137e-01,  4.8335e-01, -1.9430e-01,  2.0253e-
01,
        -5.4631e-02, -1.1555e-01, -3.1462e-01,  2.1685e-01,  4.4730e-
01,
         1.1216e-01, -2.6328e-01, -1.5219e-01,  3.8434e-02,  4.1007e-
01,
         3.4000e-01, -2.4571e-01,  1.8976e-01,  1.5738e-01,  1.0190e-
01,
         5.3163e-01,  9.4799e-02, -1.2878e-01,  2.2822e-02,  3.4933e-
01,
        -1.7235e-01,  7.1848e-01, -1.4583e-01,  8.3986e-02,  1.7693e-
01,
         3.5364e-03,  5.4471e-01,  3.1471e-01,  7.0977e-02, -4.1814e-
02,
        -3.5786e-02, -5.7777e-01, -3.8444e-01, -4.3772e-02,  9.3716e-
02,
        -3.8642e-01,  1.2118e-01, -1.0593e-01, -8.7319e-01,  2.9417e-
03,
        -2.4367e-01, -8.1017e-02,  5.9755e-01, -5.3475e-02, -6.3516e-
02,
        -7.0399e-02, -3.6929e-01,  3.0763e-01, -2.7121e-01, -3.1551e-
01,
         8.1024e-02,  1.8075e-03, -7.4652e-01,  2.2116e-32, -6.5617e-
01,
        -4.5844e-02, -2.9701e-01,  4.6948e-01, -3.6773e-01, -1.8997e-
01,
        -5.6716e-01,  2.0963e-01, -3.3631e-01, -1.8693e-01, -1.0692e-
01,
         4.5195e-01,  6.8958e-02, -1.7728e-01,  8.8086e-02, -2.1125e-
01,
         4.3804e-01, -6.4040e-02, -2.4411e-01, -3.0999e-02,  2.3539e-
01,
         9.6639e-01, -3.5803e-01, -1.4037e-02, -2.1074e-01,  3.1193e-
01,
        -6.4083e-01,  2.7243e-01, -1.9208e-03,  2.5978e-01,  1.3563e-
```

```
01,
         -3.5192e-01, -1.0327e-01,  1.5638e-01, -5.0429e-01,  4.0612e-
01,
          4.8566e-01,  7.6197e-02, -4.7605e-01,  6.5227e-01,  1.0689e-
01,
          9.0244e-02, -6.5505e-01,  2.5241e-01, -2.2128e-01, -1.2927e-
01,
         -2.7164e-02,  1.4206e-01,  2.2536e-01,  6.8476e-01, -8.9964e-
02,
          6.1583e-02, -5.3056e-01, -6.2061e-01, -5.8759e-01, -3.8214e-
01,
          2.4871e-01, -3.8704e-01,  1.8732e-01, -8.1745e-02, -5.4448e-
01,
         -4.1911e-01,  1.4963e-01,  6.6102e-02,  1.1854e-01,  1.0142e-
01,
         -2.5902e-01,  6.9981e-02,  1.8782e-01, -3.1665e-01,  2.9020e-
01,
          1.2163e-01, -6.6940e-03,  5.3171e-01,  1.5771e-02,  1.6241e-
01,
          1.0972e-01,  9.6378e-02,  1.0055e-01, -8.6552e-01,  1.7811e-
01,
         -4.6272e-01,  2.5856e-01, -1.5263e-01,  1.4495e-01,  1.9415e-
01,
          1.7046e-01, -4.5235e-02, -3.6246e-02,  3.3515e-01, -2.7906e-
01,
          1.6485e-01,  2.3053e-01,  3.9298e-02, -5.9461e-01, -9.1625e-
08,
         -1.9207e-01, -2.4894e-01,  6.8256e-01, -4.0690e-02,  5.1340e-
01,
         -9.2984e-02, -4.1751e-01,  4.1394e-01, -2.4099e-01, -1.6613e-
01,
          4.3301e-01, -1.4509e-01, -7.7160e-02,  3.1908e-01,  8.9053e-
01,
         -1.2464e-01,  4.3208e-02,  4.1325e-01, -4.7693e-01,  1.0673e-
02,
          3.7721e-01,  6.9834e-02, -2.6876e-01,  7.8558e-02,  1.2499e-
01,
         -2.7087e-01, -2.3754e-01, -1.9778e-01, -3.3002e-01,  5.0278e-
01,
         -6.4471e-03,  4.0013e-01,  1.7316e-01,  3.6946e-01,  3.3426e-
01,
          5.0020e-01, -2.7357e-02, -1.8436e-01, -2.4601e-01, -3.2875e-
01,
          2.2791e-02,  4.5198e-01, -1.7115e-01,  9.6636e-03,  1.8151e-
01,
         -4.1547e-02,  2.2768e-01, -1.1629e+00,  1.4420e-01,  4.2559e-
01,
          2.8944e-01,  1.0382e-01,  3.9571e-01,  2.7584e-02,  5.6935e-
01,
          1.1514e-01,  2.2647e-01, -2.4009e-01,  1.0294e-01,  3.4508e-
01,
          1.5022e-01,  4.8730e-01,  3.4122e-01, -1.2641e-01]],
       grad_fn=<MeanBackward1>)
```