In [1]:
```
import torch
import torch.nn as nn
import string #Text Processing Without NLP Libraries
```

In [2]:
```
paragraph = """
Machine Learning (ML) is a fascinating field that enables COMPUTERS to LEARN fro
It uses ALGORITHMS like decision trees, neural networks, and k-means clustering.
various applications.ML is transforming how we live and work.By 2030, this TECHN
TRANSPORT and EDUCATION. Are we ready for ML?!"""
```

In [3]:
```
# List to store the extracted tokens (words)
# Temporary string to build a word character by character
#The function tokenize(text) takes a string (text) as input.
def tokenize(text):
    tokens = []
    word = ''
    # Iterate through each character in the input text
    for char in text:
      # Check if the character is a whitespace or punctuation (word boundary)
        if char in string.whitespace or char in string.punctuation:
          # If there's an accumulated word, add it to the tokens list and reset
            if word:
                tokens.append(word)
                word = ''
        else:
             # If the character is not whitespace or punctuation, it is added to
            word += char
    if word:
       tokens.append(word) # Add the last word to the tokens list
    return tokens

tokens = tokenize(paragraph) #extracts words from the paragraph.
print(tokens)
```

```
['Machine', 'Learning', 'ML', 'is', 'a', 'fascinating', 'field', 'that', 'enable
s', 'COMPUTERS', 'to', 'LEARN', 'from', 'data', 'and', 'make', 'predictions', 'I
t', 'uses', 'ALGORITHMS', 'like', 'decision', 'trees', 'neural', 'networks', 'an
d', 'k', 'means', 'clustering', 'Over', '50', 'of', 'industries', 'now', 'use',
'ML', 'in', 'various', 'applications', 'ML', 'is', 'transforming', 'how', 'we',
'live', 'and', 'work', 'By', '2030', 'this', 'TECHNOLOGY', 'will', 'shape', 'man
y', 'fields', 'including', 'TRANSPORT', 'and', 'EDUCATION', 'Are', 'we', 'ready',
'for', 'ML']
```

Processing

Character	Action
`H`	Start word `"H"`
`e`	`"He"`
`l`	`"Hel"`
`l`	`"Hell"`
`o`	`"Hello"`
`,` (Punctuation)	Add `"Hello"` to tokens, reset `word`
(Whitespace)	Skip

In [7]:
```
#This splits the paragraph string whenever a period (.), exclamation mark (!),
#or question mark (?) appears.
#s.strip() != '' ensures that empty strings are ignored.
import re
sentences = re.split(r'[.!?]', paragraph)
sentence_count = len([s for s in sentences if s.strip() != ''])
print("Number of Sentences:", sentence_count)
```

```
Number of Sentences: 6
```

In [9]:
```
#Iterates through each word in the tokens list.
#Checks if the word is completely uppercase using .isupper().
#If the condition is met, the word is included in the uppercase_words list.
uppercase_words = [word for word in tokens if word.isupper()]
print("Uppercase Words:", uppercase_words)
# Step 2: Convert the uppercase words to lowercase
lowercase_words = [word.lower() for word in uppercase_words]
print("Converted to Lowercase:", lowercase_words)
```

```
Uppercase Words: ['ML', 'COMPUTERS', 'LEARN', 'ALGORITHMS', 'ML', 'ML', 'TECHNOLO
GY', 'TRANSPORT', 'EDUCATION', 'ML']
Converted to Lowercase: ['ml', 'computers', 'learn', 'algorithms', 'ml', 'ml', 't
echnology', 'transport', 'education', 'ml']
```

In [8]:
```
# Define a set of stopwords manually
stopwords = {"is", "a", "that", "to", "from", "and", "it",
             "like", "of", "in", "how", "we", "for", "by", "this"}
# Tokenize the paragraph by splitting on spaces
#paragraph is split into a list of words using the .split() method.
words = paragraph.split()
# Remove stopwords
#Iterates through each word in the words list and converts the word to lowercase
# Checks if the word is not in the stopwords set,
#If the word is not in stopwords, it is included in filtered_words.
filtered_words = [word for word in words if word.lower()
                  not in stopwords]
# Print filtered words as tokens
print(filtered_words)
```

```
['Machine', 'Learning', '(ML)', 'fascinating', 'field', 'enables', 'COMPUTERS',
'LEARN', 'data', 'make', 'predictions.', 'uses', 'ALGORITHMS', 'decision', 'tree
s,', 'neural', 'networks,', 'k-means', 'clustering.', 'Over', '50%', 'industrie
s', 'now', 'use', 'ML', 'various', 'applications.ML', 'transforming', 'live', 'wo
rk.By', '2030,', 'TECHNOLOGY', 'will', 'shape', 'many', 'fields,', 'including',
'TRANSPORT', 'EDUCATION.', 'Are', 'ready', 'ML?!']
```

In [ ]: