Optimizing Search Efficiency: Differential Learning & Differentiable Search Index Approaches

Information Retrieval

May 8, 2024

Navid Azimi, Alireza Rafiei

Department of Computer Science, Emory University

Introduction

Motivation

- The efficacy of IR systems in providing pertinent document rankings in response to user queries is essential
- Traditional IR systems commonly employ an index-then-retrieve pipeline (not always the most efficient approach!)
- Alternative approaches like the Differentiable Search Index (DSI) aim to integrate indexing and retrieval processes into a unified model. By doing so, they offer the potential for more seamless and efficient document ranking in response to user queries.

Problems

- **Sequential nature of traditional IR methods:** Indexing occurs first, followed by retrieval
  → Latency issues, especially when dealing with large datasets or in real-time search scenarios
- **Struggle with handling dynamic or evolving datasets:** The index becomes outdated over time
  → Stale search results and diminish the user experience

Goal

- Inspired by the differential learning approach and Differentiable Search Index concept, our goal is to merge these traditionally separate stages and developing a unified model, designated as 'f', utilizing a sequence-to-sequence architecture, which handles user queries ('q') and employs an auto-regressive approach to generate relevant document IDs.

Dataset

The MS MARCO dataset → Pre-built index provided by **Pyserini** library

1 Building Resources

- **documents (dictionary):** document ID (docid) as key, a dictionary with field 'raw' (containing the raw text as string) as value
- **queries (dictionary):** query ID as key , a dictionary with field 'raw' (containing the raw text as string) and 'docids_list' (containing the list of correlated document IDs) as value

2 Computing Word2Vec Embeddings for queries and documents

- Using the trained Word2Vec model (on corpus data, containing all the processed 'raw' data)
- **Processed 'raw' data:** Converted to lowercase, tokenized, and stopwords and punctuation removed
- **Created 2 types of embeddings for each query/document:** emb, obtained as the average of the embeddings of all words, and first_L_emb, obtained by concatenating the embeddings of the first MAX_TOKENS words

3 Creating Datasets for Siamese Models

- **Pairwise Dataset (query, document, relevance):**

  → Designed for pairwise learning, where each sample consists of a query paired with a document
- **Triplet Dataset (query, doc+, doc-):**

  → Designed for triplet learning, where each sample comprises a query along with a positive and a negative document for training

Dataset

Example of Pairwise/Triplet Datasets

4 Creating Datasets for Sequence-to-Sequence (seq2seq) Model

- **Document Dataset (encoded document, encoded docid):**

  → Designed to facilitate training Seq2Seq models by providing documents as input sequences

- **Retrieval Dataset (encoded query, encoded docid):**

  → Designed for Seq2Seq models used in retrieval tasks. It pairs documents with corresponding queries for training

Documents and queries encodings are computed using the **T5-small tokenizer**, and we choose to pick the first **L=32** tokens for representing the documents, and the first **L=9** for representing the queries.

Example of Document/Retrieval Datasets

Siamese Neural Network: SNN-Convolutional (Baseline)

Siamese Neural Network: Neural network architecture that consists of two identical subnetworks, which have the same architecture and share the same parameters (weights)

→ Learn the similarity or dissimilarity between pairs of inputs (Differential learning approach)

1 Convolutional Siamese Neural Network (using Word2Vec embeddings)

Steps:

- **Changing Dataset Return Type:** The dataset pairs_dataset return type is modified to 'emb' (Word2Vec embeddings)
- **Initializing Siamese Network (Parameters):**
  - input_size: Set to the dimensionality of Word2Vec embeddings
  - conv_channels: The number of channels for convolutional layers
- **Training the Model (Parameters):**
  - pairs_dataset, siamese_net, max_epochs, batch_size, split_ratio, etc.

- Inability to effectively discriminate between relevant and random documents, assigning high similarity scores to irrelevant documents

	MAP		nDCG@10		Precision@10		Recall@1K	
	Train	Test	Train	Test	Train	Test	Train	Test
CNN-SNN	0.1181	0.0500	0.1937	0.1782	0.0399	0.0100	0.3900	0.5

Siamese Neural Network: SNN-Attention

2 Siamese-Attention-Net Transformer (using token embeddings)

- **Attention network:** Designed to identify the highest correlations amongst words within a sentence, assuming that it has learned those patterns from the training corpus
- This enables learning representations based on token-level information (capture dependencies regardless of their distance in the input sequence.)
- Generates a relevance score using cosine similarity

Steps:

- **Changing Dataset Return Type to Token Embeddings:** The return type of the dataset pairs_dataset is adjusted to 'first_L_emb' (embeddings of first L tokens) - Use the stack of the first MAX_TOKENS embeddings (first_L_emb) of queries and documents
- **Initializing Siamese Transformer Network (Parameters):**
  embedding_size (total): Set to the product of EMBEDDING_SIZE and MAX_TOKENS
- **Training the Model (Parameters):** pairs_dataset, siamese_transformer, max_epochs, batch_size, split_ratio, etc.

- This model also struggles to distinguish relevant documents from random ones, frequently attributing high scores to both

	MAP		nDCG@10		Precision@10		Recall@1K	
	Train	Test	Train	Test	Train	Test	Train	Test
SNN-Att	0.0999	0.0	0.034	0.0236	0.01	0.0	0.08	0.0499

Siamese Neural Network: SNN-Contrastive

3 Siamese Lightning module using embeddings (Contrastive Learning Approach with Triplet Loss)

- **Triplet loss function:** The network learns by comparing a set of three inputs: an anchor image, a positive image (similar to the anchor), and a negative image (dissimilar to the anchor). The goal is to bring the anchor and positive image embeddings closer while pushing the negative embedding further away.

Steps:

- **Changing Dataset Return Type to Embeddings:** The return type of the dataset `triplets_dataset` is adjusted to 'emb' (embeddings)
- **Initializing Siamese Lightning Module (Parameters):**
  input_size: Set to the dimensionality of embeddings (EMBEDDING_SIZE)
  margin: Defines the margin for the triplet loss function
  arch_type: Specifies the architecture type, set to 'linear'
- **Training the Model (Parameters):** triplets_dataset, siamese_lightning_module, max_epochs, batch_size, split_ratio, etc.

- This method, by leveraging the principles of contrastive learning, showed notable effectiveness over previous baselines and effectively discerns between similar and dissimilar query-document pairs

	MAP		nDCG@10		Precision@10		Recall@1K	
	Train	Test	Train	Test	Train	Test	Train	Test
SNN-Cons	0.2725	0.1706	0.3784	0.3211	0.1659	0.1599	0.7319	0.666

DSI Transformer-Based Model

1 Sequence-to-Sequence Model

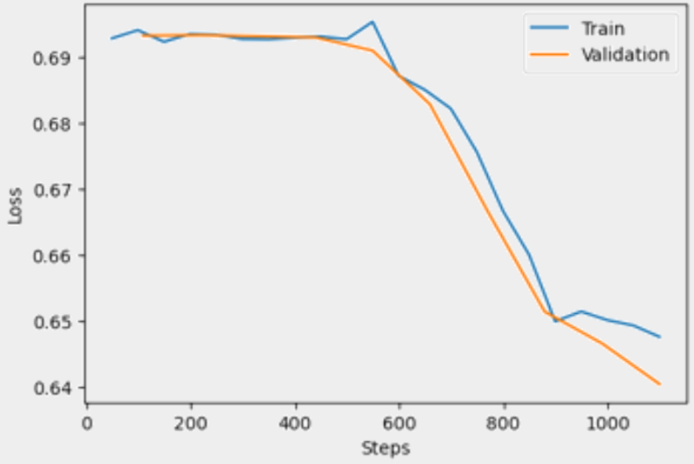
Model's architecture: A transformer-based encoder-decoder that involves three transformer layers for each component. Encoder plays the role of indexing, in which the model has to acquire the documents knowledge to map every document to its related docid. Meanwhile, the decoder is tasked with retrieval, generating the complete target sequence from a given query. Ranking is conducted according to semantic similarity.

Steps:

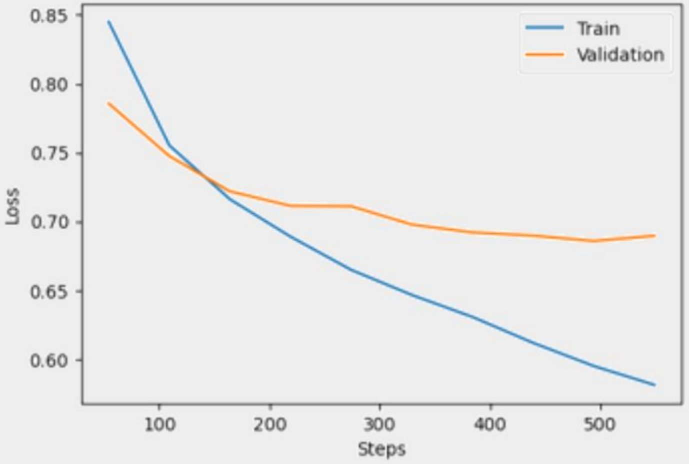
- **Initializing the sequence-to-sequence model (Parameters):**
  nlayers: Set the number of transformer layers for the encoder-decoder model
  nhead: The number of heads in the multi-head attention models
  nhid: The dimension of the feedforward network model
  droupout: The dropout rate for each layer
- **Training the Model (Parameters):**
  tokenized_dataset, Transformer_module, max_epochs, batch_size, split_ratio, etc.
- This model can be a more generalizable one, but needs more powerful computational racecourse for training

	MAP		nDCG@10		Precision@10		Recall@1K	
	Train	Test	Train	Test	Train	Test	Train	Test
Seq-2-Seq	0.0682	0.0684	0.0909	0.0906	0.0167	0.0167	0.1678	0.1671

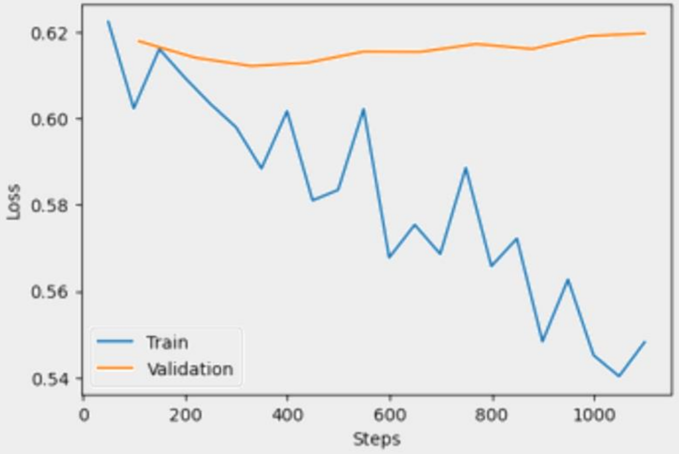
DSI Transformer-Based Model



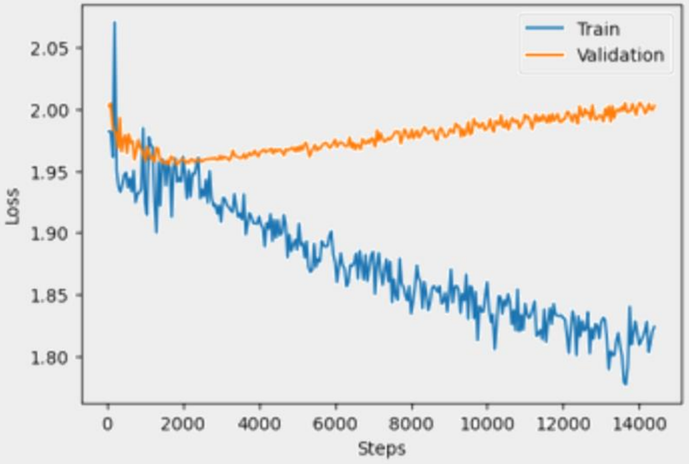
Siamese Network



Siamese-attention Network



Siamese-contrastive Network



Sequence-to-Sequence Model

Conclusion

Evaluation Metrics In congruence with: Pyserini Reproductions for MS Marco Passage → MAP, nDCG@10, Pr@10, and R@1K

Comparing the Results

	MAP		nDCG@10		Precision@10		Recall@1K	
	Train	Test	Train	Test	Train	Test	Train	Test
CNN-SNN	0.1181	0.0500	0.1937	0.1782	0.0399	0.0100	0.3900	0.5
SNN-Att	0.0999	0.0	0.034	0.0236	0.01	0.0	0.08	0.0499
SNN-Cons	**0.2725**	**0.1706**	**0.3784**	**0.3211**	**0.1659**	**0.1599**	**0.7319**	**0.666**
Seq-2-Seq	0.0682	0.0684	0.0909	0.0906	0.0167	0.0167	0.1678	0.1671

1 According to our observations, DSI can make the training process faster, and needs less computational resources for IR tasks

→ But might not show better efficiency compared to the index-then-retrieve approaches

Future Works

- If computational resources are available, we can evaluate the trained models using a higher number of epochs
- A comprehensive comparison in terms of time, computational costs, scalability, and efficiency between the developed models and other index-then-retrieve pipelines

References

1 **Transformer memory as a differentiable search index**

Tay, Y., Tran, V., Dehghani, M., Ni, J., Bahri, D., Mehta, H., ... & Metzler, D. (2022), Advances in Neural Information Processing Systems, 35, 21831-21843.

2 **Fully Convolutional Siamese Networks for Change Detection**

Daudt, R. C., Le Saux, B., & Boulch, A. (2018, October), In 2018 25th IEEE international conference on image processing (ICIP) (pp. 4063-4067). IEEE.

3 **Siamese hierarchical attention networks for extractive summarization**

González, J. Á., Segarra, E., García-Granada, F., Sanchis, E., & Hurtado, L. I. F. (2019), Journal of Intelligent & Fuzzy Systems, 36(5), 4599-4607.

4 **Together we stand: Siamese networks for similar question retrieval**

Das, A., Yenala, H., Chinnakotla, M., & Shrivastava, M. (2016, August), In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 378-387).

5 **Beamqa: Multi-hop knowledge graph question answering with sequence-to-sequence prediction and beam search**

Atif, F., El Khatib, O., & Difallah, D. (2023, July), In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 781-790).

?

Thank You!