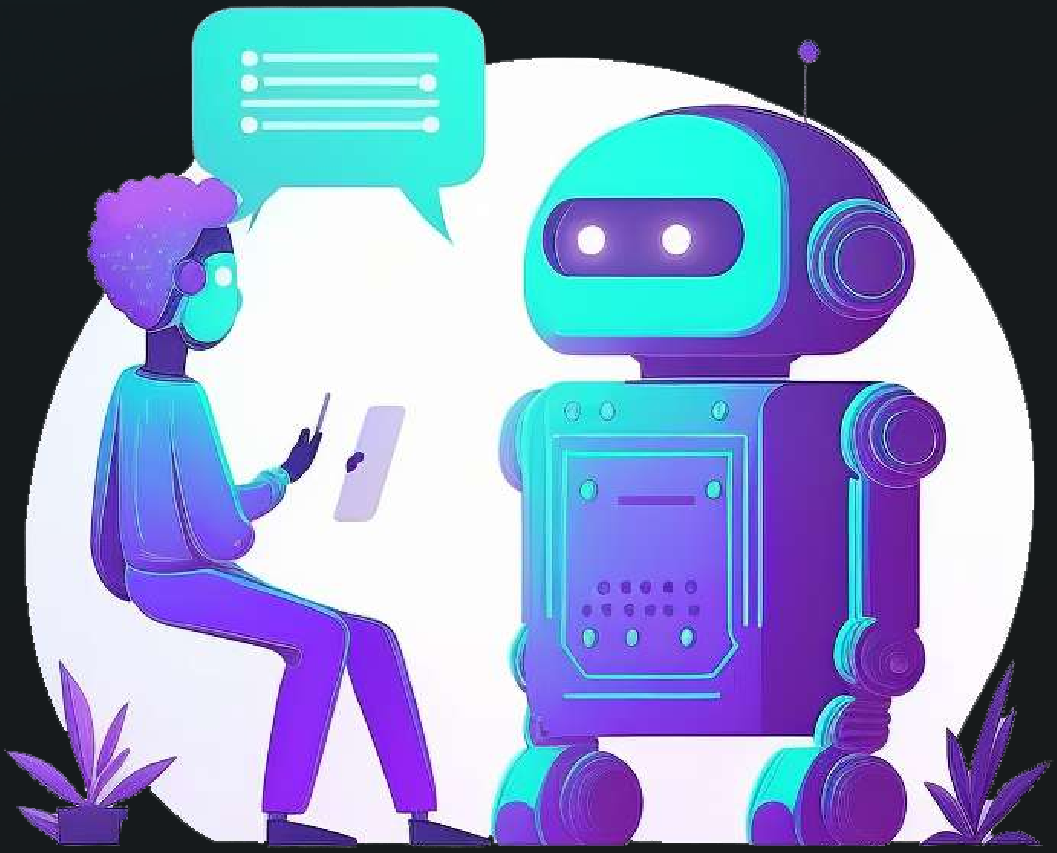


MID TERM REPORT

Expert Answers in a Flash: Improving Domain-Specific QA

We are building a model to develop a system that retrieves a relevant knowledge base article for a given query. Our objective is to select the best article from the extensive knowledge base and highlight the specific portion of the passage that mainly answers the question. What makes the problem interesting is that various organizations can have diverse knowledge bases. Thus we need a general solution that can be fine-tuned on the required knowledge bases. Also, It is not guaranteed that any given query can be answered with the existing knowledge, and thus the question is not answerable. Hence, the model needs to know that a given query can not be answered with its knowledge, and therefore the query needs to be passed to a human support agent.

Primary Team Number: 45 **Secondary Team Number: 32**

Literature Review

Retrieval

Open Domain Question Answering (ODQA) has been studied widely recently and a classic framework of ODQA system is implemented by encompassing an information retriever (IR) and a reader, i.e., Retriever-Reader. The task of IR is to retrieve evidence-related text pieces from the large knowledge corpus. Popularly used IR can be TF-IDF, BM25 and DPR (dense passage retriever) , etc. The target of the reader is understanding and reasoning the retrieved evidence to yield the answer. It is often achieved by transformer-based language models, such as BERT, RoBERTa, ALBERT or sequence-to-sequence generator T5, BART, GPT, etc. There is a very recent approach where we are stacking the retriever layer, reranking layer and the reading layer into a single model name YONO which aims at reducing the model size.

There are different types of framework along which the question answering models have been developed namely:

- Retriever and Reader
- Retriever-only
- Generator-only.

Most recent works follow Retriever and Reader framework and further supersede the TF-IDF or CNN based retriever with stronger transformer-based models, such as BERT, T5, BART, etc. The readers can be classified into generative and extractive readers. Dense Passage Retriever directly leverages pre-trained BERT models to build a dual-encoder retriever without additional pre-training. Dual-encoder retrievers like DPR, encode the questions and documents independently, ignoring interaction between questions and documents, and limiting their retrieval performance. To remedy this issue, Colbert adds interaction between different embeddings on the top of a dual-encoder, and ColbertQA applies it into ODQA domain to gain better performance.

Retriever-only systems tackle ODQA tasks with a single retriever, eliminating reading or generating step. Generator-only ODQA models are normally based on single generators, mainly seq2seq generative language models, like T5, GPT and BART.

However, most general-purpose ODQA models are computationally intensive, inference slowly, and training expensive. One reason is the huge index/document size (see Table 2). Concretely, a corpus typically contains millions of long-form articles that need to be encoded and indexed for evidence retrieval. As we want our model to deliver answers quickly by using limited resources, all these resource heavy and slow inference methods are not appropriate for our tasks.

Question Answering

Early approaches of Question Answering involved rule-based systems[28], relying on predefined rules and patterns to extract answers from the text. These systems had limited capability to handle complex and ambiguous questions. They were not robust to out-of-domain tasks and languages. With the advent of Machine learning, large text datasets were used to train language models with a self-supervised training mechanism, leading to considerable improvements in answer extraction and generation. These methods relied on information retrieval (IR) techniques that involved ranking and retrieving the most relevant documents or passages from a given dataset.

Recently, Large deep-learning models have offered significant accuracy gains, but training these large language models is challenging. These models are not usable in real-time applications due to resource and device constraints. Currently, many pre-trained models are available on Hugging Face (RoBERTa, BERT, Deberta-v3, XLM Net, etc.) for extractive Question Answering. Given the resource constraints in the problem statement, these models are rendered unusable for this task. We research possible methods to optimize the available models on CPU[26] are:

- Quantization
- Pruning[25]
- Distillation[22]
- Model Architecture

Most works on model compression focus on "distilling" a pre-trained model through expensive finetuning, while some reduce model complexity by structured pruning of model parameters. Structured Pruning of BERT-based Question Answering Models[20] uses a hybrid combination of task-specific structured pruning and distillation and shows significant gains in speed and performance. Zero Redundancy Optimizer (ZeRO)[29] optimizes language model memory requirements, enabling lower latency.

Paragraph Retrieval Task

A. Task Description

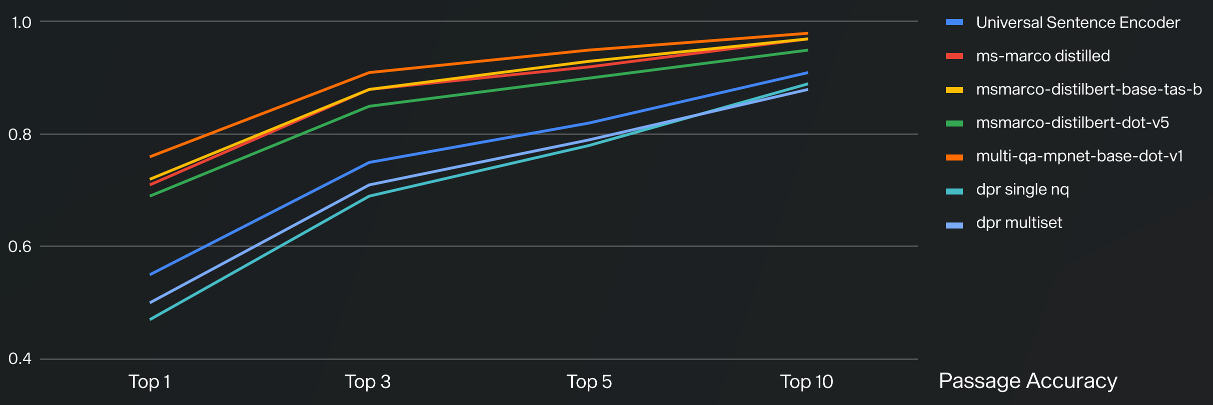
Question answering systems usually involve a retriever that selects relevant subsets of given text corpora that may contain the answer. The question and the context i.e. paragraphs are encoded and the question encoding is used to retrieve the most relevant context encodings which may contain the answer. The focus lies on making the retrieval as efficient and effective as possible, since it affects the performance of the downstream task of finding the correct answer span.

B. In Depth Approaches with Results

Our approach can be broadly subdivided into two parts: passage-based retrieval and phrase/sentence-based retrieval. Under both these settings, we broadly classify our approach as cross-encoder, bi-encoder/embedding based or a combination of both approaches.

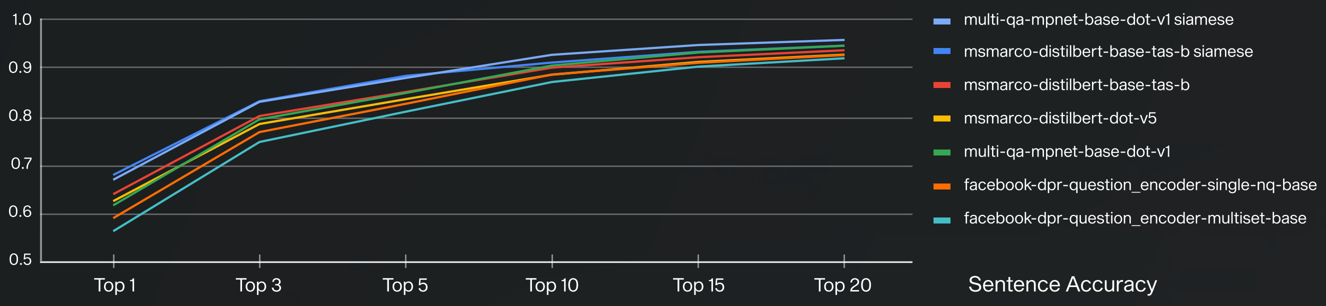
1. Phrase vs Passage based approaches

We follow the intuition that retrieving phrases naturally entails retrieving larger passages. This is particularly appealing because the phrases can directly be used as the output for question answering. This approach reduced inference time by 70% while achieving both better retrieval accuracies and QA F1 scores over its passage counterparts. However, this approach does not take into account answers that span multiple sentences. Another con of phrase-based approaches is possible context loss (provided by other sentences in the passage) which is also an issue for its passage counterparts (in the form of mean pooling). This context loss could be mitigated by combining a few sentences. But we have little incentive for doing so, since most questions in SQuAD have single phrase answers.



Different Model Performance With Passage Based Approach

**Tables containing the results for validation fold 4 for all models (comparison b/w passage and phrase)

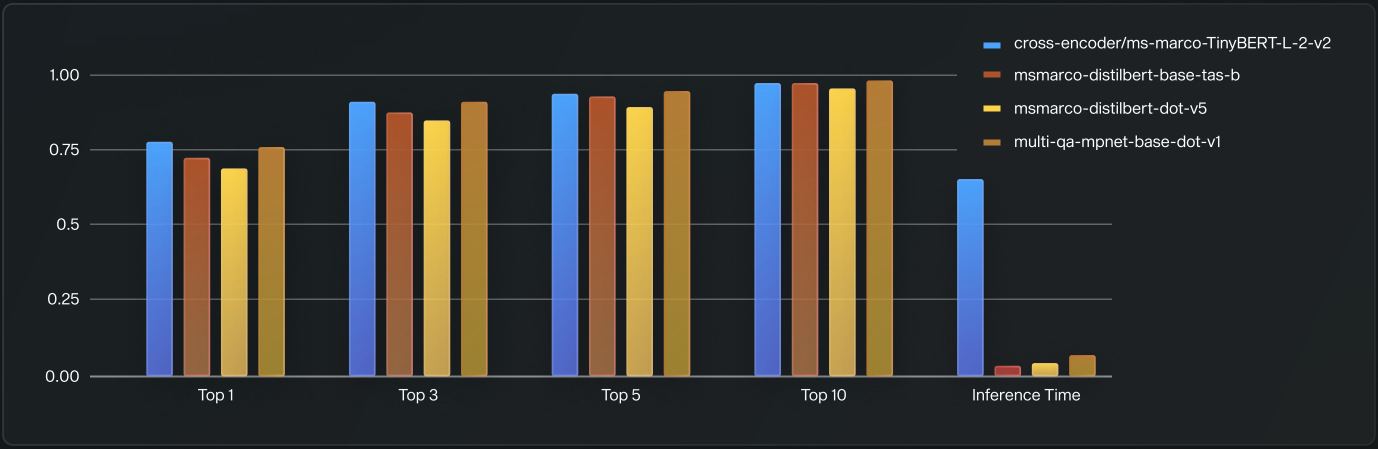


Different Model Performance With Sentence Based Approach

**Tables containing the results for validation fold 4 for all models (comparison b/w passage and phrase)

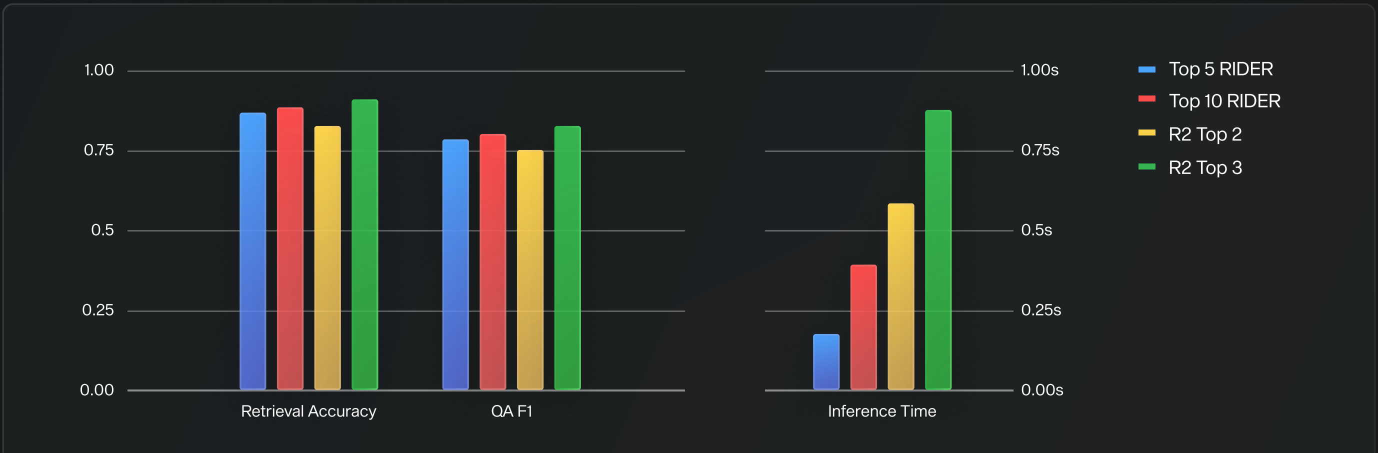
2. Bi-Encoders v/s Cross-Encoders

State-of-the-art Cross-Encoders achieve better performances than Bi-Encoders. However, for many applications, they are not practical due to their inefficiency. They do not produce reusable embeddings which could efficiently be indexed. Overall, our results indicate that the cross-encoder performance boost is not significant enough to compensate for the large inference time. Hence, we decided to move forward with embedding-based approaches.



3. Reader-Guided Passage Reranking

Significantly better QA performance can be achieved when the retrieval results are improved. Usual methods follow the R2(Retriever-Reader) or R3(Retriever-Reranker-Reader) frameworks. **We propose a novel approach (Refer Flowchart 1)** to rerank based on the reader's/QA model's inferences which significantly improves the Top 1 retrieval accuracy and performance. We take a phrase based approach and divide the passages into sentences. Each sentence is then encoded. We narrow down the list of plausible sentences using cosine similarity. Now, to rerank the sentences, we concatenate them into a new paragraph which is then fed in to the reader model. The answer returned by the model is traced back to the original sentence and hence the original passage, hence completing the retrieval and the QA step simultaneously while drastically improving scores and efficiency.



**Tables containing the results for validation fold 4 for all models (comparison b/w passage and phrase)

4. Theme-Based Fine Tuning

Knowledge required to answer questions across themes may vary significantly. Hence, the performance of embedding models is inconsistent across themes. In an ideal situation, we would select a different embedding model for each theme. However, this approach has two major drawbacks: new themes cannot be incorporated easily, and the small sample set in each theme leads to very high variance in results. Hence, we encode the theme names/expressions using suitable Phrase Bert embeddings and cluster these embeddings using state-of-the-art Affinity Propagation Clustering. We choose Affinity Propagation because of its property to choose the number of clusters automatically. Since the new theme would have very few training samples, we can use the samples of the relevant cluster to augment the data for further fine-tuning.

5. Siamese Network-Based Fine Tuning

Siamese networks are often used to extend sentence embeddings from English to new languages. We alter this approach to fine-tune sentence transformers to retrieval tasks. Classically, we pass the embedding of the English sentence to the teacher model and train the student model to get the same embedding for a translated sentence via mean-squared error loss. We modify this approach to pass the query encoding to the teacher model and the paragraph encoding to the student model. This allows the student model to give similar embeddings for queries and passages which can answer those queries, maximizing similarity while retrieving.

Model Name	Top 1 sentence accuracy	Top 5 sentence accuracy	Top 15 sentence accuracy	Top 20 sentence accuracy
msmarco-distilbert-base-tas-b siamese	0.679	0.884	0.934	0.946
msmarco-distilbert-base-tas-b	0.640	0.851	0.922	0.937
multi-qa-mpnet-base-dot-v1 siamese	0.669	0.880	0.948	0.958
multi-qa-mpnet-base-dot-v1	0.617	0.849	0.932	0.946

Question Answering Task

A. Task Description

Extractive question answering is a task in natural language processing (NLP) where the objective is to extract the most pertinent answer to a question from a given document or group of documents. Pretrained models have proven to be highly effective in natural language processing tasks such as question answering. However, one major drawback of these models is the computational time required for inference. For our time-sensitive application where real-time response is required, using larger models may not be feasible.

B. In-Depth approaches with results

1. Pretrained Model Analysis

Models trained on SQuAD2.0 are expected to not only extract the answer from the provided context, but also to determine whether the question is answerable. For the training purpose we divided the given training data into five folds. Each fold contains a different set of themes. The scores in the table below are on the fold four of the training data.

The Limitation of using these pretrained models is that most of them are trained on SQuAD2.0 data so they might not perform well on other themes.

Also, there is still a significant inference time when it comes to answering Questions with large knowledge bases. To mitigate these problems we used the below approach.

2. Static Quantisation

Quantization is a technique to reduce computational and memory costs by representing weights and activations with lower precision data types. We changed the fp32 data type to int8 by converting the model to ONNX format, thus reducing memory requirement and matrix multiplication operation resulting in a reduction of the average latency of vanilla model from 295.10 +\- 6.77 to 224.01 +\- 6.64.

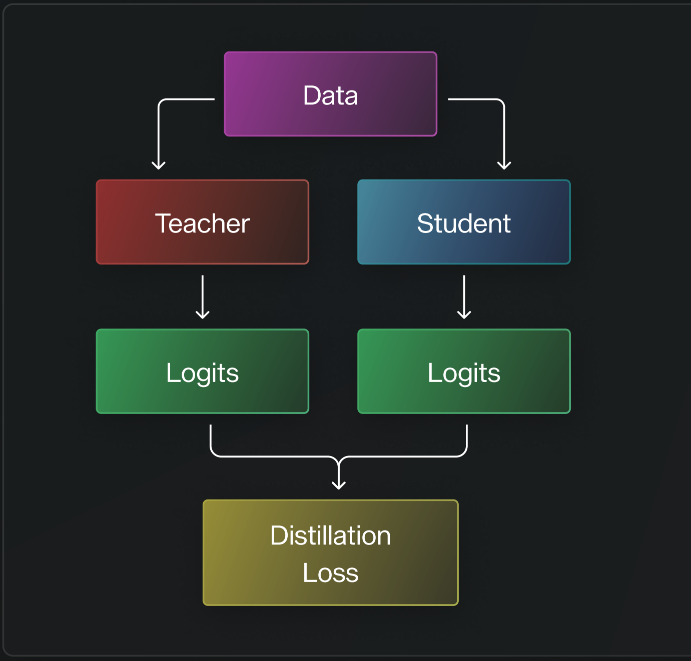
Model Name	F1 Score	EM	Average Inference Time (Per Sample in ms)
deepset/roberta-base-squad2-distilled	91.960	83.324	806.83
deepset/roberta-base-squad2	87.823	79.533	812.69
deepset/tinyroberta-squad2	91.382	82.705	515.10
deepset/deberta-v3-base-squad2	97.038	94.406	1163.18
nlpconnect/roberta-base-squad2-nq	92.634	85.186	762.59
twmkn9/albert-base-v2-squad2	92.596	85.692	874.33
sultan/BioM-ELECTRA-Large-SQuAD2	93.035	85.213	3308.56

**Tables containing the results for validation fold 4 for all models (comparison b/w passage and phrase)

Table 2: Scores on the given training data

3. Response Based Knowledge Distillation

Response-based knowledge focuses on the final output layer of the teacher model. The hypothesis is that the student model will learn to mimic the predictions of the teacher model. We chose this strategy because of the limitations on the given training data as well as the need to make our model more robust to new themes. Based on the results of pre-trained models on the SQuAD2.0 validation data we decided to use tiny roberta as a student model. Since the utilized Teacher Model must be compatible with the student model, we used a Roberta base model trained on both SQuAD2.0 and natural question dataset. The distilled tiny roberta model was validated on trivia qa dataset, increasing the F1 score from 42.9 to 43.9.

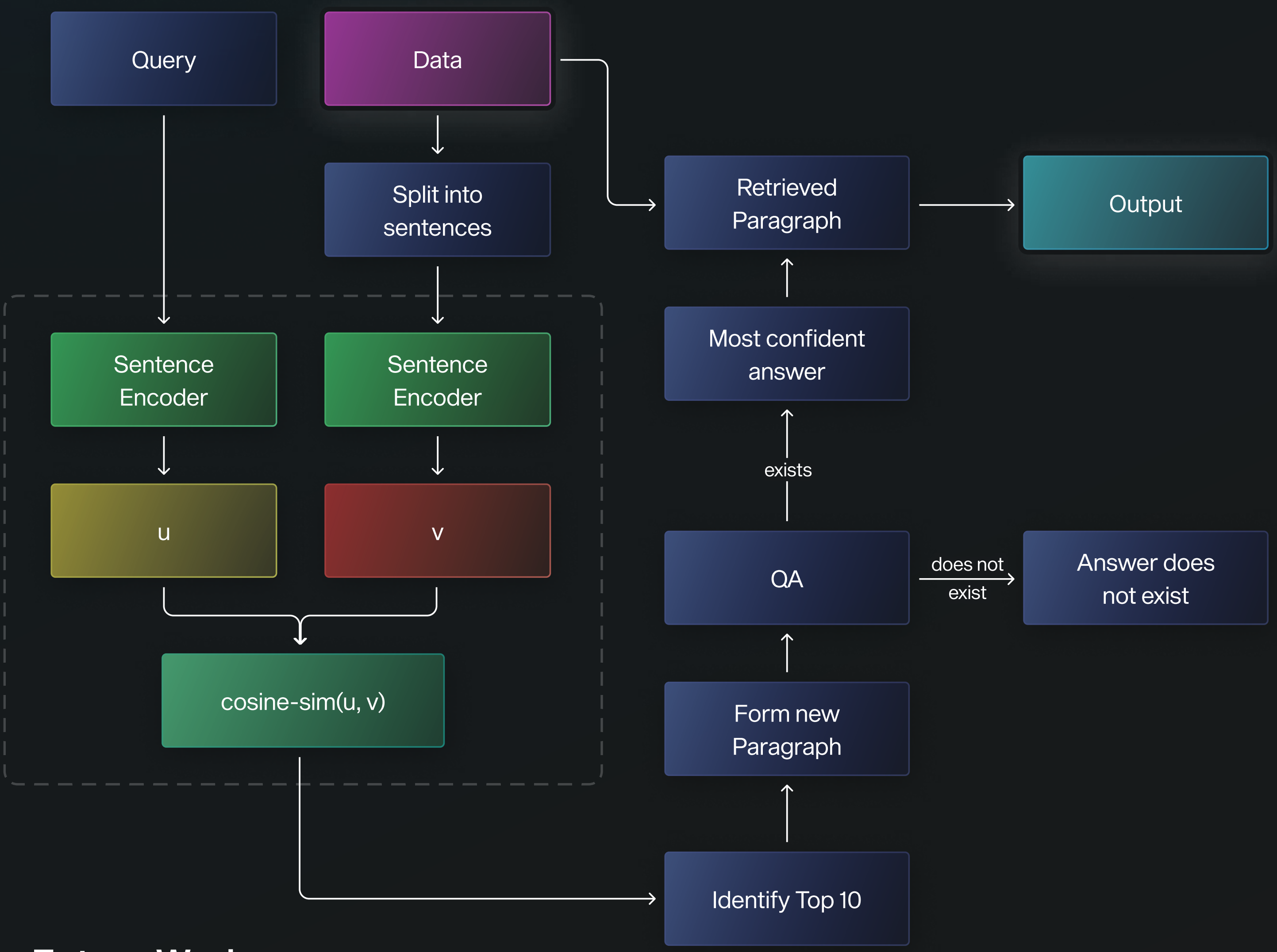


4. Model Pruning & Dynamic Quantisation

Pruning is a technique which focuses on eliminating some of the model weights to reduce the size of the models and thus decreasing the inference requirements. Pruning achieves significant efficiency improvements while minimizing the drop in prediction quality. Pruning involves taking the pre-trained weights, then we pruned the less important connections using magnitude as a proxy for saliency and then retraining the network to finetune weights on the remaining connections. We used magnitude pruning to remove the weights which are very close to zero thus reducing latency of our vanilla model(deepset/tinyroberta-squad2) from 517.82 +\- 8.49 to Optimized & Quantized model Average latency (ms) - 411.73 +\- 56.07.

Pipeline

Flowchart 01



Future Work

Improving Answerability

We plan to use a tree based network to identify negative samples with the use of the features such as reader confidence, probability of start index and end index, various similarity measures such as Cosine similarity, Jaccard similarity, Euclidean distance, Hamming distance scores and 25+ other such metrics.

Exploiting other similarity measures

As of now, we only compare embeddings using cosine measure. We plan to exploit several other similarity measures such as Manhattan Distance, Minkowski Distance,, Jaccard Similarity, Word Mover Distance, to form a more robust better performing measure of similarity between embeddings.

Domain Adaptation

We plan to make use of unsupervised techniques such as autoencoders, and various contrastive learning techniques to improve the retrieval task. We will also try out domain adaptation techniques such as simCSE, CT, TSDAE to adapt text embedding models to the specific text domain without the need to label training data.

Drive Link

Please follow this link for notebooks, data and models (https://drive.google.com/drive/folders/1Qlr504-eh8yPMF6juHJpvaqXPTqO0m1D?usp=sharing)

References

[1] Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading wikipedia to answer open-domain questions. arXiv preprint arXiv:1704.00051

[2] Mao, Y., He, P., Liu, X., Shen, Y., Gao, J., Han, J., & Chen, W. (2020). Generation-augmented retrieval for open-domain question answering. arXiv preprint arXiv:2009.08553.

[3] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[4] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[5] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942

[6] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21(140), 1-67.

[7] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

[8] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

[9] Lee, H., Kedia, A., Lee, J., Paranjape, A., Manning, C. D., & Woo, K. G. (2021). You only need one model for open-domain question answering. arXiv preprint arXiv:2112.07381.

[10] Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., ... & Yih, W. T. (2020). Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906.

[11] Humeau, S., Shuster, K., Lachaux, M. A., & Weston, J. (2019). Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. arXiv preprint arXiv:1905.01969.

References

[12] Khattab, O., Potts, C., & Zaharia, M. (2021). Relevance-guided supervision for openqa with colbert. Transactions of the Association for Computational Linguistics, 9, 929-944.

[13] Lu, Y., Liu, Y., Liu, J., Shi, Y., Huang, Z., Sun, S. F. Y., ... & Wang, H. (2022). ERNIE-Search: Bridging Cross-Encoder with Dual-Encoder via Self On-the-fly Distillation for Dense Passage Retrieval. arXiv preprint arXiv:2205.09153.

[14] Khattab, O., & Zaharia, M. (2020, July). Colbert: Efficient and effective passage search via contextualized late interaction over bert. In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (pp. 39-48).

[15] Mao, Y., He, P., Liu, X., Shen, Y., Gao, J., Han, J., & Chen, W. (2021). Rider: Reader-Guided Passage Reranking for Open-Domain Question Answering. arXiv preprint arXiv:2101.00294.

[16] Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

[17] Wang, K., Reimers, N., & Gurevych, I. (2021). Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning. arXiv preprint arXiv:2104.06979.

[18] Gao, T., Yao, X., & Chen, D. (2021). Simcse: Simple contrastive learning of sentence embeddings. arXiv preprint arXiv:2104.08821.

[19] Janson, S., Gogoulou, E., Ylipää, E., Cuba Gyllensten, A., & Sahlgren, M. (2021). Semantic re-tuning with contrastive tension. In International Conference on Learning Representations, 2021.

[20] McCarley, J. S., Chakravarti, R., & Sil, A. (2019). Structured pruning of a BERT-based question answering model. arXiv preprint arXiv:1910.06360.

[22] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2(7).

[25] Jin, T., Bercea, G. T., Le, T. D., Chen, T., Su, G., Imai, H., ... & Eichenberger, A. E. (2020). Compiling ONNX neural network models using MLIR. arXiv preprint arXiv:2008.08272.

References

[26] Dice, D., & Kogan, A. (2021). Optimizing inference performance of Transformers on CPUs. arXiv preprint arXiv:2102.06621.

[28] Riloff, E., & Thelen, M. (2000). A rule-based question answering system for reading comprehension tests. In ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems.

[29] Rajbhandari, S., Rasley, J., Ruwase, O., & He, Y. (2020, November). Zero: Memory optimizations toward training trillion parameter models. In SC20: International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 1-16). IEEE.

THE END