Dense Passage Retriever (DPR)

The DPRContextEncoder (along with the DPRQuestionEncoder) is primarily trained using a contrastive loss function.

Here's a breakdown of how it's typically trained:

Dataset: The training requires a dataset of (question, positive passage, negative passage) triplets or, more commonly, pairs of (question, positive passage) within a batch where other passages in the same batch serve as "in-batch negatives". A well-known dataset for this is the Natural Questions (NQ) dataset, which is often used to train the base DPR models you see loaded from Hugging Face.

...

Dual Encoder Architecture: DPR uses a dual-encoder architecture. This means it has two independent encoders:

=====> DPRQuestionEncoder: Encodes the question.

=====>DPRContextEncoder: Encodes the passage.

These encoders are typically initialized with pre-trained language models (like BERT), but they are fine-tuned independently.

...

Training Objective (Contrastive Loss): The goal of the training is to adjust the parameters of both encoders such that:

=====>The embedding of a question is close to the embedding of its corresponding positive passage.

=====>The embedding of a question is far from the embeddings of negative passages.

The contrastive loss function (often an In-Batch Negative Softmax Loss, as briefly mentioned before) quantifies how well the model achieves this goal.

...

In-Batch Negative Softmax Loss Explained:

Consider a training batch containing N question-passage pairs:

$(q_1, p_1^+), (q_2, p_2^+), \ldots, (q_N, p_N^+)$, where p_i^+ is the positive passage for q_i.

The DPRQuestionEncoder computes the embedding for each question e_{q_i}.

The DPRContextEncoder computes the embedding for each positive passage $e_{p_i^+}$.

For a given question q_i, all other positive passages in the batch p_j^+ $(j \neq i)$ are treated as

negative passages. So, for q_i, the negative passages are $p_j^+ \mid j \neq i$. The embeddings for these are $e_{p_j^+}$.

The model computes similarity scores (often using dot product) between the question embedding e_{q_i} and the embedding of every passage in the batch (both its positive and all the negatives). Let $s(e_a, e_b) = e_a \cdot e_b$.

For question q_i, the model wants the similarity $s(e_{q_i}, e_{p_i^+})$ to be higher than all $s(e_{q_i}, e_{p_j^+})$ for $j \neq i$.

This is framed as a multi-class classification problem where, for each question q_i, the "correct class" is its positive passage p_i^+, and the "negative classes" are all other passages in the batch.

A Softmax function is applied to the similarity scores between q_i and all passages in the batch, turning them into probabilities. The loss is the negative log-likelihood of the model predicting the positive passage p_i^+ as the most relevant one.

The loss for the i-th question L_i using In-Batch Negatives is:

$$L_i = -\log \frac{\exp(s(e_{q_i}, e_{p_i^+}))}{\sum_{j=1}^{N} \exp(s(e_{q_i}, e_{p_j^+}))}$$

The total loss for the batch is the average loss across all questions: $L = \frac{1}{N} \sum_{i=1}^{N} L_i$.

...

Optimization: Gradient descent is used to minimize this loss function. During backpropagation, the gradients flow through both the DPRQuestionEncoder and the DPRContextEncoder, updating their weights simultaneously. This training process forces the encoders to learn representations where relevant questions and passages have high similarity scores, and irrelevant ones have low scores.

In essence, the DPRContextEncoder learns what makes a passage relevant to a question by being trained to produce embeddings that are close to the corresponding question embeddings and far from irrelevant question embeddings, all within the contrastive learning framework.