Week 2 - Learn to Search

Welcome to week 2 :)

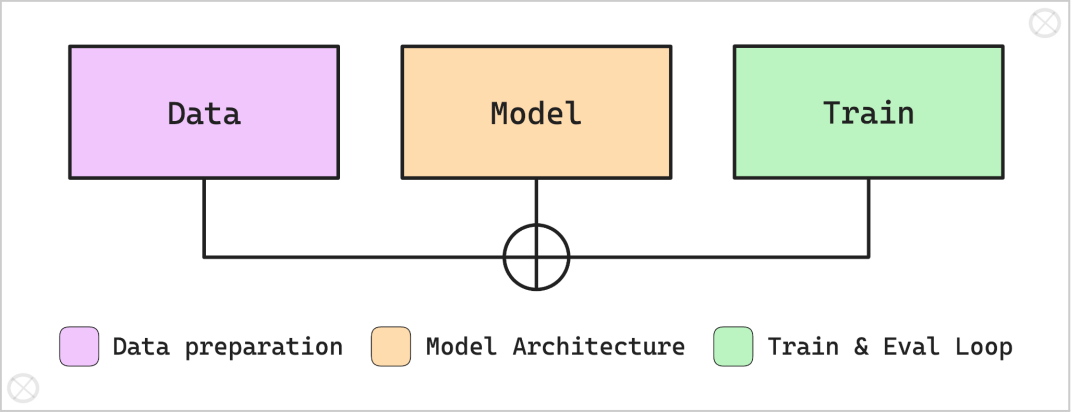
As usual we will cover new concepts and architectures. Take a deep breath and welcome to another week of incredible learning.

Ensure you understand the big picture and the main concepts and discuss them out loud with your team. This approach will help you cover ground and stay on track.

Goal of the week

This week's goal is to generate embeddings for both documents and queries. Then create a multi-stage search pipeline, first candidate generation and then ranking.

Make sure to have clean code and well defined abstractions. If useful, leverage folder structure and class inheritance. Keep this in mind and don't rush when defining your data processing, modelling and train loop. Make sure they are clean.



Learning outcomes

Make sure to understand, at least at a high level, what each of those concepts represent. Plan ahead with your team and cover each topic on its own and as part of the main goal.

The main learning outcomes are:

- SentencePiece tokeniser
- Word2Vec to generate token embeddings
- RNNs (maybe GRU, LSTM) to generate sentence embeddings
- kNN and PCA for candidates generation
- Two Tower Neural Networks for ranking
- Weights & Biases library to keep track of your runs
- TF-IDF old methodology (just ChatGPT it)

Datasets

Microsoft Machine Reading Comprehension (MS MARCO) is a collection of datasets for deep learning related to Search. You can find more about the dataset on the official website. Feel free to look at the papers who made it to the leaderboards, they might inspire you with new architectures. Use the version v1.1 from HuggingFace.

Architecture

Below I outlined a path you might consider following. Similar to the previous week:

- tokenize all our dataset and output a vocabulary of words or subwords
- assign each token an embedding and train it accordingly using one of the models
- then use the pre-trained embeddings and further create sentence embeddings
- use the sentence and query embeddings in a Two Tower to learn associations
- produce a score that will be used to rank documents vs a given query