Enhancing Retrieval-Based Question Answering

Thesis submitted in partial fulfilment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering
by Research

by

Manish Kumar Singh
2020701024
manish.singh@research.iiit.ac.in

International Institute of Information Technology
(Deemed to be University)
Hyderabad - 500032, India
July 2024

International Institute of Information Technology

Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Enhancing Retrieval-Based Question Answering**" by **Manish Kumar Singh**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Dr. Manish Shrivastava

To, my parents and brother,

who were always there by my side and assured me of their everlasting love and support throughout the challenging game called life.

Acknowledgements

I would like to take this opportunity to express my sincere appreciation to all those who supported me during the completion of my research thesis and contributed to making my time at IIIT Hyderabad both memorable and successful.

First and foremost, I extend my heartfelt gratitude to my supervisor, Dr. Manish Shrivastava, for his consistent support throughout my Master's thesis. His guidance and encouragement from our initial discussions enabled me to formulate the problem statement and deepen my understanding of the research area. Dr. Shrivastava's technical expertise and valuable insights significantly shaped the development of this thesis at every step. Furthermore, his personal and ethical perspectives on research will undoubtedly inform my future pursuits.

I am also indebted to my fellow researchers and colleagues, Ashok Urlana, Gopichand Kanumolu, and Lokesh Madasu from LTRC Lab, for their unwavering support and camaraderie during the research process. Their encouragement played a crucial role in the realization of this endeavor. Likewise, I am grateful to my friends, Sanjay, Rishi, and Aashish, whose companionship made this journey unforgettable. I must also acknowledge the contributions of my friends Abhinav, Nikhil, Konda, Bharat, Saideep, and Sumanth, whose presence added joy and excitement to my IT life.

Finally, I express my deepest appreciation to my parents and brother, whose belief in my abilities, unwavering support, and presence during challenging times were instrumental in my success.

Abstract

This thesis delves into the critical domains of retrieval-based question-answering (QA) tasks, particularly in enhancing the quality of retrieved texts. Traditional IR methods often generate noisy text, hindering the accuracy of subsequent answer extraction. While neural models have been employed to re-rank retrieved passages, this work delves deeper into improving retrieval efficiency and precision.

In the open-domain QA domain, where the system tackles any user-posed question, the thesis introduces a novel Passage Ranker model. This model surpasses existing approaches by incorporating local-context information through cross-passage interaction. Unlike prior methods, it leverages the initial ranking provided by search engines and utilizes tailored attention mechanisms for more accurate confidence score calculation. Furthermore, semantic role labeling (SRL) is integrated into the passage reader, enabling it to better grasp contextual semantics. Extensive evaluations demonstrate the significant superiority of this model compared to recent baselines.

For long-context multiple-choice question answering, the thesis proposes Options Aware Dense Retrieval (OADR) as a novel approach. OADR addresses the challenges of reasoning over extensive textual sources by utilizing query-options embeddings. This innovative strategy aims to align with the embeddings of the "Oracle query" (query paired with the correct answer), allowing for better identification of crucial evidence spans essential for accurate answer selection. Experiments on the QuALITY benchmark dataset validate the effectiveness of OADR, showcasing its superior performance and accuracy compared to established baselines.

Contents

List of Figures

Figure Page

List of Tables

Chapter 1

Introduction

In today's information-rich world, the importance of retrieval-based question-answering systems cannot be overstated. As the volume of digital content continues to grow, ranging from web pages to scholarly articles, there's an urgent need for robust methods to effectively sift through this vast sea of information. The ability to accurately and efficiently retrieve relevant information is crucial, not only for academic research and professional applications but also for everyday information needs. Whether it's finding specific details in a large dataset or obtaining concise answers from a broad array of sources, the demand for sophisticated retrieval-based question-answering systems is ever-increasing.

This thesis focuses on enhancing the quality of evidence retrieved in question answering by introducing novel neural architectures. Leveraging advancements in natural language processing and machine learning, we aim to overcome limitations of current systems, such as handling ambiguous queries, understanding context, and filtering irrelevant data. These innovations promise to revolutionize information retrieval, making it more intelligent and useful.

In this chapter, we outline the motivation behind our research, identifying the pressing need for improved retrieval techniques in the face of ever-expanding digital information.

We delve into the key problems we aim to address, such as enhancing retrieval accuracy, increasing the relevance of retrieved results, and improving the overall efficiency of the question-answering process. We also highlight our contributions to the field, including the design and implementation of novel neural architectures that set new benchmarks in retrieval-based question answering. Finally, we provide an overview of the organization of this thesis, guiding the reader through the subsequent chapters that detail our research findings, methodologies, experiments, and conclusions. Through this comprehensive exploration, we aim to demonstrate the transformative impact of our work on the future of information retrieval and question answering systems.

1.1 Motivation

Recent improvements in reading comprehension (RC) systems have renewed interest in open-domain question answering (OpenQA). RC tasks are simpler versions of OpenQA, where models find answers within given contexts. However, practical needs require extracting answers from large collections of articles like Wikipedia. OpenQA systems usually have two parts: a retriever and a reader. Initially, the retriever uses information retrieval (IR) techniques to find relevant passages from a large corpus. Then, the reader extracts the answer from these passages. However, passages retrieved by traditional IR models are noisy and may contain irrelevant information.

In OpenQA, re-ranking is crucial for selecting relevant passages from large text collections. Initial techniques like keyword matching or vector similarity retrieval help, but often miss the most informative passages. So, re-ranking methods are used to prioritize passages likely to contain the correct answer.

Transitioning from OpenQA to the realm of Multiple Choice Question Answering (MCQA), we encounter a task that presents its own set of challenges. Unlike OpenQA, where the model must retrieve and extract an answer from a given context, MCQA involves selecting the correct answer from a set of predefined choices based on the provided context. Despite

this distinction, the core challenge remains: accurately discerning the most relevant information within extensive textual sources to arrive at the correct answer. In long-context scenarios, where the contextual information spans a significant length, this task becomes particularly intricate. Models for natural language understanding are typically limited in their ability to process lengthy passages comprehensively, making effective retrieval and selection of pertinent evidence crucial for accurate answer selection in MCQA.

1.2 Problem Statement

In this thesis, we tackle a multi-fold research problem to improve the answering task.

- **Improve Passage Ranking** We aim to improve the ranking of the passages for the open-domain question-answering tasks. We demonstrate that an improved ranker helps rank passages for higher answer recall with less noise.

- **Enhanced Passage Reader** Reading comprehension is a critical phase in open-domain question answering. We emphasise on enriching the process of identifying answer spans by refining the reader. We demonstrate that capturing the nuanced semantics of passages, such as understanding "Who did what to whom, when, and why," enhances the scoring of answer spans.

- **Improve Retrieval** We aim to enhance sentence retrieval in long-context multiple-choice question answering. We concentrate on leveraging contrastive learning to fine-tune dense retrieval. Our demonstration underscores how this strategy captures subtle relationships between the query and answer options, thereby enhancing evidence identification and elevating the overall performance of multiple-choice question answering.

1.3 Thesis Contributions

Our major contributions towards this thesis are as follows.

Open-Domain QA

- We introduce an innovative approach to ranking passages, integrating both question-passage interaction and cross-passage interaction. This method aims to capture local context information effectively, enabling the ranker model to prioritize passages with higher answer recall while minimizing noise.

- We utilize the initial passage ranking provided by the search engine and implement modified attention within the Passage Ranker model. This enables us to compute a more precise confidence score for each passage.

- Through experimentation, we demonstrate that our model surpasses various baseline models and achieves notable performance improvements across QuasarT, TriviaQA, SearchQA, and OpenSQuAD datasets.

Long-Context MCQA

- We have introduced a new approach to fine-tuning dense retrieval in MCQA, particularly focusing on scenarios where the relevance label of a sentence is absent.

- We improved the dense retrieval process by fine-tuning it using a contrastive dataset we generated. Training the model to align the option-aware query embeddings with those of the oracle query encourages the retrieval model to prioritize evidence spans highly relevant to the correct answer.

- Our experimental findings demonstrate that OADR consistently outperforms existing baselines in both accuracy and retrieval quality on the QuALITY benchmark dataset.

1.4 Organisation of the Thesis

In this section, we outline the structure of our thesis.

Chapter 2 provides a comprehensive review of prior research in Question Answering. We begin by surveying prominent methodologies in open-domain question answering, followed by an exploration of approaches in multiple-choice question answering.

Chapter 3 delves into our innovative passage ranking architecture designed for open-domain question answering. Our central thesis posits that integrating cross-passage attention mechanisms into the ranking process enhances passage prioritization, thereby elevating overall question-answering performance. We commence with an overview of our model, elucidate its constituent components, and subsequently detail our experimental methodology, including dataset selection. This chapter also features a thorough analysis of our system's efficacy, supplemented with illustrative examples.

Chapter 4 introduces a novel technique for refining dense retrieval through a contrastive framework tailored to optimize retrieval efficiency in long-context multiple-choice question-answering scenarios. We present the architecture of our approach, accompanied by the underlying rationale. Additionally, we provide a comprehensive account of our experimental setup, engaging in extensive discussions and analysis to elucidate the system's performance and implications.

In Chapter 5, we present conclusions of our work followed by possible future work in this direction.

Chapter 2

Literature Review

Question answering (QA) poses a significant challenge within the realm of Natural Language Processing (NLP), demanding nuanced approaches to parse and understand complex inquiries. This challenge is multifaceted, requiring distinct methodologies to address diverse forms of questioning, including open-domain inquiries and long-context multiple-choice questions. Despite their differences, both categories necessitate a common starting point: the retrieval of relevant textual information to formulate accurate responses. Thus, the initial phase of QA systems involves the crucial task of identifying and retrieving the most pertinent text segments to address the given query effectively. This chapter critically examines a spectrum of existing works dedicated to retrieval-based question answering, delving into the intricacies of various approaches and techniques employed in this domain. Beginning with an exploration of methods devised to augment open-domain question answering, we unravel the intricacies of retrieval strategies and their efficacy in sourcing relevant information. Subsequently, the discourse transitions to an analysis of methodologies specifically tailored for tackling the challenges posed by multiple-choice question answering, shedding light on innovative techniques devised to navigate through extensive contextual information while making informed selections among provided options. Through

this comprehensive exploration, we aim to elucidate the evolution of retrieval-based QA systems and identify emerging trends and challenges in the field.

2.1 Open-Domain QA

The evolution of machine reading comprehension has been significantly influenced by the advent of large-scale datasets such as SQuAD [1], catalyzing a surge in the accuracy of extracting answers from provided passages. Noteworthy among the trailblazing works in this domain is DrQA [2], which employs a TF-IDF-based retriever to sift through vast corpora, followed by the application of a reading comprehension (RC) model for answer extraction. However, the texts retrieved through this process often suffer from noise, prompting the exploration of various techniques to alleviate this issue. One notable approach involves distant supervision techniques to re-rank the retrieved passages [3]. Others have introduced ranker models aimed at refining the initial passage list, discerning the most relevant passages for answer extraction [4, 5], potentially overlooking valuable insights hidden within other passages.

Acknowledging the inherent limitations of distant supervision, alternative methods such as denoising ranking have been proposed [6]. Furthermore, the integration of a shared-norm objective function [7] has been instrumental in training RC models to filter out negative (no answer) passages, thus mitigating the impact of noisy data. The advent of pre-trained language models like BERT [8] has ushered in a new era for various NLP tasks. Initial investigations leverage BERT for passage re-ranking [9], while others combine Anserini IR toolkit for passage ranking and BERT for answer extraction [10]. BERT has also found utility in both passage ranking and answer extraction [11]. More recently, a novel paradigm has emerged, involving the projection of queries and passages into a shared dense space [12, 13, 14], outperforming traditional TF-IDF and BM25 ranking methods for passage retrieval. Models within this framework, termed dense-retrieval or dual encoder models, exhibit superior performance compared to traditional sparse retrieval methods,

albeit demanding significantly more computational resources during indexing and retrieval. It's noteworthy that dual-encoder models and their variants are not integrated into our baseline.

In contrast to the aforementioned models, our proposed approach enhances ranker performance by modeling passage interaction and harnessing both initial ranking and modified attention mechanisms to mitigate the impact of noisy passages. We introduce a novel training data selection technique for our reader model and integrate semantic role labeling (SRL) to bolster answer accuracy, ultimately amplifying the overall performance of our OpenQA system. Through these innovative enhancements, we aim to address the existing challenges in retrieval-based question answering and push the boundaries of system efficacy and accuracy.

2.2 Long-Context MCQA

MCQA stands as a longstanding and well-recognized research conundrum within the domain of machine reading comprehension, pivotal in gauging the proficiency of automated systems in understanding and responding to complex textual inquiries. At its core, MCQA tasks entail the discernment of the correct answer from a pool of candidate options, contingent upon the context provided alongside the query. Within this landscape, two prevalent paradigms of models have emerged: the Encoder-Only and Encoder-Decoder architectures, each offering distinctive methodologies for tackling the challenge.

In the realm of the Encoder-Only paradigm [8, 15, 16], the interplay among the context C, query Q, and individual options $(O_1, ..., O_n)$ is orchestrated to compute a score for each option. Subsequently, the option garnering the highest score is adjudged as the most plausible answer. Conversely, the Encoder-Decoder approach [17, 18] diverges in its strategy by amalgamating the context, query, and all candidate options into a unified textual representation, treating the identification of the correct answer akin to a generative task where the system is tasked with producing the answer textually.

Despite the proliferation of MCQA datasets [19, 20, 21], it's notable that they predominantly feature contexts of relatively modest lengths, typically containing fewer than 500 tokens. However, the emergence of datasets like QuALITY introduces a distinctive challenge by presenting contexts with an average length extending up to 5,000 tokens, surpassing the capacity constraints of conventional pre-trained language models. To contend with this formidable challenge, a recourse to dense retrievers becomes imperative, aimed at discerning and selecting the most salient sentences from these lengthy contexts, thereby facilitating accurate answer selection amidst the deluge of textual information. This strategic integration of dense retrievers not only serves to streamline the processing pipeline but also underscores the evolving strategies employed to surmount the escalating complexities posed by MCQA tasks within contemporary research endeavors.

Dense Retrieval

In recent times, dense retrieval techniques have demonstrated impressive empirical performance in diverse applications like search and open-domain query answering. These methods leverage the power of pre-trained language models like BERT [8] or RoBERTa [15] to encode the query and context into embeddings, enabling retrieval in a dense representation space. Through effective fine-tuning strategies, dense retrieval approaches have consistently outperformed sparse retrieval methods. However, it is important to note that achieving an effective dense representation space often requires a larger number of relevance labels. While dense retrieval methods have exhibited remarkable performance in scenarios with sufficient labelled data, their effectiveness in few-shot scenarios is not as widely observed [14].

Exploring Fine-tuning in Dense Retrieval

In the expansive domain of open-search scenarios, researchers have been diligently exploring various fine-tuning techniques to elevate the performance of dense retrieval systems [22, 23]. A noteworthy advancement in this pursuit is ConvDR [24], which introduced an additional Contrastive loss to refine the query encoder, enabling it to mirror the embeddings

of manually crafted oracle queries. While these prior methods have made commendable progress in fine-tuning dense retrieval, they have typically relied on some form of supervision. This reliance poses a significant challenge, particularly in the context of long-context Multiple-Choice Question Answering (MCQA), where relevance levels for context sentences are not readily available.

Our approach finds inspiration in the Setfit model [25], which capitalizes on sentence transformers [26] trained on both positive and negative sentences for few-shot classification tasks. This conceptual similarity lays the groundwork for our proposed methodology. However, it's important to highlight that the specific challenge of fine-tuning retrieval in long-context MCQA, especially in the absence of relevance levels for context sentences, remains relatively underexplored.

By tackling this research gap head-on, our objective is to pioneer innovative techniques that enhance the retrieval stage in long-context MCQA without the reliance on explicit relevance levels for context sentences. This endeavor not only presents an exciting opportunity but also holds immense potential to significantly advance the performance of MCQA models, particularly in scenarios where such supervision is scarce or non-existent. Through our research, we aim to catalyze breakthroughs in dense retrieval methodologies, ushering in a new era of efficiency and efficacy in the realm of MCQA systems.

Chapter 3

Open-Domain Question Answering

3.1 Overview

In this Chapter, we focus on improving Open-domain question answering (OpenQA) performance by boosting passage ranker and reader. Primarily OpenQA systems typically consist of two major components, a retriever and a reader [2]. A retriever model first retrieves a relevant set of passages using the information retrieval (IR) technique from the entire corpus (such as Wikipedia), which includes tens of millions of passages. Then, the reader model is applied to extract the answer span from these retrieved passages. The passages retrieved by the traditional IR models are merely relevant to the question and are often noisy.

As shown in Figure 3.1, some negative passages do not contain the answer (passage 4) but are similar to the question. Few passages contain answers but do not answer the question (passage 3), but in distantly supervised setup [27] these noisy passages are considered as valid instances. Since OpenQA relies heavily on efficient passages for better answer prediction, many passage ranker models have been proposed to filter out noisy

passages. Some methods focus on extracting answers solely from the most relevant passage [5, 28, 29], while others utilize multiple passages for answer extraction [2, 6, 30, 7]. One of the earlier works uses BERT for extracting answers from passages [10], while another work uses BERT for both ranking passages and extracting answers from these ranked passages [11]. However, these models encounter two main challenges: (1) their ranking models treat passage relevance as independent probabilities, neglecting cross-passage interaction and local contextual information; (2) they disregard the initial ranking of passages retrieved by search engines, which contain valuable lexical and semantic matching information from the BM25 algorithm.

As shown in Figure 3.1 and in general, positive and relevant passages generally have some similarities since they mainly refer to the same subject. Noisy and negative passages, on the other hand, often have a different theme (dissimilar) from the positive ones. We propose a Passage Ranker that relies on cross-passage interaction to filter out negative and noisy passages. Cross-passage interaction uses a modified attention mechanism that employs a dissimilarity matrix and acts as a gating mechanism that gives more attention to similar passages and less or no attention to dissimilar passages. Moreover, we integrate Semantic Role Labeler (SRL) into the passage reader.

We propose a two-stage framework model. Our Passage Ranker model first filters out noisy passages through multi-level attention mechanisms using initial IR ranking, question-passage and cross-passage interaction. Then, we apply our semantic-aware reader to read each selected passage for answer extraction.

Question: Who was the captain of titanic?
Answer: edward smith
Passage 1: (positive, relevant) the captain of the titanic was edward smith ...
Passage 2: (positive, relevant) in 1895, the majestic was assigned a new captain, an up-and-coming officer named edward smith, who years later would gain lasting fame as the captain of the titanic.
Paggase 3: (positive, noisy) edward smith was one of those who died ...
Passage 4:(negative) the messages are once again relayed to the captain, who is attending a dinner party.

Figure 3.1: A sample of retrieved text for a question

3.2 Proposed Model

In this section, we present our model in detail. Our architecture aims to answer a query based on a set of passages retrieved using IR technique. Formally, given a query $q = \{q^1, q^2, ..., q^{|q|}\}$ and a set of m retrieved passages $P = \{p_1, p_2, ..., p_m\}$ where $p_i = \{p_i^1, p_i^2, ..., p_i^{|p_i|}\}$ is the i^{th} retrieved passage, $|q|$ and $|p_i|$ are the number of words in the query and i^{th} passage respectively. Our model extracts an answer to a given query (q) from a collection of passages P. Figure 3.2 shows the architecture of our model. Our model consists of two primary components.

1. Passage Ranker: The Passage Ranker measures the probability distribution $P_r(p_i|q, P)$ over all retrieved passages. Instead of feeding all passages to Passage Reader, we select only top-K ranked passages using Passage Ranker.

2. Passage Reader: The Passage Reader will compute a score $P_r(a|q, p_i)$ for each possible answer span a in a passage.

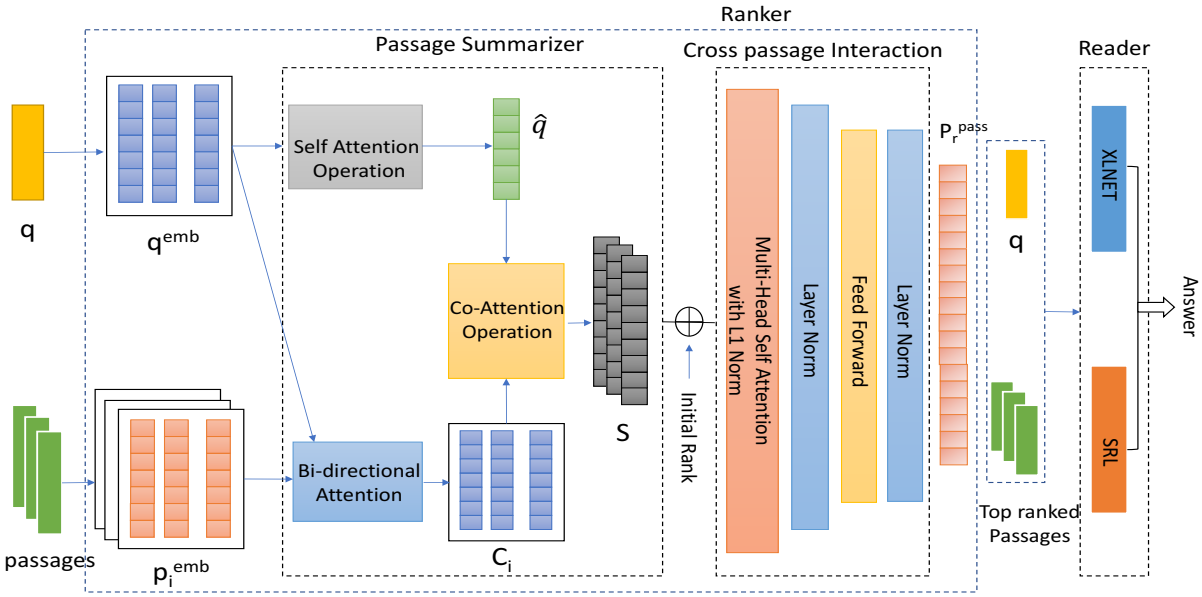

Figure 3.2: Proposed Model Architecture.

Overall, the probability $P_r(a|q, P)$ for extracting the answer a given question q and passage set P can be calculated as follows:

$$P_r(a|q, P) = \sum_{p_i \in P} P_r(p_i|q, P) P_r(a|q, p_i) \tag{3.1}$$

3.2.1 Passage Ranker

We employ a ranker to compute the confidence score of each passage and filter out noisy passages based on the computed score. The principal components of our ranker architecture are as follows.

3.2.1.1 Embedding Layer

We adopted the previous embedding settings used in [7, 30], encoding each word of the query q and a passage p_i into a vector representation by combining the following features:

Word Embeddings: use size 300 pre-trained GloVe [31]word embeddings.

Char Embeddings: encode characters of size 20, which are learnable. Then they are passed to the convolutional layer and max-pooling layer to obtain the embedding of each word.

By concatenating word embeddings and character embeddings and applying bi-directional LSTM to obtain contextual embeddings of both passage and query respectively:

$$p_i^{emb} \in \mathbb{R}^{h \times |p_i|} \tag{3.2}$$

$$q^{emb} \in \mathbb{R}^{h \times |q|} \tag{3.3}$$

3.2.1.2 Query-Aware Passage-Summary Layer:

This layer takes q^{emb} and p_i^{emb} as inputs and creates a new query-aware passage summary vector S_i for each passage. Following BIDAF [32], we first perform bi-directional attention between query q^{emb} and passage p_i^{emb} to obtain H_i for each passage. Then, we pass H_i through the self-attention layer and apply a bi-directional LSTM to obtain the final query-aware passage embedding of the passage p_i:

$$C_i \in \mathbb{R}^{h \times |p_i|} \tag{3.4}$$

we use self-attention operation to obtain the final fixed-size vector representation $\hat{q}$ for the encoded query q^{emb} :

$$\hat{q} = \sum_t \alpha_t q_{:t}^{emb} \in \mathbb{R}^h \tag{3.5}$$

where, α_t encodes the importance of each question word and is calculated as :

$$\alpha_t = \frac{exp(\mathbf{w} q_{:t}^{emb})}{\sum_j exp(\mathbf{w} q_{:j}^{emb})} \tag{3.6}$$

and $\mathbf{w}$ is a weight vector to learn.

Next, we perform the co-attention operation to obtain a fixed-length summary vector for each passage.

$$S_i = \sum_t b_t C_{i:t} \in \mathbb{R}^h \tag{3.7}$$

where b_t is a similarity score between $C_{i:t}$ and $\hat{q}_{:t}$ and is calculated by:

$$b = softmax(\hat{q}^T C_i) \in \mathbb{R}^{|p_i|} \tag{3.8}$$

let S be the sequence of summary vectors for all passages.

$$S = \{S_i\}_{i=1}^n \in \mathbb{R}^{h \times n} \tag{3.9}$$

3.2.1.3 Cross-Passage Interaction Layer:

Cross-Passage Interaction Layer takes $S = \{S_1, S_2, ..., S_n\}$ as input and outputs a representation Z_i for each passage that has the same dimension as S_i. Inspired by the position encoding in the transformer architecture [33], we added the encoding of the passage summary with its respective initial rank encoding. We use an embedding function E to encode each passage's absolute initial rank position into a vector with the same dimension as S_i.

$$R_i = E(rank(S_i)) \in \mathbb{R}^h \tag{3.10}$$

where rank(S_i) denotes the absolute rank position of the passage S_i generated by search engine.

Then the vector of the paragraph summary and the corresponding vector of the initial rank is added.

$$X = [S_1 + R_1, S_2 + R_2, ...S_n + R_n] \tag{3.11}$$

Our Cross-Passage Interaction Layer is a stack of N multi-head self-attention blocks (MSAB) [33] with identical structures. Each MSAB receives a set of N vectors X, which gets processed in a multi-head self-attention layer, followed by a layer normalization, then a feed-forward layer, followed by another layer normalization. Finally, the vector X^1 is the

output, which is input to the next MSAB. In the multi-head self-attention layer, we use a modified attention [34] instead of the traditional self-attention function:

$$A = X(softmax(X^T X) \odot 2sigmoid(D)) \tag{3.12}$$

Here, $A \in R^{h \times n}$, and $\odot$ is element-wise multiplication, and D is a dissimilarity matrix and is computed as:

$$D_{ij} = -||X_{:i} - X_{:j}||_{l_1} \tag{3.13}$$

in this case, l_1 is $L1$ norm and D has same dimension as matrix X.

Since the distance value D_{ij} is always negative, the value of $(2 * sigmoid(D_{ij})) \in [0, 1]$ will saturate towards 0 or 1. 0 if the $L1$ norm is high (dissimilar), 1 if the $L1$ norm is low (similar), and can be interpreted as a kind of gating mechanism. When the dissimilarity between two passages is high, the model learns to erase the positive attention score.

The final representation Z is:

$$Z = MSAB_N(MSAB_{N-1}...(MSAB_1(X)) \tag{3.14}$$

where, $Z \in R^{h \times n}$, and $Z_{:i}$ is final passage representation of i-th passage.

3.2.1.4 Scoring Layer

This layer consists of a linear layer and a softmax function that gives a normalized score for each passage.

$$Pr^{pass} = softmax(W^T Z) \in \mathbb{R}^n \tag{3.15}$$

where W is a trainable weight vector, Pr_i^{pass} represents the probability that the i-th passage contains the answer.

3.2.2 Passage Reader

Passage Reader aims to extract a span of tokens from each passage which is most likely to be the correct answer, i.e., RC. Pre-trained language models based on the transformer's

Model	QuasarT		SearchQA		TriviaQA		OpenSQuAD	
	EM	F1	EM	F1	EM	F1	Em	F1
DrQA	37.7	44.5	41.9	48.7	32.3	38.3	29.8	-
DS-QA	42.2	49.3	58.8	64.5	48.7	56.3	28.7	36.6
R3	35.3	41.7	49.0	55.3	47.3	53.7	29.1	37.5
Re-Ranker	42.3	49.6	57.0	63.2	50.6	53.7	-	-
MSR(BIDAF)	40.6	47.0	56.2	61.4	55.9	61.7	-	-
HAS-QA	43.2	48.9	62.7	68.7	63.6	68.7	-	-
RSQA	48.6	57.8	62.9	69.8	65.2	71.3	-	-
Multi-Passage Bert$_{base}$	51.3	59.0	65.2	70.6	62.0	67.5	51.2	59.0
Ours	56.1	61.4	67.3	71.8	68.3	73.6	54.8	61.4

Table 3.1: Experimental results on four OpenQA datasets: Quasar-T, SearchQA, TriviaQA, OpenSQuAD.

encoder have become the preferred model for RC tasks. XLNet outperforms Bert on several NLP tasks, including RC [35] which motivated us to build our semantic-aware Passage Reader on XLNet. Query q and each passage p_i are concatenated into an input sequence " $q^1, q^2, ..., q^{|q|}$ $[SEP]$ $p_i^1, p_i^2, ..., p_i^{|pi|}$ $[SEP]$ $[CLS]$" and passed through XLNET-base [35] to encode this sequence. Then, the output sub-word representations are transformed into word-level representations via a convolutional layer to obtain the contextual word representation. In parallel, to learn the semantic role of each word/phrase, words in the input sequence are passed through the pre-trained semantic role labeler (SRL) [36] to fetch a predicate-derived semantic embedding. At last, the semantic embedding and the word representation are concatenated to form the final representation of words. Unless specified, we strictly implemented the steps given in [37].

Following the traditional RC model, we use two independent linear layers to predict the start and end positions of the answer span. At training time, we apply the shared-normalization technique [7] to compute the span probabilities, normalizing the span score across all retrieved top-K passages to encourage the model to produce a globally comparable answer score. We train the Passage Reader model on positive and negative passages to achieve better generalization and to restrain the model from generating a high score for a span in a negative passage. For negative passages, we set the start and end positions of the answer span to $[CLS]$ token. During inference, the Passage Reader model is applied to each passage individually to calculate span scores. Then the span with the highest score is selected as the answer from that passage.

3.2.3 Learning and Prediction

Since our model is a two-framework network, both Passage Ranker and Passage Reader are trained separately. Following distantly supervised [27] setup, passages containing ground truth are marked as positive passages, otherwise as negative.

For Ranker, each passage is labelled with $y_i \in \{0, 1\}$. y_i is 1 if a passage contains the ground truth else 0 otherwise. The Ranker model is trained to maximise the log-likelihood of passages containing the ground truth. We use a novel training data sampling technique to efficiently optimise the Passage Reader. Ranker could retrieve many easy negative passages. These easy negative samples can only provide an ineffective supervision signal, so we want to exclude these samples. To achieve this goal, we form a new query by concatenating the question and the ground truth. Moreover, a BM25 ranker[38] is used to rank passages based on this new query, where the top 20 ranked passages are denoted as P_{q+a}. The top 20 ranked passages by our Passage Ranker are denoted as P_{Ran}. We select positive and negative examples only from their intersection $P_{Ran} \cap P_{q+a}$, allowing the Passage Reader to fine-tune only on hard examples. We first train the Passage Reader on data that has randomly selected k passages for each query until convergence is achieved and then perform fine-tuning on the sampled data.

During inference, we use the top-k passages ranked by Passage Ranker and then pass each passage along with the query to the Passage Reader to compute each word's start and end score. We extract the answer a with the highest probability, as shown below:

$$\hat{a} =_a P_r(a|q, P) \tag{3.16}$$

$$\hat{a} =_a \sum_{p_i \in P} P_r(a|q, p_i) P_r(p_i|q, P) \tag{3.17}$$

3.3 Experiments

In this section, we present our experimental setup for evaluating the performance of our system. We discuss the corpora used for training and evaluation and provide details on the implementation details of our approach.

3.3.1 Datasets

We evaluate our model on four Open-domain question answering datasets, Quasar-T [39], unfiltered version of Trivia QA [29], SearchQA [40] and OpenSQuAD [1].

Quasar-T[1]: consists of 43k open-domain trivia query-answer pairs, whose answers obtained from various internet sources. The paragraphs for each question is obtained from ClueWeb09 using Solr search.

Trivia QA[2]: includes 95k open domain query-answer pairs. We focus on open-domain setting that contains unfiltered documents. Following[30], we randomly hold out 5,000 QA pairs from the original training set as our validation set, and take the remaining pairs as our new training set.

SearchQA[3]: has more than 140k query-answer pairs based on Jeopardy! questions and collects about top 50 web page snippets form google search engine for each question.

OpenSQuAD: query-answer pairs are from SQuAD 1.1 but the OpenQA model will find an answer from the entire Wikipedia rather than a given context. Original dev set is used as test set.

For SQuAD, we use the 2016-12-21 dump of English Wikipedia as our knowledge source. We apply the pre-processing code released in [2] to extract the clean text portion of the articles from the Wikipedia dump. This step removes semi-structured data such as tables, info-boxes, lists, and disambiguation pages. Then, Anserini [38] is used to retrieve the top 100 passages for each query.

3.3.2 Baseline

For the comparison, we select several public models as baselines, including (1) DrQA [2] uses a TF-IDF based retrieval to find relavant passages and then apply a RC to each passage to find answer.(2) DS-QA [6], a multi-paragraph model for extracting answer. (3)

[1] https://github.com/bdhingra/quasar
[2] https://nlp.cs.washington.edu/triviaqa/
[3] https://github.com/nyu-dl/SearchQA

Question: who was the first fallen angel ? **Answer:** lucifer
Passage selected by MSR ranker **P1: (negative)** as for first of the fallen , that is constantine and not something i 'm familiar with ." **P2: (positive,relevant)**lucifer is the well known first fallen angel who was exiled for rebelling against god and was put out from heaven **P3:(positive,irrelevant)** Lucifer was the first angel to have sinned **P4:(positive,relevent)** Lucifer was the first fallen angel sandalphon who told elaine that easterman was not her biological father **P5:(negative)** in christianity , fallen angel refers to the angel either banished or exiled from the heaven
Passages selected by Passage Ranker: **P1: (positive,relevant)** Lucifer was the first fallen angel sandalphon who told elaine that easterman was not her biological father **P2: (positive,relevant)** however , it was not until the new testament that satan was portrayed as lucifer , the first of the fallen angels to rebel against', **P3: (positive,relevant)** another famous arch angel is lucifer , who became the fallen angel known as satan **P4: (positive,relevant)**lucifer is the well known first fallen angel who was exiled for rebelling against god and was put out from heaven **P5: (positive,relevant)** unfortunately , malak ta'us is equated with lucifer " the first fallen angel

Table 3.2: The examples query from Quasar-T test set with top-5 passages returned by the two rankers.

R3 [28], a Reinforced Ranker-Reader model that ranks the retrieved passages and assigns different weight to passages prior to pocessing by the reader. (4) Re-Ranker [41], an answer re-ranking model. (5)MSR(BIDAF) [42], a multi-step retriever-reader model with bi-directional attention flow network. (6)DECAPROP [43], a densely connected neural architecture for reading comprehension. (7)RSQA [44], a model based on ranking paragraphs and sampling strategies. (8) HAS-QA [30] an end-to-end model that uses multiple answer span present in a context. (9) Multi-Passage BERT [11], a BERT-based ranker and reader model.

Model	QuasarT		
	Top-1	Top-3	Top-5
IR	21.9	38.5	47.6
R3	40.3	51.3	54.5
DS-QA	27.7	36.8	42.6
FSE	42.8	55.2	58.9
RSQA	49.9	58.8	62.6
MSR	42.9	55.5	59.3
Passage ranker	56.3	63.6	67.4

Table 3.3: Performance of Passage Ranker on Quasar-T.

3.3.3 Implementation Details

Our model uses the same data preprocessing[4] presented in [7]. For the ranker, we use the 300-dimensional pre-trained word embeddings from GloVe[5] [31] and do not fine-tune them in the training step. Additionally, 20-dimensional character embeddings are left as

[4] https://github.com/allenai/document-qa
[5] https://github.com/stanfordnlp/GloVe

learnable parameters. We set the hidden size of the LSTM to 128, the number of LSTM layers to 1, the dropout ratio to 0.2, and the batch size to 8. To optimize the model, we use Adam [45] optimizer with a learning rate of 5e-4. The number of stacked MSAB is set to 4, and each block has 128 hidden units and 8 heads. For the reader, we use the pre-trained XLNET base model with default hyperparameters. We use pre-trained SRL of [36] to obtain the semantic labels.

3.4 Results and Analysis

In this section, we compare the performance of our system against various OpenQA baselines using Exact Match(EM) and F1 scores as evaluation metrics. EM measures the percentage of documents where the predicted answer is identical to the correct answer. For each question, if the characters of the answer predicted by the model exactly match the characters of any of the ground truths, EM = 1, otherwise EM = 0. The F1 score captures the precision and recall that the words selected as part of the answer are actually part of the ground truths. We also attempt to analyze the performance of Passage Ranker, Passage Reader, and the effect of the number of ranked passages for extracting an answer by presenting empirical evidence in the form of tables, graphs, and examples. Finally, we conclude this section by performing an ablation study to investigate the contribution of each proposed method in our model.

3.4.1 Overall Result

Table 3.1 shows the performance of our model and the baseline models. As shown, our model outperforms the baseline models in all four datasets. The SQuAD1.1 test set is not publicly available, and an unfiltered version of TriviaQA does not provide answers for the test set either. Therefore, we follow previous work and report the results on their development sets. In particular, for Quasar-T, our model improves by 7.5 % EM over the non-BERT and 4.8 % EM over the BERT-based model. On average, for each query in

Question	Naive Reader	Passage Reader
Who succeeded winston churchill as prime minister of england ?	sir Anthony	anthony eden
In the song the 12 days of Christmas what did my true love give on the 5th day?	gold rings	5 gold rings
Pat sullivan created which cartoon character?	Felix	felix the cat
Under what structure was the first nuclear reactor built in chicago?	stands of an old stadium	football stadium
what kind of creature is a centaur	member-of elftown	half man, and half horse
Who ordered the building of the tower of london ?	william	william the conqueror

Table 3.4: Examples from Quasar-T test set to compare answers from a naive version of Passage Reader and Passage Reader. In these examples answers from Passage Reader are same as the ground truth answers.

the Quasar-T development set, only 13.4% of the passages contain ground truth, much lower than TriviaQA (28.7%) and SearchQA (35.2%). Since Passage Ranker is better at filtering out negative and noisy passages, improvements are highest in Quasar-T and lowest in SearchQA.

3.4.2 Passage Ranker Performance

To take a deeper look at the contribution of Passage Ranker in filtering out these noisy passages, we compare Passage Ranker with traditional information retrieval (IR) and recent OpenQA models that have a ranking sub-module to select relevant passages, R3 [28], DS-QA [6], FSE [46], RSQA [44], MSR [42]. We compare the performance of selecting an informative passage by Passage Ranker with others by evaluating whether the ground truth appears in the top-k ranked passages and presenting an example of the top-5 ranked passages. The result is shown in Table-3.3 for the Quasar-T dataset. From the table, it can be seen that:
(1) Passage Ranker significantly outperforms the traditional IR model in selecting informative passages, indicating that our Ranker is able to capture the semantic correlation between question and passages
(2) Passage Ranker also performs better than neural network-based rankers (R3, DS-QA, MSR, RSQA, FSE). It can be seen that modelling the interaction between passages, integrating the initial ranking, and using the modified attention improves the performance of the Ranker.

An example of the top 5 ranked passages by Passage Ranker and MSR Ranker from Quasar-T is shown in Table-3.2. For the question "Who was the first fallen angel?" all of the top 5 passages retrieved by our Ranker are relevant and support the ground truth answer ("Lucifer"). However, two of the passages identified by the MSR ranker do not contain a ground truth (P1, P5), although they are somewhat similar to the question. P3 contains ground truth but does not answer the question. The MSR reader computes the passage score of the passages based on the question-passage relation. However, our Ranker uses both question-passage and cross-passage interaction and also uses the initial ranking and modified attention to retrieve more relevant passages for the Reader to extract the answer.

3.4.3 Passage Reader Performance

We evaluated the performance of Passage Reader by comparing it to a naive version of Passage Reader that omits the integration of SRL. The sub-word representation generated from the XLNET-base model is passed directly through a linear layer for answer's span prediction. Table-3.4 shows a list of queries with predicted answers from Passage Reader and Naive Reader. Many of Naive Reader's answers deviate from the ground truths by only a word or two, which mainly affects the EM points. The answers extracted by Passage Reader match the ground truth answers, indicating that Passage Reader can extract semantically more accurate answers compared to Naive Reader. This shows that integrating contextual semantics is quite helpful in predicting meaningful answers by guiding the model to learn the semantic role of words/phrases in sentences, such as *who did what to whom, when, and why.* Intuitively, this suggests that the Passage Reader evolves from sub-word representation to word-level and final semantic representation.

3.4.4 Effect of Passage Number

We investigated the impact of passage selection on answer prediction by comparing our model's performance with passages chosen by both our Passage Ranker and a traditional information retrieval (IR) model. Figure 3.3 depicts how our model's performance changes based on the size of the search space, represented by the number of considered passages. Initially, both approaches exhibit performance improvement as the number of passages increases, eventually reaching a plateau. Notably, the Passage Ranker achieves peak performance earlier and surpasses that of the IR model. This suggests that our model effectively identifies the most relevant passages for accurate answer prediction, even with a limited search space, highlighting the efficiency of the Passage Ranker.

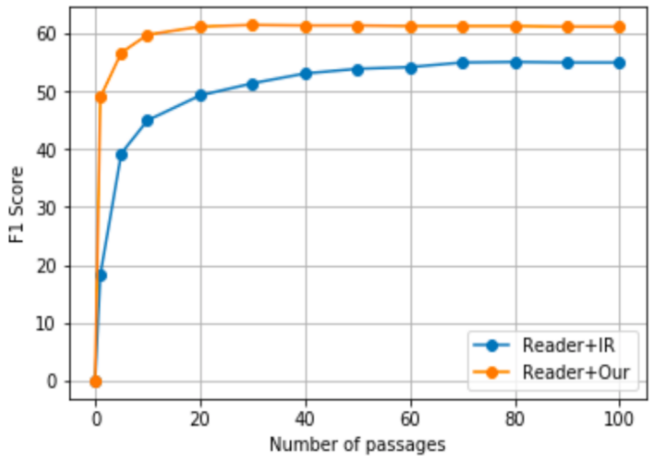

Figure 3.3: Performance with different number of top passages on Quasar-T

3.4.5 Ablation Study

We conducted an ablation study on the Open-SQuAD development set to assess the contribution of individual components within our model. Table 3.5 summarizes the results of five ablated baselines compared to our full model.

The first ablation removes the integration of semantic role labeling (SRL) from the Passage Reader, highlighting its significant impact on performance. Ablation (3) replaces the modified attention mechanism with standard self-attention for cross-passage interaction, demonstrating the importance of this specific attention technique. Similarly, ablation (4) removes the initial rank integration during cross-passage interaction, showcasing its contribution to the overall effectiveness.

Ablation (5) excludes the cross-passage (C-P) interaction layer from the ranker, demonstrating that solely relying on question-passage integration for ranking passages leads to decreased performance. Finally, ablation (6) substitutes our ranker with a traditional information retrieval (IR) model, resulting in a 7.2

Models	EM	F1
Full Model	54.8	61.4
(1) w/o SRL	53.6	60.8
(2) w/o training on sampled data	52.9	60.4
(3) w/o modified attention	52.2	60.0
(4) w/o initial rank	51.5	59.6
(5) w/o C-P interaction	50.2	58.1
(6)w/o ranker	47.6	55.3

Table 3.5: Ablation study on Open-SQuAD dev set.

3.5 Conclusion

We introduced a pipeline for open domain query answering and introduced a Passage Ranker that accounts for both query-passage and cross-passage relevance. By integrating initial rank and employing modified attention, our ranker produces more precise confidence scores for each passage. Our approach substantially enhances answer recall by filtering out irrelevant passages. Using a sematic-aware reader further demonstrates the effectiveness of incorporating explicit contextual semantics. Experimental findings across four OpenQA datasets indicate our model's superior performance compared to all baseline methods.

Chapter 4

Options-Aware Retrieval For Long-Context MCQA

4.1 Overview

Long-context multiple-choice question-answering tasks require robust reasoning over extensive text sources. Since most of the pre-trained transformer models are restricted to processing only a few hundred words at a time, successful completion of such tasks often relies on the identification of evidence spans, such as sentences, that provide supporting evidence for selecting the correct answer. Prior research in this domain has predominantly utilized pre-trained dense retrieval models, given the absence of supervision to fine-tune the retrieval process. We propose a novel method called Options Aware Dense Retrieval (OADR) to address these challenges. ORDA uses an innovative approach to fine-tuning retrieval by leveraging query-options embeddings, which aim to mimic the embeddings of the oracle query (i.e., the query paired with the correct answer) for enhanced identification of supporting evidence.

4.2 METHODOLOGY

To address the long-context MCQA task, we employ a two-step system that includes an extraction step followed by a QA model. The input consists of a query Q, a background context $C =< s_1, ..., s_M >$ containing M sentences, and a set of n options $O_1, O_2, ..., O_n$ with only one option being the correct answer. In the extraction step, we identify a set of K evidence spans from the context C. These selected sentences are then sorted based on their original order in the passage and concatenated to form a 'passage' specific to that query. In the second step, we use a standard Pre-trained language model for multiple-choice question answering to determine the correct answer from the given options. Previous studies have found that using oracle query leads to improved retrieval of relevant evidence spans for accurate answer selection. In OADR, we aim to fine-tune the retrieval to ensure that embeddings of options-aware query closely mimic the embeddings of the oracle query. By doing so, we enhance the effectiveness of the retrieval stage and increase the likelihood of identifying the correct answer. The proposed method is depicted in the block diagram shown in Figure 4.1. In the subsequent section, we discuss the steps involved in the method, explaining how each component contributes to the overall approach.

4.2.1 Options-Aware Dense Retrieval

To enable dense retrieval in our approach, we leverage Sentence Transformers, which are modified versions of pre-trained transformer models. These models employ Siamese and triplet network architectures to generate embeddings for sentences, capturing their semantic meaning. The goal is to minimize the distance between pairs of sentences with similar meanings while maximizing the distance between pairs that are semantically dissimilar. By doing so, Sentence Transformer produce dense and fixed vectors that effectively represent the meaning of each sentence or paragraph. Since relevance levels for context sentences are unavailable, we fine-tuned the Sentence Transformers using a contrastive, Siamese approach on a newly generated dataset.

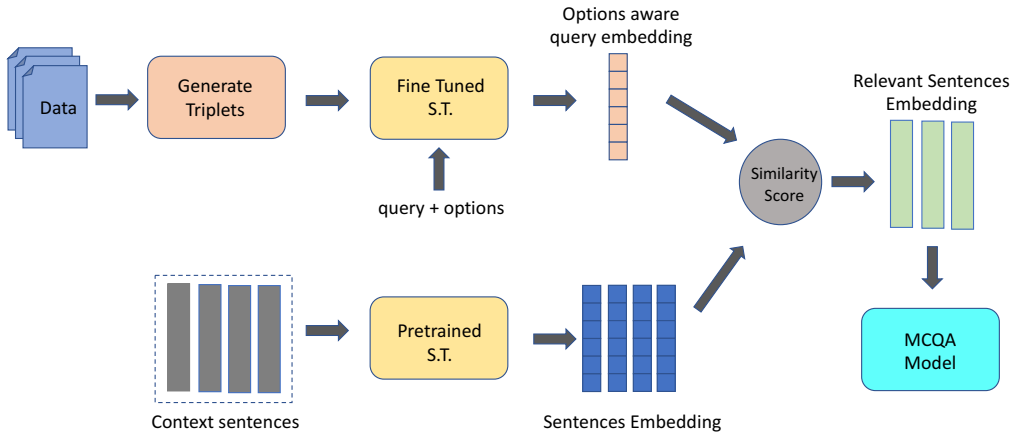

Figure 4.1: Architecture of option-aware retrieval

Fine Tuning Dense Retrieval

We employ a contrastive training approach to achieve embeddings that closely resemble the oracle query. The first step is to generate a contrastive fine-tuning dataset. For each sample in the dataset, which includes a query Q, options $O_1, ..., O_n$, and the correct answer O_1, we create a triplet $(Anchor, Positive, Negative)$. In this triplet, the $Anchor$ is an oracle query $(Q + O_1)$. The $Positive$ component is obtained by concatenating the query with all options $(Q + O_1 + O_2 + O_3 + O_4)$. On the other hand, the $Negative$ component is created by concatenating the query with all wrong options $(Q + O_2 + O_3 + O_4)$, assuming that O_1 is the correct answer. To fine-tune the Sentence Transformer model, we utilize the generated dataset and apply TripletLoss during training. TripletLoss aims to minimize the distance between the Anchor and the Positive sentences while simultaneously maximizing the distance between the Anchor and the Negative sentences. It computes the following loss function.

$$Loss = max(||AnchorPositive|| - ||AnchorNegative|| + margin, 0) \tag{4.1}$$

where the margin is a hyperparameter which ensures that the *Negative* sample is at least this much further away from the *Anchor* than the *Positive* sample in the embedding space. This constraint helps to maintain a clear separation between relevant and irrelevant options during training. The margin value is chosen carefully to strike a balance between maximizing the distinction between positive and negative examples while avoiding excessive separation that may lead to the loss of important information.

This training objective encourages the Sentence Transformer model to learn embeddings that effectively capture the similarities and differences between relevant and irrelevant options for a given query.

4.2.2 MCQA Model

In our approach for the MCQA task, we employ various pre-trained language models to choose the correct answer based on the given query, context, and options. However, instead of inputting the entire context into the MCQA model or truncating it, we leverage the fine-tuned OADR to extract the most relevant sentences from the context which enhances the model's ability to identify the correct answer. This approach proves effective in handling long-context scenarios where traditional methods may face limitations. As a result, we can utilize a range of high-performing short-sequence encoder models like RoBERTa [15], DeBERTaV3 [16], as well as encoder-decoder models like T5 [18] and BART [17] as our MCQA model. However, it is important to note that this approach may lead to the exclusion of some parts of the input context, potentially affecting the model's understanding of the overall context.

Inference During the inference phase, we follow a specific procedure. Firstly, the options-aware query, which combines the query with each option, is passed through the fine-tuned sentence transformer to obtain the options-aware query embedding. Next, the context sentences are individually processed by a pre-trained sentence transformer, resulting in dense representations for each sentence. To select the most relevant sentences from

Model	**QUALITY** Accuracy	**RACE →QuALITY** Accuracy
Longformer-base	33.7 / 32.6	38.1 / 32.8
LED-base	25.1 / 24.3	38.1 / 32.8
LED-large	25.1 / 25.6	35.6 / 32.0
DPR → RoBERTa-base	40.0 / 36.4	43.8 / 37.2
DPR → RoBERTa-large	26.7 / 24.0	50.8 / 46.2
DPR → DeBERTaV3-base	41.8 / 37.4	46.7 / 40.9
DPR → DeBERTaV3-large	45.1 / 39.2	53.6 / 47.4
ST(pre-train) → RoBERTa-base	39.4 / 35.3	43.1 / 37.4
ST(pre-train) → DeBERTaV3-base	40.6 / 38.1	44.6 / 39.5
Col-Bert → DeBERTaV3-base	42.3 / 37.6	49.2 / 43.4
OADR → RoBERTa-base	43.5 / 36.8	47.2 / 41.4
OADR → RoBERTa-large	32.4 / 27.7	53.7 / 46.8
OADR → DeBERTaV3-base	44.1 / 38.5	51.6 / 43.2
OADR → DeBERTaV3-large	49.2 / 42.4	59.3 / 48.9

Table 4.1: Accuracy on full QUALITY test set and QUALITY-Hard subset(full/Hard). In RACE → QUALITY model is trained on RACE dataset than on QUALITY dataset

the context, we compute negative Euclidean (L2) distance scores between the options-aware query embedding and the embeddings of the context sentences. After identifying the relevant sentences, we sort them based on their original order and concatenate them to form a coherent passage. The length of the passage is limited to a maximum of 300 tokens, ensuring concise and manageable input for downstream tasks. Finally, this constructed passage is fed into the trained MCQA model, which takes into account the options and the context to determine the most suitable answer among the given options.

4.3 Experiments

In this section, we cover three main aspects: the corpora used for training and evaluation, the comparison baselines, and our approach's implementation details.

4.3.1 Dataset

To assess the effectiveness of our approach, we conducted experiments using the QuALITY benchmark, which specifically focuses on long-context MCQA. The benchmark comprises two subsets, namely QuALITY-EASY and QuALITY-HARD, designed to represent different levels of query difficulty. During the annotation process, the evaluators were given 45 seconds to access the text to select the correct answer. This limited timeframe simulated a scenario where annotators had to skim or search for relevant phrases to respond. If a query proved unanswerable within this time constraint but was answerable under normal conditions, it indicated a higher level of query difficulty. The average length of the context passages in the benchmark is approximately 5,159 tokens, and the training set consists of 2,523 queries. As the correct answers in the official test set are not publicly available, we randomly selected half of their official development set to serve as our test set. Therefore, both our development and test sets consisted of 1,043 samples each.

Additional Training Data To enhance our training process, we incorporated additional training data from the RACE [19] dataset. The RACE dataset comprises multiple-

choice queries that are based on passages. While the passages in RACE are generally shorter, the dataset offers a significant number of queries, totalling around 88,000. This presents an advantageous opportunity for knowledge transfer, allowing us to enrich our QuALITY model. For both the training of the OADR and the MCQA model, we utilized the entire RACE dataset, which includes queries from both middle-school and high-school levels.

4.3.2 Baseline

Our comparison includes two sets of baselines: those from published research and our own implementations. These baselines consist of (1) the Longformer [47] encoder model, which can handle up to 4,096 tokens and is well-suited for capturing most of the context, (2) the Longformer Encoder-Decoder (LED) model, which extends the encoder input token limit to 16,000, allowing for a broader understanding of the context, and (3) the DPR [12] model trained for open-domain retrieval, which extracts relevant sentences and uses an encoder model for answer selection. In addition to these published methods, we incorporated a pre-trained Sentence Transformer to extract relevant sentences. We then employed an encoder-based MCQA model to determine the answer. It's important to note that the queries provided to these models are in their raw form.

4.3.3 Implementation Details

For our retrieval process, we employed the "multi-qa-mpnet-base-dot-v1" model as the sentence transformer. To fine-tune this model, we utilized the TripletLoss function with a learning rate of 1e-4. The fine-tuning process involved using a batch size of 8 and a maximum sequence length of 128 tokens. The training was performed for a single epoch. For selecting the correct answer, we used RoBERTa and DeBERTaV3 as our MCQA models. To fine-tune the encoder MCQA model, we followed the settings outlined in [4], unless

stated otherwise. The learning rate for the encoder MCQA model was set to 5e-4. For the base-model, we used a batch size of 4, while for the large-model, we used a batch size of 2.

Models	EM
Q	53.6
OAQ	50.3
$(OAQ)_R$	61.4
OAQ_{RQ}	62.1

Table 4.2: Average % over-lap of retrieved sentences with sentences retrieved sentences from OQ on dev-set

4.4 Results and Analysis

This section compares our system's performance against different baselines, employing accuracy as the primary evaluation metric. Additionally, we delve into an analysis of the learned options-aware query embeddings, providing empirical evidence in tables and plots.

4.4.1 Overall Result

Table 4.1 presents the performance comparison between our OADR model and various baseline models. The results for the baseline models are obtained from published research [48]. It is worth noting that the results reported in the table are based on the dev set, as our test set is derived from the same dev set by randomly selecting half of the samples. The table clearly indicates that the OADR with DeBERTaV3-large model, both initially fine-tuned on the RACE dataset and further fine-tuned on the QuALITY dataset, achieves the highest performance among all the baseline models examined. When comparing models trained with different datasets, it becomes evident that the RACE dataset followed by the

QuALITY dataset (RACE → QuALITY) yields significantly better performance compared to models trained solely on the QuALITY dataset. This performance improvement can be attributed to the limited size of the QuALITY training set. The results strongly suggest that knowledge transfer from the RACE dataset is beneficial, enabling the model to learn from a larger and more diverse set of training examples.

4.4.2 Learned Options-Aware Query Representation

In this section, we analyse the learned options-aware query embedding by comparing it with other query embeddings. To assess whether OADR can mimic the embeddings of the oracle query, we chose a specific example from the development set. In this example, the query is "Who is The Pooch?" and the available choices are ['The family dog.', 'Dick and Eleanor's child.', 'Grampaw Moseley's alter-ego.', 'Mom and Pop's youngest child.']. To visualize and compare the different types of query embeddings, we employed t-SNE [49] and generated plots as shown in Figure 4.2. Where

- Q: Query embedding from pre-trained Sentence Transformer.
- OQ: Oracle query embedding from pre-trained Sentence Transformer.
- OAQ: Options-aware query embedding from pre-trained Sentence Transformer.
- OAQ_R: Options-aware query embedding from OADR fine-tuned on RACE.
- OAQ_{RQ}: Options-aware query embedding from OADR fine-tuned on RACE then on QuALITY.

Upon examining the figure4.2, we observe that the options-aware query embedding (OAQ) appears to be the farthest from the oracle query embedding (OQ). In contrast, the fine-tuned options-aware query embeddings (OAQ_R and OAQ_{RQ}) are positioned closest to the oracle query embedding. This observation strongly suggests that OADR is capable of effectively mimicking the embeddings of the oracle query.

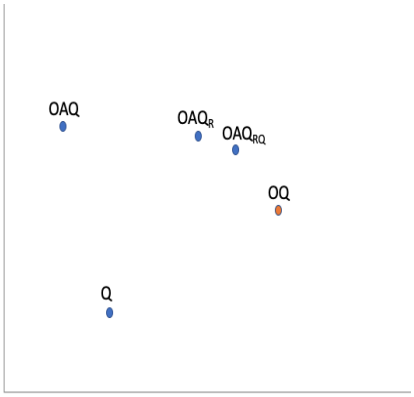

Figure 4.2: t-SNE plot for different queries

We considered the sentences retrieved from the oracle query as the upper limit of retrieval performance, and we calculated the percentage overlap of sentences obtained from different query embeddings. As shown in Table 4.2, concatenating all the options with the query and using it for retrieval leads to a degradation in performance. However, there is a significant increase in overlap when using the options-aware query embedding derived from OADR fine-tuned on RACE. This suggests that OADR prioritizes evidence sentences that are highly relevant to the correct answer.

4.5 Conclusion

This work introduces Options Aware Dense Retrieval, a novel approach designed to improve Multiple Choice Question Answering performance when dealing with extensive contextual information. OADR leverages a contrastive learning strategy during dense retrieval, incorporating query-options embeddings to pinpoint evidence spans directly associated with the correct answer choice. Our evaluation on the QuALITY benchmark dataset demonstrates the effectiveness of OADR, showcasing its superior accuracy and retrieval quality compared to established baseline methods.

Chapter 5

Conclusion

This thesis introduces two novel approaches that significantly enhance retrieval-based question-answering systems in open domains and multiple-choice settings.

Chapter 3 presents a Passage Ranker designed to improve the relevance of retrieved passages. It leverages both query-passage and cross-passage relationships to filter out irrelevant passages, leading to improved answer recall. The ranker incorporates initial rank information and utilizes modified attention mechanisms for more accurate confidence score calculation for each passage. Subsequent processing with a semantic-aware reader identifies the most accurate answer spans within the retrieved passages. Extensive evaluations on four OpenQA datasets demonstrate the superiority of our model compared to existing baselines.

Chapter 4 tackles the challenge of fine-tuning dense retrieval for Multiple-Choice Question Answering (MCQA), where sentence relevance is crucial. We propose Options Aware Dense Retrieval (OADR), which enhances retrieved text in long-context MCQA scenarios. OADR leverages a contrastive learning strategy during dense retrieval, incorporating query-options embeddings to prioritize evidence spans directly associated with the correct answer choice. Our experiments on the QuALITY benchmark dataset validate OADR's ef-

fectiveness, showcasing its superior accuracy and retrieval quality compared to established baselines.

Looking ahead, several promising avenues for future research exist. Firstly, in OpenQA, exploring the possibility of multiple answer spans within a single passage could further improve model performance by enhancing precision and robustness. Additionally, integrating background knowledge, including factual knowledge and common sense, holds promise for further performance gains across various tasks. Furthermore, OADR can be further refined by exploring different contrastive dataset constructions and integrating additional contextual information to potentially achieve even higher retrieval accuracy. By pursuing these directions, we aim to continue pushing the boundaries of question-answering systems in open domains and long-context scenarios.

Publications

Relevant Publications

1. Manish Singh and Manish Shrivastava. "BRR-QA: Boosting Ranking and Reading in Open-Domain Question Answering". Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD).

2. Manish Singh and Manish Shrivastava. "Options-Aware Dense Retrieval for Multiple-Choice query Answering" submitted at ArXiv.

Other Publications

1. Manish Singh and Manish Shrivastava, "Fusion of Intrinsic & Extrinsic Sentential Traits for Text Coherence Assessment". Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)

Bibliography

[1] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[2] D. Chen, A. Fisch, J. Weston, and A. Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

[3] P. M. Htut, S. R. Bowman, and K. Cho. Training a ranking function for open-domain question answering. *arXiv preprint arXiv:1804.04264*, 2018.

[4] S. Wang, M. Yu, X. Guo, Z. Wang, T. Klinger, W. Zhang, S. Chang, G. Tesauro, B. Zhou, and J. Jiang. R 3: Reinforced ranker-reader for open-domain question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[5] E. Choi, D. Hewlett, J. Uszkoreit, I. Polosukhin, A. Lacoste, and J. Berant. Coarse-to-fine question answering for long documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 209–220, 2017.

[6] Y. Lin, H. Ji, Z. Liu, and M. Sun. Denoising distantly supervised open-domain question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1736–1745, 2018.

[7] C. Clark and M. Gardner. Simple and effective multi-paragraph reading comprehension. *arXiv preprint arXiv:1710.10723*, 2017.

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

[9] R. Nogueira and K. Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

[10] W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718*, 2019.

[11] Z. Wang, P. Ng, X. Ma, R. Nallapati, and B. Xiang. Multi-passage bert: A globally normalized bert model for open-domain question answering. *arXiv preprint arXiv:1908.08167*, 2019.

[12] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

[13] Y. Luan, J. Eisenstein, K. Toutanova, and M. Collins. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345, 2021.

[14] C. Xiong, Z. Liu, S. Sun, Z. Dai, K. Zhang, S. Yu, Z. Liu, H. Poon, J. Gao, and P. Bennett. Cmt in trec-covid round 2: mitigating the generalization gaps from web to special domain search. *arXiv preprint arXiv:2011.01580*, 2020.

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[16] P. He, J. Gao, and W. Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

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

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[19] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

[20] L. Huang, R. L. Bras, C. Bhagavatula, and Y. Choi. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*, 2019.

[21] M. Richardson, C. J. Burges, and E. Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 193–203, 2013.

[22] J. Zhan, J. Mao, Y. Liu, J. Guo, M. Zhang, and S. Ma. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1503–1512, 2021.

[23] H.-C. Fang, K.-H. Hung, C.-W. Huang, and Y.-N. Chen. Open-domain conversational question answering with historical answers. *arXiv preprint arXiv:2211.09401*, 2022.

[24] S. Yu, Z. Liu, C. Xiong, T. Feng, and Z. Liu. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on research and development in information retrieval*, pages 829–838, 2021.

[25] L. Tunstall, N. Reimers, U. E. S. Jo, L. Bates, D. Korat, M. Wasserblat, and O. Pereg. Efficient few-shot learning without prompts. *arXiv preprint arXiv:2209.11055*, 2022.

[26] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[27] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, 2009.

[28] Z. Wang, J. Liu, X. Xiao, Y. Lyu, and T. Wu. Joint training of candidate extraction and answer selection for reading comprehension. *arXiv preprint arXiv:1805.06145*, 2018.

[29] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[30] L. Pang, Y. Lan, J. Guo, J. Xu, L. Su, and X. Cheng. Has-qa: Hierarchical answer spans model for open-domain question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6875–6882, 2019.

[31] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[32] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[34] Y. Tay, A. T. Luu, A. Zhang, S. Wang, and S. C. Hui. Compositional de-attention networks. In *Advances in Neural Information Processing Systems*, pages 6135–6145, 2019.

[35] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019.

[36] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[37] Z. Zhang, Y. Wu, H. Zhao, Z. Li, S. Zhang, X. Zhou, and X. Zhou. Semantics-aware bert for language understanding. *arXiv preprint arXiv:1909.02209*, 2019.

[38] P. Yang, H. Fang, and J. Lin. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1253–1256, 2017.

[39] B. Dhingra, K. Mazaitis, and W. W. Cohen. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*, 2017.

[40] M. Dunn, L. Sagun, M. Higgins, V. U. Guney, V. Cirik, and K. Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*, 2017.

[41] S. Wang, M. Yu, J. Jiang, W. Zhang, X. Guo, S. Chang, Z. Wang, T. Klinger, G. Tesauro, and M. Campbell. Evidence aggregation for answer re-ranking in open-domain question answering. *arXiv preprint arXiv:1711.05116*, 2017.

[42] R. Das, S. Dhuliawala, M. Zaheer, and A. McCallum. Multi-step retriever-reader interaction for scalable open-domain question answering. *arXiv preprint arXiv:1905.05733*, 2019.

[43] Y. Tay, A. T. Luu, S. C. Hui, and J. Su. Densely connected attention propagation for reading comprehension. In *Advances in Neural Information Processing Systems*, pages 4906–4917, 2018.

[44] Y. Xu, Z. Lin, Y. Liu, R. Liu, W. Wang, and D. Meng. Ranking and sampling in open-domain question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2412–2421, 2019.

[45] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[46] C. Zhang, X. Zhang, and H. Wang. A machine reading comprehension-based approach for featured snippet extraction. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1416–1421. IEEE, 2018.

[47] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[48] S. R. Bowman, A. Chen, H. He, N. Joshi, J. Ma, N. Nangia, V. Padmakumar, R. Y. Pang, A. Parrish, J. Phang, et al. Quality: Question answering with long input texts, yes! *NAACL 2022*, 2022.

[49] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.