TWO-STAGE APPROACH FOR DSI

Amine Ahardane
University Sapienza
Rome
ahardane.2050689@studenti.uniroma1.it

ABSTRACT

In this research, we propose a novel two-stage model approach to enhance document retrieval using the MSMARCO dataset. The first stage, we embed the documents using all-MiniLM-L6-v2, and use these embeddings to cluster them into groups usimg KMeans, specifically into 2,000 clusters. Then using Cluster Predictor Model we predict the relevant cluster for a given query, thereby significantly reducing the search space and improving model efficiency. This model can be trained extensively one time and can be used in many cases. The second stage, the Refinement Model (seq2seq), refines the search within the predicted cluster to accurately retrieve the specific document. Both models are trained and evaluated on a dataset composed of 80,000 training query-document pairs, 18,000 test pseudo-queries –document pairs, and 2,000 evaluation pseudo-queries –document pairs, pseudo-queries generated using the doc2query/all-t5-base-v1 model.

***K*eywords** Information Retrieval · Document Retrieval · Deep Learning · Clustering · MSMARCO

1 Introduction

1.1 Problem Statement

The current paper presents a way in which clustering techniques and sequence-to-sequence models can be combined to potentially improve document retrieval performance in large datasets.

We use the power of MSMARCO, a benchmark collection very widely applied for training and evaluating IR systems. Alone with its rich source of queries and corresponding document passages, it will be an ideal dataset to run our experiments. In this paper, we mainly have two components: Cluster Predictor and Refinement Model.

The Cluster Predictor is designed to assign each query to a related document cluster. Specifically, we can use the all-MiniLM-L6-v2 pre-trained SentenceTransformer model to encode queries into dense vector representations, which later predict cluster ID and subsequently lead to semantic grouping.

The other one is the seq2seq Refinement Model, which further refines retrieval based on the output from the Cluster Predictor. What this model does is to exactly specify which document in the predicted cluster fits best with the query. This refines the documents' retrieval precision by integration of the query embeddings with cluster embeddings.

First, train the cluster predictor model independently; then train the combined model (Cluster Predictor + Refinement Model) with the weights of the Cluster Predictor model frozen. This way, the Refinement Model will be adjusted towards better retrieval accuracy within the predicted clusters.

1.2 Objectives

This research aims to improve document retrieval performance by introducing a two-stage model approach. The proposed approach consists of:

1. **Cluster Predictor Model**: A model designed to reduce the search space by clustering documents and predicting the relevant cluster for a given query.

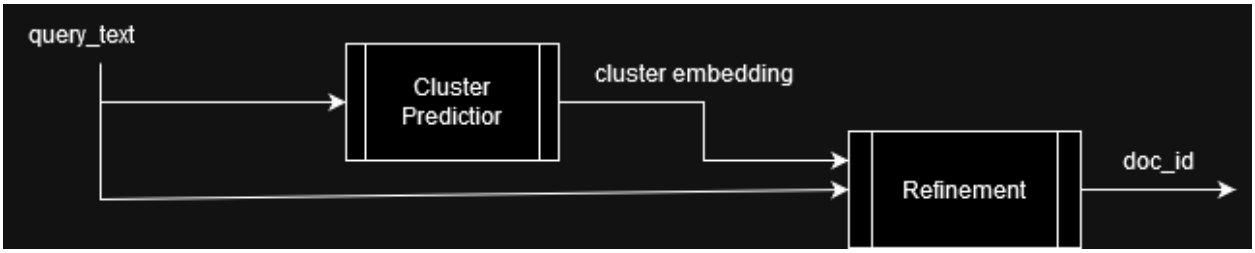

Figure 1: Architecture of the two-stage DSI.

2. **Refinement Model**: A seq2seq model that refines the search within the predicted cluster to accurately retrieve the specific document.

1.3 Significance of the Study

The significance of this study lies in its potential to enhance the efficiency and accuracy of document retrieval systems. By reducing the search space through clustering and refining the search within clusters, the proposed approach addresses the limitations of traditional retrieval methods. We may extensively train the Cluster Predictor Model using it in different applications as a pretrained model, having 'half' of the job done. And then using state of the art DSI models that exploits Cluster Predictor Model. The MSMARCO dataset provides a suitable testbed with its diversified large query-document pairs for assessing the efficacy of the proposed models.

2 Methodology

2.1 Dataset

We use the MSMARCO dataset, which consists of real anonymized user queries from Bing and corresponding relevant passages. The dataset includes a training set with over 500,000 queries and a development set with around 6,980 queries. For our experiments, we sampled 80,000 queries for training, 18,000 queries for testing, and 2,000 queries for evaluation. In particular, testing and evaluation set are sampled from the training set. Then by using doc2query/all-t5-base-v1 model, we substitute each original query (of testing and evaluation dataset) with a generated pseudo-query, ensuring that the documents used to evaluate the model are consistent with those seen during the training phase.

2.2 Preprocessing and Augumentation Dataset

The preprocessing steps involve tokenizing the queries and docouments using the all-MiniLM-L6-v2 model. we also tokenize the same queries using another tokenizer, T5 tokenizer. This is because the same query, later, will be used in two different model as input. Using all-MiniLM-L6-v2, we embed each document and then we cluster them, having 2,000 clusters using KMeans. Each query is assigned to the nearest cluster based on the embedding of the corresponding document.

2.3 Cluster Predictor Model

The Cluster Predictor Model utilizes the all-MiniLM-L6-v2 variant of SentenceTransformers. It embeds queries into dense vectors, which are then used to predict cluster IDs. The model is trained using the Cross-Entropy loss function to minimize the error in cluster prediction.

2.4 Refinement Model

The Refinement Model integrates the contextual understanding of the T5 model with the structured information provided by the cluster logits. In particular, the Refinement Model takes the predicted cluster logits and refines the search to retrieve the most relevant document.This model leverages the T5-small architecture to generate the doc id for the most relevant document. The process begins by encoding the query text into an embedding using the T5 encoder. Simultaneously, the predicted cluster logits are projected into smaller space using a linear layer. These projected cluster logits are summed with the query embedding to result in a combined embedding. This combined embedding is then fed into the decoder of T5, which should generate the doc id for the document most relevant to the query. The specific implementation of this model is as follows:

1. **Model Initialization**: The Refinement Model is built upon the T5-small model. A linear layer is used to project the cluster logits (which have a higher dimensionality) into the same space as the T5 embeddings.
    - The model is initialized with `AutoModelForSeq2SeqLM.from_pretrained('t5-small')`.
    - A projection layer (`nn.Linear`) maps the cluster embeddings to match the T5 input embedding dimensions.
2. **Forward Pass**: During the forward pass:
    - The query text is tokenized and passed through the T5 encoder to produce `query_embeddings`.
    - The cluster logits, which are predictions from the Cluster Predictor Model, are passed through the projection layer to match the dimensionality of the T5 embeddings.
    - These projected cluster embeddings are then summed with the query embeddings to create a combined embedding.
    - The combined embedding is then passed through the T5 decoder to predict the `doc_id`.
3. **Loss Function**: The model is trained using the CrossEntropyLoss

2.5 Inference Process

During the inference phase, the Refinement Model employs beam search to enhance the quality of the generated `doc_id` sequences. Beam search is a heuristic search algorithm that explores multiple possible sequences simultaneously and selects the top-k most likely sequences based on their probability scores.

1. **Beam Search**:
    - The model uses beam search to explore multiple potential `doc_id` sequences.
    - **Beam Width**: The beam width (number of sequences considered at each step) is set to a predefined value (e.g., 5), balancing the trade-off between computational efficiency and retrieval accuracy.
    - **Top-k Sequences**: The model generates the top-k sequences with the highest probability scores, considering these as the most likely `doc_id` candidates.
2. **Final Selection**: The sequence with the highest probability among the top-k generated sequences is selected as the final `doc_id` output for the query.

In preliminary experiments, the combined model was trained for a single epoch due to computational constraints but showed promising potential that could be improved with extended training.

2.6 Baseline: Pyserini BM25

To provide a baseline for comparison, we utilize the Pyserini library to implement the BM25 retrieval model. Pyserini is an IR toolkit built on Lucene, offering robust support for modern IR research. We index the same set of documents used in our experiments and perform retrieval using BM25.

3 Experimental Setup and Results

3.1 Evaluation Metrics

We evaluate the performance of our models using the following metrics:

- **Mean Average Precision (MAP)**: Measures the mean of the average precision scores across all queries, focusing on the ranking of relevant documents. This metric evaluates how well the system ranks relevant documents higher in the retrieval list.
- **Set_MAP**: Evaluates the performance in set-based retrieval scenarios by measuring how well the system retrieves the entire set of relevant documents for each query, treating the retrieved documents as a set rather than a ranked list.
- **Precision at K (P@K)**: Measures the proportion of relevant documents in the top-K retrieved documents, assessing the accuracy of the system at a specific cutoff rank.

3.2 Training and Validation Losses

Figures 2 and 3 plots the training step loss and validation step loss for the Cluster Predictor Model for four different learning rates: 2×10^{-8}, 2×10^{-5}, 1×10^{-3}, and 1×10^{-2}, while keeping a batch size equal to 2048..

The order of the loss curves in both graphs, from top to bottom, is: 2×10^{-8}, 2×10^{-5}, 1×10^{-3}, and 1×10^{-2}. Here, one can notice a trend: the smaller the learning rate, the better the training and validation loss.

This outcome aligns with expectations, since we are using a large batch size.

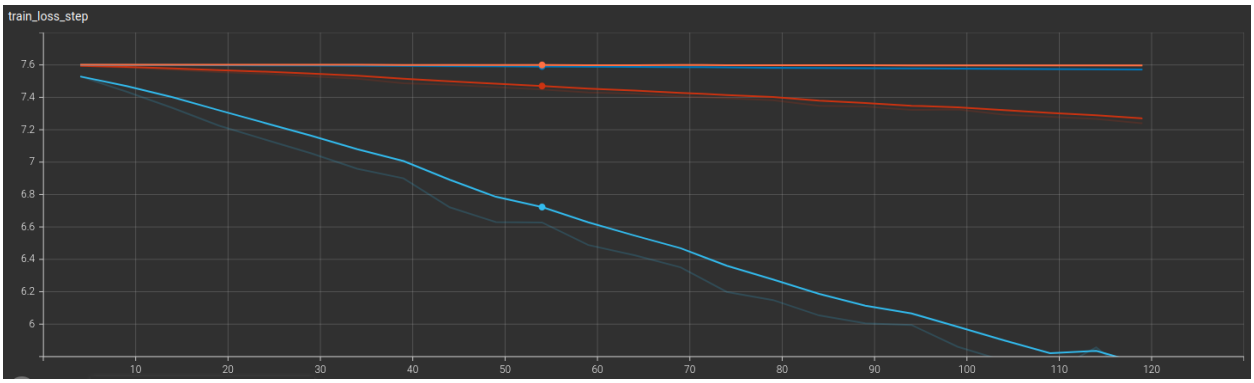

Figure 2: Train step loss when training only the cluster predictor model.

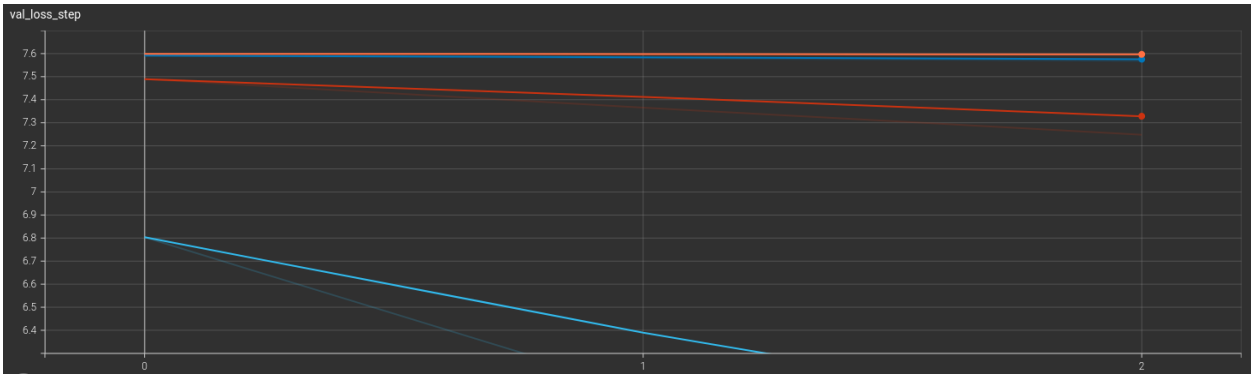

Figure 3: Validation step loss when training only the cluster predictor model.

Similarly, Figures 4 and 5 illustrate the training and validation losses for the combined model (Cluster Predictor + Refinement Model). The steep decline in loss during initial steps followed by stabilization suggests that the model was learning effectively during the initial phase but plateaued, possibly due to limited training epochs (just 1).

Table 1: Evaluation Results

Model	MAP	P@3	Set_MAP
BM25 Baseline	0.6954	0.2543	0.0012
Combined Model	0.0000	0.0000	0.0000

3.3 Results

Table 1 contains results showing that in every metric, a BM25 baseline substantially outperforms our proposed models. More specifically, the metrics MAP, P@3 and Set_MAP are equal to zero or near zero and this indicates that we have a poor performance in our current approach due to not having sufficient training.

It should, however, be noted that the Cluster Predictor Model alone was able to achieve an accuracy of about 30% in correctly predicting the cluster ID. It means that the model can narrow down almost one-third of the search space to a

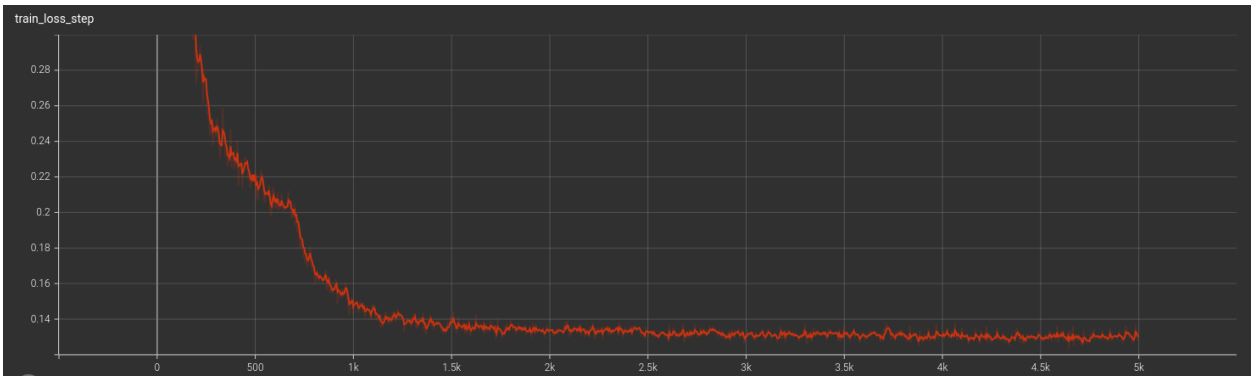

Figure 4: Train step loss when training combined model.

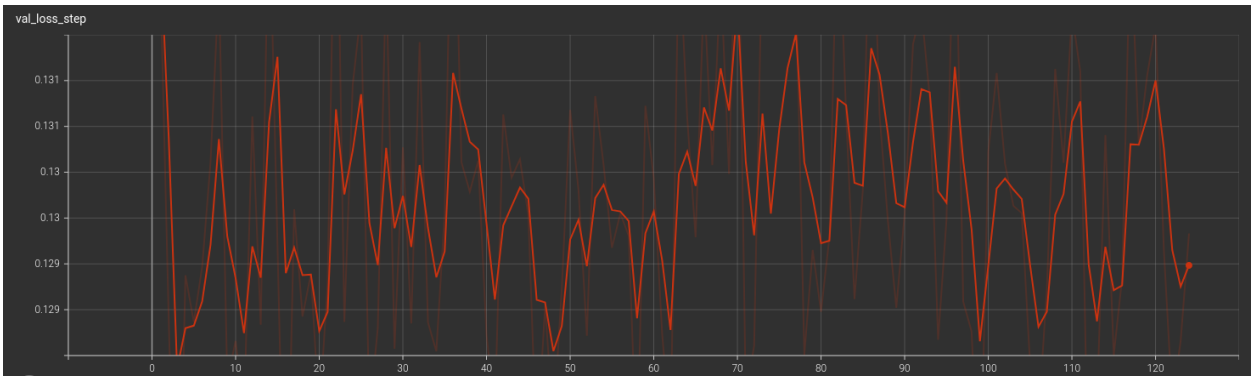

Figure 5: Validation step loss when training combined model.

smaller and relevant subset of documents. We believe this accuracy could be much improved with further training and hence the overall performance of this two-stage retrieval system to be improved.

4 Discussion

Experimental results, indicates that our proposed method, is not able to provide competitive results with respect the baseline. The primary reason is the limited compotational resources available, in fact we trained the model only for one single epoch, and it is not enough to capture the complex patterns.

Despite the poor evaluation, the conceptual framework of our two-stage approach reamins promising. The idea of exploiting clustering techniques to reduce the search space allowing the subsequent seq2seq model be more prices, is theoretically sound strategy that could improve retrieval accuracy if given sufficient training time.