Introduction to Computational Linguistics  
Exploring PyTorch for NLP

Reference to this Colab [PyTorch tutorial.ipynb](https://colab.research.google.com/drive/12rOYfXTdpIjLXXtxvgVfQor1vtZwCG8J#scrollTo=78-gPpKy3wqc)

# Tokenizer Class

1. What is the purpose of the `word2idx` and `idx2word` dictionaries in the `Tokenizer` class, and how are they initialized?
2. What does the `pad\_or\_truncate` method do, and why is it necessary when working with neural networks?
3. How does the `tokenize\_dataset` method handle unknown words that are not seen during training?
4. What happens if a sentence contains only words that were not seen in the training set? How does the tokenizer handle it?
5. If `seq\_length = 10` and an input sentence contains 15 tokens, how will the tokenizer modify the sentence? What if it contains only 5 tokens?

# TextDataset Class

1. What is the purpose of the `TextDataset` class, and how does it relate to PyTorch’s `Dataset` class?
2. What is the expected shape and data type of the tensors returned by `\_\_getitem\_\_`?
3. Why does `\_\_getitem\_\_` return `lengths[idx]`, and how might this be useful when designing a neural network for text classification?

# Deep Averaging Network (DAN)

1. What is the purpose of the `nn.Embedding` layer in this model, and why is `padding\_idx=0` specified?
2. How does dividing by `lengths.unsqueeze(1).float()` create an average embedding for each sequence? What would happen if this step were omitted?
3. Why is `Softmax(dim=1)` applied at the final layer, and what does `dim=1` refer to?
4. What happens if `lengths` contains zeros? Why might this cause issues in the computation of `avg\_embeds`, and how could you fix it?

# Dataset Preparation

1. What is the difference between `tokenize\_training\_set` and `tokenize\_dataset`, and why is `tokenize\_training\_set` used for the training set while `tokenize\_dataset` is used for validation and test sets?
2. What is the role of the `label\_dict`, and why are labels converted into integers instead of keeping them as strings?
3. Why do we use dictionary lookups (`label\_dict[label]`) when creating `Y\_train`, `Y\_dev`, and `Y\_test`? What would happen if a label were missing from `label\_dict`?
4. If the dataset had an additional label, say 'Spam', how would you modify the code to include it?

# Training Loop

1. What is the purpose of defining `VOCAB\_SIZE`, `EMBED\_DIM`, `HIDDEN\_DIM`, and `OUTPUT\_DIM` before instantiating the model?
2. Why do we use `DataLoader(train\_dataset, batch\_size=BATCH\_SIZE, shuffle=True)` instead of passing the dataset directly to the training loop?
3. What is the purpose of calling `model.train()` before training and `model.eval()` before evaluation?
4. What happens when `optimizer.zero\_grad()` is called before computing the loss and performing backpropagation?
5. What would happen if `shuffle=True` were omitted from `train\_dataloader`? Would this affect the training process?
6. Why is `torch.no\_grad()` used during evaluation, and what would happen if it were removed?