mathematics

MDPI

Article

AORO: Auto-Optimizing Reasoning Order for Multi-Hop Question Answering

Shaobo Li [1,2,*,†], Ziyi Cao [2,†], Kun Bu [2] and Zhenzhou Ji [1,2]

[1] Department of Computer Science and Technology, Harbin Institute of Technology (Weihai), Weihai 264209, China; jizhenzhou@hit.edu.cn
[2] Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China; zyc@stu.hit.edu.cn (Z.C.); 21b903084@stu.hit.edu.cn (K.B.)
* Correspondence: lishaobo@hit.edu.cn
† These authors contributed equally to this work.

Abstract

Answering multi-hop questions requires first retrieving a sequence of supporting facts, and the order in which these facts are retrieved significantly affects retriever performance. To achieve a clearer reasoning order, it is beneficial to address the easier facts first then move to the more difficult ones. However, current orders are usually pre-defined during data construction or specified manually, which restricts the model's reasoning potential. This paper proposes Auto-Optimizing Reasoning Order (AORO), a method to automatically optimize the reasoning order for each sample, where difficulty is determined by a retrieval model trained with carefully curated data. First, a retriever is trained using data that encompasses all combinations of the possible reasoning orders. The trained retriever is then used to assess the difficulty of each fact, placing the fact with the least difficulty at the beginning of the sequence. Next, the retrieval model is retrained based on these optimized sequences, which are empirically better suited to its capabilities. This process creates an iterative self-debiasing paradigm, and these steps are repeated until all facts are reordered. Experiments conducted on two multi-hop QA benchmarks, QASC and MultiRC, demonstrate the effectiveness of AORO, which outperforms strong baselines using the same PTM, and further enables advanced PTMs to achieve improvements of up to 1.6 points in Recall@10 and 3.7 points in F1 score. Additional case analyses reveal empirical patterns in the optimal reasoning order: the pattern appears independent of the dataset and the underlying pre-trained model; and the sequence proceeds by confirming the truth of the question, answering the question, and filling in any gaps, which aligns with human reasoning.

Keywords: natural language processing; question answering; information retrieval; neural network; machine learning

MSC: 68T50

check for updates

Academic Editors: Zhao Kang and Marjan Mernik

Received: 27 August 2025
Revised: 26 September 2025
Accepted: 21 October 2025
Published: 1 November 2025

Citation: Li, S.; Cao, Z.; Bu, K.; Ji, Z. AORO: Auto-Optimizing Reasoning Order for Multi-Hop Question Answering. *Mathematics* **2025**, *13*, 3489. https://doi.org/10.3390/math13213489

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Multi-hop Question Answering (QA) is a challenging and complex form of QA [1–3] that requires addressing information needs that cannot be satisfied by a single query [4–6]. In such tasks, the answer cannot be derived from a single piece of fact [7]; rather, it must be obtained by retrieving, integrating, and reasoning over multiple factual statements to produce the correct answer [8,9]. A key advantage [10–12] of multi-hop QA is its

ability to address questions that require complex reasoning or a deep understanding of the context [13,14]. In particular, gathering multiple facts typically requires several retrieval steps and can take the form of iterative single-hop retrieval [15–17]. During iterations, each subsequent retrieval step is guided by the initial question and the previously retrieved facts [18,19], producing a sequence of retrieved evidence referred to as the **reasoning order** [20].

Figure 1 presents an example of multi-hop multi-choice QA. The question is *What makes food easier to chew?* and the correct choice is *liquids*. The retriever is required to collect both $\hat{f}_a$ and $\hat{f}_b$. In the single-hop setting, $\hat{f}_b$ ranks 1st and $\hat{f}_a$ ranks 16th. In the multi-hop setting, the first hop is identical to the single-hop result, and the second hop, conditioned on Q and $\hat{f}_b$, promotes $\hat{f}_a$ to the first place. Compared with the single-hop setting, the multi-hop setting includes an additional search step, which can use the clue *saliva* in the single-hop result $\hat{f}_t extb$ to find $\hat{f}_a$ more easily. Consequently, the multi-hop setting can outperform the single-hop setting by retrieving and combining newly emerged content that contains the necessary information.

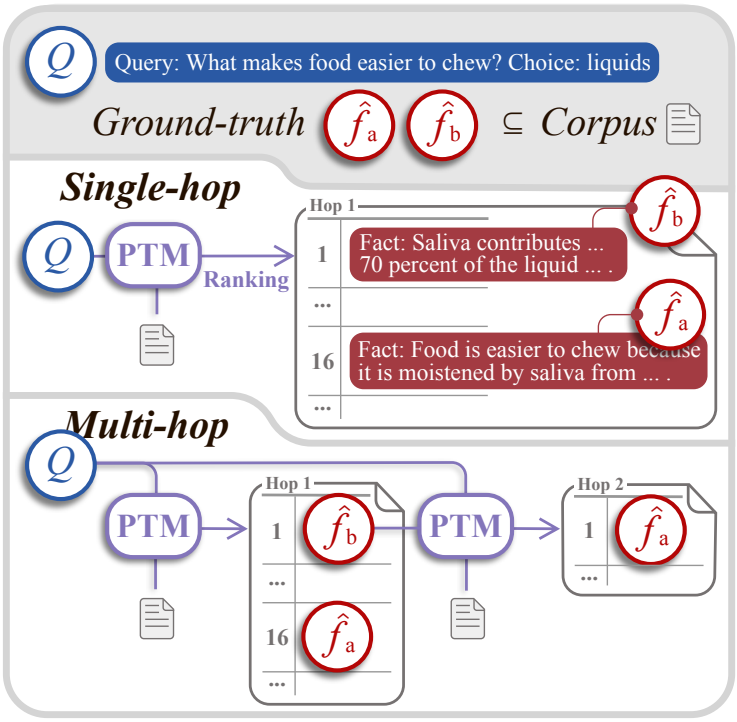


Figure 1. An example of an inconsistency in the reasoning order $\{\hat{f}_b, \hat{f}_a\}$ in the facts retrieved by the Pre-Trained Model (PTM) compared with the ground truth $\{\hat{f}_a, \hat{f}_b\}$ on the QASC dataset [21].

However, a phenomenon usually arises in multi-hop retrieval: differences in reasoning order. As shown in Figure 1, the retriever outputs the order $\{\hat{f}_b, \hat{f}_a\}$, whereas the ground-truth order in the training set is $\{\hat{f}_a, \hat{f}_b\}$. The retrieved information is the same, but the orders differ. Naturally, two questions arise: **Question 1 (Q1): What is the better reasoning order? and Question 2 (Q2): How can that order be obtained?** Inspired by curriculum learning [22], this work posits that the reasoning process should progress from easier to harder retrieval steps. In this context, "easier" refers to the high probability of a fact being correctly retrieved at the current step. Addressing the "easier" facts first mitigates the risk of error propagation across subsequent reasoning steps [15]. Specifically, in AORO, "easier" is indicated by the retrieval model itself; the item that receives a higher rank under the current query state is deemed easier.

Based on the above assumption, in **Q1**, the highest-ranked item at a given hop is considered easier to retrieve at that retrieval step. For instance, in the multi-hop retrieval

example illustrated in Figure 1, for hop 1, based on the query (Q), item $\hat{f}_b$ is ranked higher than the item $\hat{f}_a$. This suggests that retrieving $\hat{f}_b$ first and then $\hat{f}_a$ is better, whereas trying to retrieve $\hat{f}_a$, which ranks 16th in the first hop, would make retrieval more challenging.

For **Q2**, a retrieval model is required to produce a superior reasoning order by placing the easier item earlier in the sequence. The central task is therefore to construct such a "unbiased" model that can determine whether an item is easier. To this end, a retrieval model is trained on a consistency-based curated training set as the unbiased model. As shown in Figure 2, an illustrative procedure is as follows.

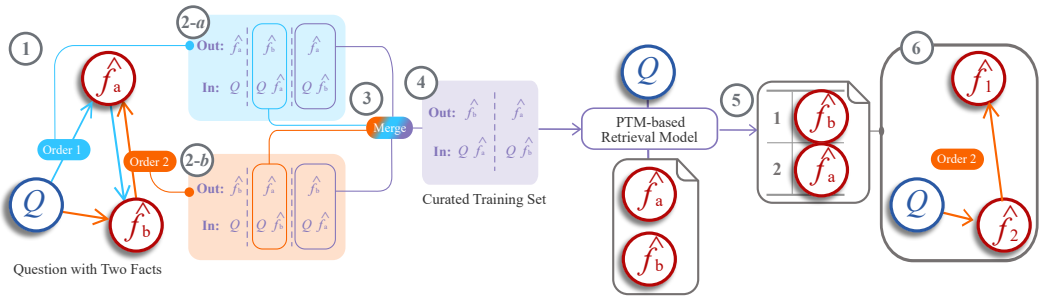


Figure 2. Overview of the automatic optimization of the reasoning order for two facts.

Step ①: Enumerate all feasible reasoning orders for a given sequence, namely Order 1 and Order 2. Step ②: Identify consistency samples across different orders and remove conflicting samples to avoid biased or confusing training data. For example, Order 1 yields three samples, which are marked with ②-*a* (in blue), and Order 2 yields three samples as well, marked with ②-*b* (in orange). The first sample in Order 1 conflicts with the first sample in Order 2, and the second and third samples are identical across the two orders. Step ③: Merge the samples that share identical input and output components to form a curated training set. The PTM-based retrieval model is then trained PTM-based retrieval model in Step ④. The resulting retrieval model is subsequently used to rank the candidate facts in Step ⑤. This result indicates that $\hat{f}_b$ ranks higher than $\hat{f}_a$, suggesting that the model finds $\hat{f}_b$ easier to retrieve. Consequently, $\hat{f}_b$ is designated as the first retrieval target (renamed to $\hat{f}_1$), and $\hat{f}_a$ is set as the second target (renamed to $\hat{f}_2$), following Order 2. This entire process represents the Auto-Optimizing Reasoning Order (AORO) framework.

AORO is evaluated on the multi-hop QA datasets QASC and MultiRC. The results show that AORO can improve the search performance over the strong baselines on both datasets, indicating that AORO yields a more effective reasoning order. Furthermore, AORO not only surpasses manual methods in optimizing orders but also produces reasoning sequences with high similarity across different PTMs. This similarity remains stable across datasets, suggesting that the optimal reasoning order is largely independent of the particular PTM or dataset. Case studies further reveal a consistent pattern in the induced order.

The contributions of this paper can be summarized as follows:

- This paper presents AORO to automatically optimize the reasoning order by iteratively training the model and selecting the better retrieval target.
- On two multi-hop QA datasets, QASC and MultiRC, AORO outperforms both non-ranking and rule-based ranking methods, demonstrating its effectiveness.
- Analysis across different PTMs and datasets indicates that the learned order is consistently superior and that the easier to harder retrieval strategy generalizes, underscoring the importance of selecting an appropriate reasoning order before launching the reasoning process.

This article is organized as follows. Section 2 briefly reviews related work. Section 3 presents the proposed approach, AORO. Specifically, Section 3.1 defines the concepts and symbols used throughout this paper. Section 3.2 introduces the process for automatically optimizing the reasoning order. Then, Section 3.3 describes the use of the optimized reasoning order to finetune the PTMs in the retrieval models. Section 4 describes the experimental procedures, and Section 5 presents the results and analysis of the findings. Finally, this paper concludes with an outline of directions for future work.

2. Background and Related Work

2.1. Multi-Hop Task

The initial component of multi-hop QA is multi-hop retrieval [23], which aims to gather the relevant facts needed for the subsequent answering phase [24,25]. Unlike single-hop retrieval tasks, such as MS MARCO [26] and WikiQA [27], the multi-hop retrieval requires the collection of more than one fact. Typical examples of multi-hop tasks are HotpotQA [8] and 2WikiMultiHopQA [10], where the questions are related to multiple extensive Wikipedia pages. This task demands not only effective search capabilities but also the ability to handle long texts [28,29], which adds complexity to the study of reasoning order. To simplify the problem, the datasets used in this paper are QASC [21] and MultiRC [30]. Both datasets consist of short sentences, making the management of reasoning order the primary issue to address.

2.2. Multi-Hop Retrieval

Methods for both datasets take into account the multi-hop structure. AIR (Alignment-based Iterative Retriever) [31] employs the similarity computation of AHE (Alignment over Heterogeneous Embeddings) [32] to eliminate irrelevant sentences while retaining the top 80 candidates. AHE uses word embeddings, such as GloVe [33] and BERT [34], to create an alignment cosine-similarity matrix between the words in the query and the candidate set. This matrix calculates the maximum alignment score, weighted by the Inverse Document Frequency (IDF) of the query words, along with summary scores. The AHE matrix simultaneously filters out retrieved words while keeping those that were not retrieved for further processing. However, AHE does not account for the meaning of entire sentences, which limits its effectiveness. To address this limitation, SingleRR and JointRR [35] utilize AHE as a filter and include a re-ranking component to better understand sentence meanings. In order to accommodate complete semantics, Beam Retrieval [36] models the multi-hop retrieval process in an end-to-end manner by jointly optimizing an encoder and two classification heads across all hops. In addition, RPA (Reasoning Path Augmentation) [14] employs RoBERTa-Large [37] to embed the sentences based on the same candidate set. The approach discussed in this paper uses the same multi-hop retrieval candidates with RPA, aiming for a fair comparison of multi-hop retrieval methods with the aforementioned techniques.

2.3. Reasoning Order

Reasoning orders are intended to be optimized so that easier facts are located first while more difficult ones are tackled later [14,38,39]. To achieve this, RPA [14] introduces reasoning-path augmentation to optimize the order of reasoning by extracting overlaps between sentences. The RPA consist of three optimization rules: (i) a set of query words, (ii) the set union of the query and retrieved facts, and (iii) the set intersection of the query and retrieved facts, with the third option yielding better results. However, when the number of set-intersection words among different facts is the same, it can be challenging to sort the ground-truth facts. Additionally, the priority of fact retrieval is often determined

by manual rules, raising the question of whether a better order exists. To address this issue, this paper proposes an auto-optimization approach for reasoning order, which generates the order by training a model rather than relying on predefined rules.

3. Methods

This section first introduces the definitions of the task and notation used in this paper. Next, the automatic optimization process for the reasoning order is described. Finally, the obtained order is used to train a retrieval model and examine how changes in order affect the outcomes.

3.1. Task and Notation Definition

Given a query Q, a retrieval model is designed to gather all ground-truth facts $\hat{F} = \{\hat{f}_\mathrm{a}, \hat{f}_\mathrm{b}, \cdots, \hat{f}_\mathrm{x}, \cdots\}$ ($|\hat{F}| = N$) from the corresponding candidate corpus C. The optimized reasoning order, denoted as $\hat{F}^{\mathrm{Opt}}$, is initially set to $\varnothing$. The process of automatically optimizing the reasoning order involves using the retrieval model to extract the facts from $\hat{F}$ in a specific sequence and append them to $\hat{F}^{\mathrm{Opt}}$. In the end, $\hat{F}$ is emptied (i.e., $\hat{F} = \varnothing$) and $\hat{F}^{\mathrm{Opt}}$ will contain the facts in the optimized order: $\{\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_N\}$. To indicate the optimized order and distinguish it from $\hat{F}$, the subscript of $\hat{f}$ in $\hat{F}^{\mathrm{Opt}}$ uses numbers instead of letters.

This section outlines two Information Retrieval (IR) processes: training and prediction. Their corresponding formulations differ as follows:

$$\text{Train:} \quad \max \mathrm{IR}(Q, \hat{f}_1, \cdots; \hat{f}_\mathrm{x}), \tag{1}$$

$$\text{Predict:} \ \hat{f}_\mathrm{x} = \arg\max_{\hat{f}_\mathrm{i} \in \hat{F}} \mathrm{IR}(Q, \hat{f}_1, \cdots; \hat{f}_\mathrm{i}). \tag{2}$$

Equation (1) states that when the input is $\{Q, \hat{f}_1, \cdots\}$, the training objective is to maximize the probability of obtaining $\hat{f}_\mathrm{x}$. For example, in the middle term of ②-*a* in Figure 2, "In" denotes $\{Q, \hat{f}_\mathrm{a}\}$ and "Out" denotes $\hat{f}_\mathrm{b}$. This relationship can be expressed as $\max \mathrm{IR}(Q, \hat{f}_\mathrm{a}; \hat{f}_\mathrm{b})$. Equation (2) indicates that when the input is $\{Q, \hat{f}_1, \cdots\}$, $\hat{f}_\mathrm{x}$ is the fact with the highest predicted probability among the non-input ground-truth facts. For instance, the ⑤ in Figure 2 is represented by $\hat{f}_\mathrm{b} = \arg\max_{\hat{f}_\mathrm{i} \in \{\hat{f}_\mathrm{a}, \hat{f}_\mathrm{b}\}} \mathrm{IR}(Q; \hat{f}_\mathrm{i})$. For simplicity, Equation (3) (**T**raining) is used in place of Equations (1) and (4) (**P**rediction) replaces Equation (2) in the following discussion.

$$Q + \hat{f}_1 + \cdots \xrightarrow{\mathrm{T}} \hat{f}_\mathrm{x}, \tag{3}$$

$$Q + \hat{f}_1 + \cdots \xrightarrow{\mathrm{P}} \hat{f}_\mathrm{x}. \tag{4}$$

In the equations above, "+" denotes sentence concatenation. Therefore, in Figure 2, the training process in ④ can be represented as $Q + \hat{f}_\mathrm{a} \xrightarrow{\mathrm{T}} \hat{f}_\mathrm{b}$ and $Q + \hat{f}_\mathrm{b} \xrightarrow{\mathrm{T}} \hat{f}_\mathrm{a}$. Additionally, the reasoning order in ⑥ of Figure 2 can be expressed as $Q \xrightarrow{\mathrm{P}} \hat{f}_2 \xrightarrow{\mathrm{P}} \hat{f}_1$, where the subscripts "b" and "a" are replaced with "2" and "1," respectively, to indicate the optimized order.

The AORO involves iteratively selecting the fact that is easiest to retrieve based on the query while simultaneously updating the query itself. Algorithm 1 illustrates this process. The algorithm consists of three main components: **building unbiased training data**, **training** the model, and making **predictions**. Each of these components is described in detail in this section.

Algorithm 1 Automatic optimization of reasoning order.

Input: $Q, \hat{F} = \{\hat{f}_a, \hat{f}_b, \cdots\}$, PTM
Output: $\hat{F}^{Opt} = \{\hat{f}_1, \hat{f}_2, \cdots\}$
1: $\hat{F}^{Opt} \leftarrow \varnothing$
2: **while** $\hat{F} \neq \varnothing$ **do**
3: $Q' \leftarrow Q + \hat{f}_1 + \cdots, (\hat{f}_1, \cdots \in \hat{F}^{Opt})$ ▷ Collect the feasible orders
4: orders $\leftarrow$ Permutation($\hat{F}$)
5: order2data $\leftarrow$ Mapping()
6: **foreach** order$_i$ $\in$ orders **do**
7: ori_data$_i$ $\leftarrow$ TrainingData(order$_i$)
8: aug_data$_i$ $\leftarrow$ AugmentationData(order$_i$)
9: data$_i$ $\leftarrow$ ori_data$_i$ $\cup$ aug_data$_i$
10: add $\langle$order$_i$, data$_i\rangle$ into order2data
11: **end for**
12: data $\leftarrow$ Merge(order2data) ▷ Select the unbiased samples
13: PTM $\leftarrow$ Train(PTM, data) ▷ Training retrieval model
14: $\hat{f}_x \xleftarrow{P} Q', (\hat{f}_x \in \hat{F})$ ▷ Prediction with retrieval model
15: $\hat{F} \leftarrow \hat{F} \backslash \{\hat{f}_x\}$
16: $\hat{F}^{Opt} \leftarrow \hat{F}^{Opt} \cup \{\hat{f}_x\}$
17: **end while**
18: **return** $\hat{F}^{Opt}$

3.2. Automatic Optimization of Reasoning Order

3.2.1. Building Unbiased Training Data

The core objective of this stage is to eliminate the bias introduced by the predefined reasoning order formed in the original training data. Constructing an unbiased dataset enables the retrieval model to assess the intrinsic difficulty of retrieving each fact on its own merits. The procedure unfolds in three main steps: *(Step 1)* fact permutation, *(Step 2)* data generation for each permutation, and *(Step 3)* curation via merging. This process corresponds to Lines 2–12 in Algorithm 1 and is illustrated in steps ① to ③ of Figure 2.

***Step 1*: Fact Permutation.** The initial step in mitigating order bias is to consider all potential reasoning orders. This is achieved by enumerating every possible permutation of the facts in $\hat{F}$ (Line 4 of Algorithm 1). For a set of N facts, this results in $N!$ possible orderings.

***Step 2*: Data Generation for Each Permutation.** For each permutation, a set of training samples is generated to teach the model how to follow that specific reasoning path (Lines 7–8 of Algorithm 1). This set includes two types of samples:

- **Standard samples**, which are generated based on the reasoning orders. For a given order, each fact is treated as the target output, with the query and all preceding facts serving as the input.
- **Augmented samples**, which are created by inserting a different fact into the reasoning order. This simulates a scenario where a different fact was mistakenly retrieved in a previous step, thereby training the retrieval model to recover from the error. The objective is to guide the model back to the correct target for the current stage.

***Step 3*: Curation via Merging.** The final step is to create the unbiased dataset by retaining only the training samples that are consistent across all permutations (Line 12 of Algorithm 1). This is achieved by taking the intersection of the sample sets generated in *Step 2*. This merging process naturally filters out conflicting training samples, and the resulting curated dataset is agnostic to any specific order.

To make this process concrete, consider the example from Figure 2 where the set of ground-truth facts is $\hat{F} = \{\hat{f}_a, \hat{f}_b\}$. First, the two facts yield $2! = 2$ possible reasoning orders:

Order 1 ($Q \xrightarrow{\mathrm{P}} \hat{f}_{\mathrm{a}} \xrightarrow{\mathrm{P}} \hat{f}_{\mathrm{b}}$) and Order 2 ($Q \xrightarrow{\mathrm{P}} \hat{f}_{\mathrm{b}} \xrightarrow{\mathrm{P}} \hat{f}_{\mathrm{a}}$). Second, the full set of standard and augmented samples is generated for both the above orders:

$$< \text{Order 1}, \begin{cases} Q \xrightarrow{\mathrm{T}} \hat{f}_{\mathrm{a}} & \text{(Standard)} \\ Q + \hat{f}_{\mathrm{a}} \xrightarrow{\mathrm{T}} \hat{f}_{\mathrm{b}} & \text{(Standard)} \\ Q + \hat{f}_{\mathrm{b}} \xrightarrow{\mathrm{T}} \hat{f}_{\mathrm{a}} & \text{(Augmented)} \end{cases} >, \tag{5}$$

$$< \text{Order 2}, \begin{cases} Q \xrightarrow{\mathrm{T}} \hat{f}_{\mathrm{b}} & \text{(Standard)} \\ Q + \hat{f}_{\mathrm{b}} \xrightarrow{\mathrm{T}} \hat{f}_{\mathrm{a}} & \text{(Standard)} \\ Q + \hat{f}_{\mathrm{a}} \xrightarrow{\mathrm{T}} \hat{f}_{\mathrm{b}} & \text{(Augmented)} \end{cases} > . \tag{6}$$

The samples labeled as *(Augmented)* are generated by inserting different samples into the original reasoning order. Third, by taking the intersection of the two datasets above, the conflicting first-hop samples ($Q \xrightarrow{\mathrm{T}} \hat{f}_{\mathrm{a}}$ and $Q \xrightarrow{\mathrm{T}} \hat{f}_{\mathrm{b}}$) are discarded. The final, unbiased training data consists only of the two samples common to both sets: $Q + \hat{f}_{\mathrm{a}} \xrightarrow{\mathrm{T}} \hat{f}_{\mathrm{b}}$ and $Q + \hat{f}_{\mathrm{b}} \xrightarrow{\mathrm{T}} \hat{f}_{\mathrm{a}}$. The above process effectively filters out the conflicting first-hop samples (e.g., $Q \xrightarrow{\mathrm{T}} \hat{f}_{\mathrm{a}}$ and $Q \xrightarrow{\mathrm{T}} \hat{f}_{\mathrm{b}}$), as they are inconsistent across the possible permutations. The resulting curated dataset can prevent the model from developing a bias toward any single, predefined reasoning path. Training on such unbiased data enables the model to more accurately assess the intrinsic difficulty of retrieving each fact.

3.2.2. Training

In Line 13 of Algorithm 1, the model is trained on the merged training data. In the training process, let $Q' = Q + \hat{f}_1 + \cdots$ (Equation (3)), and let F ($F \subseteq C, F \cap \hat{F} = \varnothing, F \cap \hat{F}^{\mathrm{Opt}} = \varnothing$) represent the negative information set. The training loss is the following.

$$\mathscr{L} = -\frac{1}{|F|} \sum_{f \in F} \log \frac{e^{h(Q', \hat{f}_{\mathrm{x}})}}{e^{h(Q', \hat{f}_{\mathrm{x}})} + e^{h(Q', f)}}. \tag{7}$$

The function $h(\cdot, \cdot)$ in the equation denotes the dot product of the [CLS] embeddings from the last layer of the two inputs. Q' represents the dynamic query context, which is formed by concatenating the original query Q with any facts that have already been identified ($\hat{f}_1$, etc.). This updated context provides the model with the accumulated information needed to retrieve the next piece of evidence. Meanwhile, $\hat{f}_{\mathrm{x}}$ is the target fact for the current reasoning step. In the contrastive loss formulation, $\hat{f}_{\mathrm{x}}$ serves as the positive instance that the model must learn to score higher than all other negative facts f from the set F, given the current context Q'. The [CLS] token is a special input token used by Transformer-based models to capture an aggregate sequence representation. The PTMs used in this paper, specifically RoBERTa and DeBERTa, provide this embedding from their final layer. This resulting vector is then used to compute the similarity score between the two texts.

3.2.3. Prediction

By Line 14 of Algorithm 1, the model has been trained on unbiased data, which suggests that it does not favor any particular order of inference. The predicted order then can be determined by the model's learned preferences. In the prediction step (see Equation (2)), the model assumes that when the input is $Q' = Q + \hat{f}_1 + \cdots$, the fact $\hat{f}_{\mathrm{x}}$ is the most easily searchable ground-truth fact, i.e.,

$$\begin{aligned} \hat{f}_{\mathrm{x}} &= \arg\max_{\hat{f}_{\mathrm{i}} \in \hat{F}} \mathrm{IR}(Q, \hat{f}_1, \cdots; \hat{f}_{\mathrm{i}}) \\ &= \arg\max_{\hat{f}_{\mathrm{i}} \in \hat{F}} h(Q', \hat{f}_{\mathrm{i}}). \end{aligned} \tag{8}$$

Since $\hat{f}_{\mathrm{x}}$ is selected by the model, it indicates that $\hat{f}_{\mathrm{x}}$ should be retrieved now, reflecting the preference for the better reasoning order at this time.

3.3. Retraining Based on Orders

Algorithm 1 automatically optimizes the original reasoning order from $\hat{F} = \{\hat{f}_{\mathrm{a}}, \hat{f}_{\mathrm{b}}, \cdots\}$ to $\hat{F}^{\mathrm{Opt}} = \{\hat{f}_1, \hat{f}_2, \cdots\}$. This process yields two orders: naive order and optimized order. To assess the impact of order on performance, the PTM must be retrained using these orders. The data used for retraining includes the training data and the augmentation data. For example, if the order is determined to be Order 1, the retraining data is specified in Equation (5). Therefore, the actual performance of AORO is based on retraining after the order has been established, rather than on the effects during the optimization process.

4. Experiments

This section introduces the datasets used, establishes the baselines for comparison, selects the core PTMs in AORO, and defines the metric for comparing different reasoning orders. Additionally, several methods are designed to imitate the other optimized orders for comparative evaluation. This allows for direct acquisition of a better order in the future without needing to perform the AORO process.

4.1. Datasets

In contrast to the lengthy text of HotpotQA [8] and the single fact from MS MARCO [26], this work focuses on the search order across multiple facts by selecting the multi-fact datasets of shorter texts: QASC and MultiRC.

- **QASC** [21]: This dataset consists of an 8-choice QA task (https://allenai.org/data/qasc (accessed on 4 January 2024)) with a knowledge base containing 17 million facts. For each question, there are always two ground-truth facts, denoted as $\{\hat{f}_1, \hat{f}_2\}$. Following the evaluation protocol of prior work on this dataset [31,35], performance is primarily measured using Recall@10 (*both found* and *at least one found*). This means that in the top ten retrieved facts, the recall is measured for both facts being found as well as for at least one fact being found. The candidate set C on QASC is extracted from the knowledge base using Heuristic+IR [21], resulting in an upper bound for the validation set of 81.8 for Recall@10 *at least one found* and 61.3 for Recall@10 *both found*. After training, each sample in the validation set retrieves one fact in the first hop and nine facts in the second hop, resulting in a total of ten facts collected over the two hops.
- **MultiRC** [30]: This dataset is a multiple-choice QA task, where each sample includes a question, a set of 2 to 14 answer choices, 2 to 4 ground-truth facts for the validation set, 2 to 6 ground-truth facts for the training set, and a corresponding paragraph. The version of MultiRC used in this paper is the original MultiRC (https://cogcomp.seas.upenn.edu/multirc/ (accessed on 4 January 2024)) and not the one included in SuperGLUE [40]. The candidate set, denoted as C, is derived from all the facts within the paragraph. In the training phase, the maximum number of ground-truth facts is six, which also restricts the maximum number of iterations for auto-optimization to six. To ensure a fair comparison with RPA, AORO utilizes the same dynamic hop-

stopping method during the MultiRC validation process. Specifically, for each query, the query's non-stop words that appear in the retrieved facts are removed. Once all applicable words have been removed, the process of hopping stops. Given that the validation set contains a maximum of four facts, the maximum number of hops allowed is also four. Since the number of hops changes dynamically, the F1 score is used as the evaluation metric.

4.2. Baselines

For the iterative retrieval of multiple facts, there are two types: ignoring the order and considering the order, which correspond to **no order** and **optimized order**, respectively. The methods discussed in this paper are as follows.

- **BM25** [41] (no order): The traditional BM25 retrieval algorithm ranks a set of facts based on query terms that appear within each fact, without considering their proximity.
- **AutoROCC** [41] (no order): This method introduces an unsupervised strategy for selecting facts. It is an enhanced version of BM25 that aims to (i) maximize the relevance of the selected facts, (ii) minimize the overlap between the selected facts, and (iii) maximize the coverage of both the question and the answer. Based on these principles, which are tailored for multi-hop QA, AutoROCC delivers better search results compared to BM25.
- **AIR** [31] (no order): This is a straightforward, fast, and unsupervised iterative evidence retrieval method known as AIR (Alignment-based Iterative Retriever). AIR is based on three key concepts: (i) an unsupervised alignment method that soft-aligns questions and answers with facts using embeddings, such as GloVe; (ii) an iterative process that reformulates queries to emphasize terms not addressed by the existing facts; (iii) a stopping criterion that halts retrieval when the terms in the question and candidate answers have been covered by the retrieved facts. Because AIR uses an iterative approach, facts are gathered one at a time, which establishes a specific reasoning order. Although the primary objective of AIR is collection rather than sequence, it can also incorporate a method for reordering research. In this context, one of its alignment methods, GloVe, is discussed in Section 4.6, focusing on training-free techniques.
- **RPA** [14] (optimized order): Reasoning Path Augmentation (RPA) is a method that adjusts the order of facts based on the number of overlapping words, which is an artificial rule. This method operates under the assumption that the more overlapping terms a fact has with the query, the more relevant it is to that query. By prioritizing these relevant facts, RPA can help reduce complexity and enhance retrieval results. However, it is uncertain whether the artificially determined order is indeed the most effective one. For the sake of comparison, RPA's order is replicated and tested in the AORO environment.

4.3. PTMs

To explore the relationship between the order obtained through automatic optimization and PTM, AORO employs the following two PTMs in the experiment:

- **RoBERTa** [37]: RoBERTa has the same model structure as BERT, but it differs in several important ways. RoBERTa utilizes more data, a larger batch size, and a dynamic masking method, all of which enhance its performance on downstream tasks. To compare with RPA, which also optimizes reasoning orders, AORO employs the same RoBERTa model.
- **DeBERTa** [42]: DeBERTa encodes the content and positional information of words separately and employs disentangled matrices to calculate attention weights. This

approach distinguishes it from BERT and RoBERTa. As a result, DeBERTa outperforms both RoBERTa and BERT in various downstream tasks.

The purpose of using RoBERTa is to enable a comparison with the RPA method, while DeBERTa is employed to assess the effectiveness of AORO when applied with different PTMs and to determine whether the order of optimization for various PTMs yields consistent results.

4.4. Training Details

AORO Retriever. The AORO retriever was trained on the QASC and MultiRC datasets, which have candidate set sizes of 80 and an average of 15, respectively. To balance the sampling of negative samples, the value of $|F|$ in QASC is set at 5, whereas in MultiRC, it is set at 2, as shown in Equation (7). Additionally, since a batch contains multiple samples during training, each sample in QASC includes two facts, while MultiRC can contain a maximum of six facts. This results in a smaller batch size for MultiRC, as illustrated in Table 1. Table 1 also highlights that, due to the large number of parameters in DeBERTa, its batch size is relatively small. Base on AORO, the Lamb optimizer [43] is employed in the experiments with the following hyperparameters: a learning rate of 5×10^{-6} and beta values of (0.9, 0.999) [44]. The framework used is Lightning (https://lightning.ai/ (accessed on 19 May 2024)). The minimum epoch for QASC is set at 100, with a maximum of 500 epochs, where one checkpoint is stored per epoch. For MultiRC, the minimum epoch is 24, and the maximum is also 500, with a checkpoint saved every 1000 steps. An early stopping strategy is in place: if the metrics (Recall@10 *both found* for QASC and F1 for MultiRC) do not improve after reaching the minimum epoch, Lightning will wait for 30 checkpoints saving before halting the training.

Reader Models. The reader models were finetuned for the multiple-choice question answering task. The training objective was to minimize the cross-entropy loss using the AdamW optimizer with a learning rate of 2×10^{-5}. A linear scheduler with a warm-up phase constituting 10% of the total training steps was used to adjust the learning rate. All experiments were performed on two NVIDIA RTX 3090 Ti GPUs.

Table 1. Configurations of PTMs.

Pretrained Models	RoBERTa-Large	DeBERTa-Large
Parameters (M)	355	405
Batch size (QASC)	32	4
Batch size (MultiRC)	12	2

4.5. Similarity Metric for Orders

There are two categories of $\hat{F}^{\text{Opt}}$ orders used for training, as detailed below:

- **RPA**'s optimized order ($\hat{F}^{\text{RPA}}$): The reasoning order is optimized by RPA. Since RPA uses RoBERTa, $\hat{F}^{\text{RPA}} = \hat{F}^{\text{RPA}}_{\text{RoBERTa}}$.
- **AORO**'s optimized order ($\hat{F}^{\text{AORO}}$): The reasoning order is optimized by the AORO algorithm. It is further divided into $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$ and $\hat{F}^{\text{AORO}}_{\text{DeBERTa}}$, reflecting the differences between the models.

To illustrate the gap between the above categories, a straightforward similarity calculation method (*sim*) is defined, which constructs the similarity between sentences based on the edit distance (*dist*):

$$sim(\hat{F}^{\text{Order i}}, \hat{F}^{\text{Order j}}) = \frac{N - dist(\hat{F}^{\text{Order i}}, \hat{F}^{\text{Order j}})}{N}. \quad (9)$$

For example, $sim(\{\hat{f}_a, \hat{f}_b\}, \{\hat{f}_b, \hat{f}_a\}) = (2-2)/2 = 0\%$; $sim(\hat{F}^{\text{RPA}}, \hat{F}^{\text{AORO}})$ refers to the measure of similarity between the two order categories: RPA and AORO. The term $sim(\hat{F}^{\text{RPA}}_{\text{RoBERTa}}, \hat{F}^{\text{AORO}}_{\text{RoBERTa}})$ indicates the similarity between two sequences when utilizing RoBERTa as PTM. In comparison to measuring the exact sameness of each position, similarity based on edit distance more accurately reflects the differences and similarities in order.

4.6. Training-Free Methods

In the previous experiments, AORO needs to be executed to establish a specific order among the facts. This process requires extensive iterative training time, and the final sequence must be run again to evaluate actual inference performance. Therefore, it is essential to employ a training-free method to generate an inference order that closely resembles AORO, allowing for its direct application in search and reducing the iterative training phase of AORO. To achieve this, four types of training-free methods are utilized, TF-IDF Term Frequency-Inverse Document Frequency [45], RPA [14], GloVe of AIR [31], and spaCy (https://spacy.io/ (accessed on 23 May 2024)), to generate the orders. These training-free methods can be categorized into three distinct types: word weight calculation, overlapping words, simple word vectors, and an integrated tool. These methods are discussed further in Section 5.4.

The AORO optimization process, utilizing large PTMs (RoBERTa-Large and DeBERTa-Large), is computationally intensive. For each of the ~9000 training samples in QASC ($N = 2$ facts), the process requires one $(N-1)$ full fine-tuning round. This expands for MultiRC, which demands up to five $(N-1)$ fine-tuning rounds for samples with $N = 6$ facts. The complete one-time optimization for both datasets consumed approximately 72 h on a dual NVIDIA RTX 3090 Ti platform. While this represents a significant upfront resource investment, it is a direct trade-off for achieving a superior, model-aware reasoning order. The resulting performance improvements, as demonstrated in the detailed results in the Section 5.4, validate that this resource-intensive approach is crucial for discovering the effective reasoning paths that lead to better results.

5. Experimental Results

5.1. Performance

The effectiveness of AORO's order optimization is evaluated using a comprehensive set of metrics appropriate for each dataset. For QASC, performance is measured using Recall@10. For the more dynamic MultiRC dataset, a granular analysis using Precision, Recall, and the F1 score is provided to fully characterize retrieval quality. The results, categorized into three settings (AORO with RoBERTa, with DeBERTa, and a hybrid RoBERTa on DeBERTa), demonstrate that AORO consistently outperforms the baseline methods. The detailed analysis is as follows:

- **AORO (RoBERTa)**: When the PTM is RoBERTa, AORO generates $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$, which is then used to train RoBERTa to achieve improved search performance. In the QASC dataset (see Table 2), when using the same RoBERTa as the training PTM, AORO outperforms the RPA by 0.4 in the Recall@10 *both found* metric and shows an improvement of 0.3 in Recall@10 *at least one found*. On the MultiRC dataset (see Table 3), AORO also exceeds RPA by 1.0 in the F1 metric. This indicates that the proposed AORO provides a more search-friendly reasoning order than RPA. Specifically, $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$ is superior to $\hat{F}^{\text{RPA}}_{\text{RoBERTa}}$, confirming that automatic reasoning optimization can achieve a more logical order compared to the manual rules used in RPA. However, it is evident that the improvement observed on QASC is less significant than that on MultiRC, which is analyzed in detail in Section 5.3.

- **AORO (DeBERTa)**: Similar to AORO (RoBERTa), AORO utilizes $\hat{F}^{\text{AORO}}_{\text{DeBERTa}}$ to train DeBERTa for the search task. Given that AORO outperforms RPA (RoBERTa) in search performance, DeBERTa further enhances the results of AORO. For instance, on the QASC dataset, as shown in Table 2, DeBERTa achieves a 1.2 improvement in Recall@10 *both found* over AORO (RoBERTa). Additionally, on the MultiRC dataset, as demonstrated in Table 3, it shows a 3.3 point increase in F1 score. This improvement is not only attributable to the advantages of $\hat{F}^{\text{AORO}}_{\text{DeBERTa}}$ compared to $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$ but also to DeBERTa's superior model architecture and its powerful encoding capabilities in search tasks.
- **AORO (RoBERTa on DeBERTa)**: To further investigate why AORO (DeBERTa) outperforms AORO (RoBERTa), AORO (RoBERTa on DeBERTa) is introduced, which utilizes $\hat{F}^{\text{AORO}}_{\text{DeBERTa}}$ to train RoBERTa. In this experiment, the performance of AORO (RoBERTa on DeBERTa) is compared using the same RoBERTa. If AORO (RoBERTa on DeBERTa) surpasses AORO (RoBERTa), it would indicate that $\hat{F}^{\text{AORO}}_{\text{DeBERTa}}$ is more effective than $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$ in reasoning tasks. As expected, AORO (RoBERTa on DeBERTa) performs 0.2 better on Recall@10 *both found* for the QASC dataset (see Table 2) and achieves a 0.6 increase in F1 score for the MultiRC dataset (see Table 3). Although these improvements are modest, they support the observation that $\hat{F}^{\text{AORO}}_{\text{DeBERTa}}$ outperforms $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$. This suggests that **the stronger the pre-trained model, the more effective its reasoning capabilities become**. It also implies that **even weaker models can achieve better results when using an effective reasoning order**.

Table 2. Recall@10 on the QASC dataset. "RoBERTa on DeBERTa" indicates that the PTM RoBERTa utilizes the order achieved through the DeBERTa optimization during training. The others utilize their own PTMs and orders.

Method	Both Found	At Least One Found
No order		
BM25 [31]	27.8	65.7
Heuristics [21]	41.6	64.6
BERT-LC [21]	41.6	64.4
AIR (parallel = 5) [31]	44.8	68.6
SingleRR [35]	44.4	69.6
Two-step IR [21]	44.4	69.9
SupA + QA [46]	47.8	-
Optimized order		
RPA (RoBERTa) [14]	52.7	77.5
AORO (RoBERTa)	53.1	**77.8**
AORO (RoBERTa on DeBERTa)	53.3	78.1
AORO (DeBERTa)	54.3	78.4

To provide a more comprehensive evaluation of the AORO retriever, its performance was evaluated in a standard retriever–reader pipeline when paired with various reader models [47]. In this standard retriever–reader pipeline, the reader is the component responsible for the final answer selection. For each of the eight multiple-choice options in a QASC question, the reader's input consists of the original question, the retrieved evidence sentences, and the text of that specific answer choice. The reader assigns a score to each choice, indicating the probability of that answer is correct. The AORO retriever is paired with three distinct reader models against corresponding baselines. First, for a fair comparison with the Two-step IR baseline, a bert-large-cased reader was utilized, achieving 76.1% accuracy representing a 2.9% improvement over the baseline's 73.2%. Second, to

align with more recent baselines such as SingleRR and SupA + QA, a RoBERTa-base reader was employed. This pairing reached 76.01% accuracy, surpassing the 73.9% baseline by 2.11%. Finally, to establish state-of-the-art performance, the AORO retriever was combined with a powerful DeBERTa-large reader, attaining 80.3% accuracy and outperforming the other baselines. These results demonstrate both the standalone improvement of the AORO retriever against comparable baselines and its maximum potential when paired with a powerful reader model.

Table 3. AORO's performance on MultiRC. "RoBERTa on DeBERTa" indicates that AORO uses the RoBERTa PTM for training with the order derived from DeBERTa. The other models utilize their own PTMs and orders.

Methods	Hop(s)	Precision	Recall	F1
No order				
Entire passage [31]	-	17.4	100.0	29.6
BM25 [41]	Single	43.8	61.2	51.0
AutoROCC [41]	Single	48.2	68.2	56.4
RoBERTa-retriever (All passages) [31]	Single	63.4	61.1	62.3
SingleRR (RoBERTa) [35]	Multiple	-	-	64.0
AIR top chain [31]	Multiple	66.2	63.1	64.2
Optimized order				
RPA (RoBERTa) [14]	Multiple	60.2	69.7	64.6
AORO (RoBERTa)	Multiple	61.1	70.8	65.6
AORO (RoBERTa on DeBERTa)	Multiple	61.5	71.7	66.2
AORO (DeBERTa)	Multiple	63.8	73.5	68.3

5.2. Quantitative Analysis of sim

To further investigate the impact of reasoning order on search performance, some reasoning orders are randomized to create Order x. The similarity is then calculated $\hat{F}^{\text{Order x}}$ and $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$, denoted as $sim(\hat{F}^{\text{Order x}}, \hat{F}^{\text{AORO}}_{\text{RoBERTa}})$. The similarity value ranges from 0 to 100, with increments of 10. When using RoBERTa as PTM, the training results for QASC are illustrated in Figure 3.

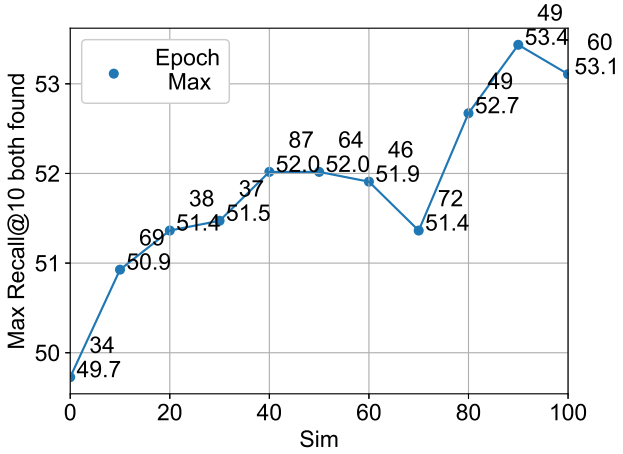


Figure 3. $sim(\hat{F}^{\text{Order x}}, \hat{F}^{\text{AORO}}_{\text{RoBERTa}})$ refers to the similarity between Order x and the AORO order using PTM of RoBERTa. In the figure, this is abbreviated as "sim". During the training process with Order x, the RoBERTa achieves the "max" Recall@10 *both found* on the QASC at a specific "epoch".

In the figure, as *sim* increases, Recall@10 *both found* shows an overall upward trend. When *sim* is at 70, there is a brief drop in Recall, which then continues to rise. Surprisingly, Recall reaches 53.4 when *sim* is at 90, exceeding the 53.1 observed when *sim* is at 100. This

phenomenon, where some results exceed those obtained with a *sim* value of 100, also occurs in the quantitative experiments with DeBERTa's *sim*. This suggests that there exists a better order of reasoning than $\hat{F}^{\text{AORO}}_{\text{DeBERTa}}$. These observations align with the last hypothesis regarding weaker models and their effective ordering discussed in Section 5.1. For a more detailed analysis of the relationship between *sim* and results, please refer to Section 5.3.

5.3. Qualitative Analysis

Figure 4a,b illustrates the *sim* between orders in the QASC and MultiRC datasets, respectively. The figures are examined from three perspectives: the absolute magnitude of *sim*, the reasons behind the differences in *sim*, and the implications of these differences. The details are presented below:

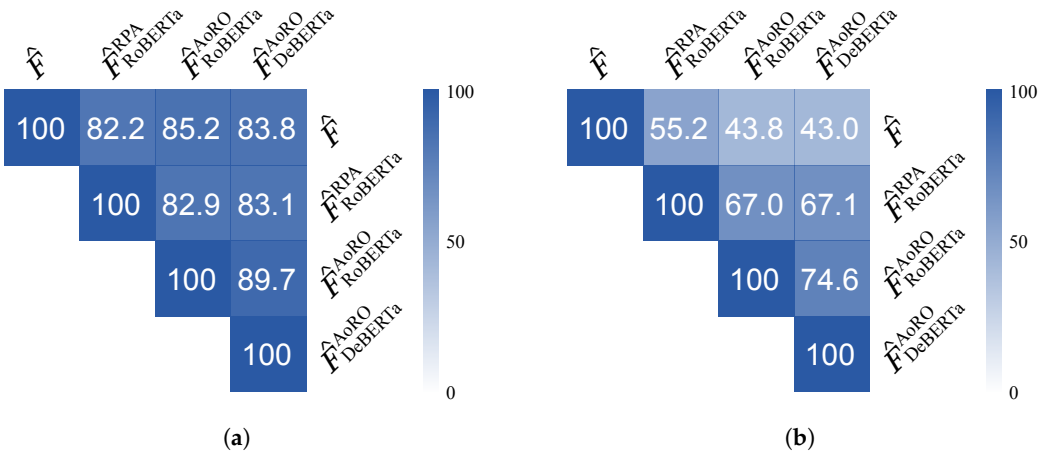


Figure 4. *sim* among the orders obtained through various methods and PTMs. (**a**) QASC. (**b**) MultiRC.

The greater the $sim(\hat{F}^{\text{Order x}}, \hat{F}^{\text{Excellent}})$, the better the performance. The reasoning order that has achieved the highest result achieved so far is $\hat{F}^{\text{AORO}}_{\text{DeBERTa}}$, which is referred to as $\hat{F}^{\text{Excellent order}}$ in this paper. Therefore, the larger the $sim(\hat{F}^{\text{Order x}}, \hat{F}^{\text{AORO}}_{\text{DeBERTa}})$, the higher the metric value. Specifically, Figure 4 shows that $sim(\hat{F}^{\text{AORO}}_{\text{DeBERTa}}, \hat{F}^{\text{AORO}}_{\text{DeBERTa}}) > sim(\hat{F}^{\text{AORO}}_{\text{RoBERTa}}, \hat{F}^{\text{AORO}}_{\text{DeBERTa}}) > sim(\hat{F}^{\text{RPA}}_{\text{RoBERTa}}, \hat{F}^{\text{AORO}}_{\text{DeBERTa}})$. This order corresponds to the Recall values in Table 2, indicating that AORO (DeBERTa) > AORO (RoBERTa) > RPA (RoBERTa). A similar pattern is observed for F1 in Table 3. Additionally, since $sim(\hat{F}^{\text{AORO}}_{\text{DeBERTa}}, \hat{F}^{\text{AORO}}_{\text{DeBERTa}}) > sim(\hat{F}^{\text{AORO}}_{\text{RoBERTa}}, \hat{F}^{\text{AORO}}_{\text{DeBERTa}})$, it can be concluded that AORO (RoBERTa on DeBERTa) > AORO (RoBERTa), as shown in Table 2 on QASC.

The greater the number of facts, the larger the *sim* difference. In the QASC datasets, there are always two facts, whereas in MultiRC, there can be up to six facts in the training set. As discussed in the permutation paragraph of Section 3.2.1, QASC has 2! (which equals to 2) possible orderings, while MultiRC has 6! (which equals to 720) possible orderings. More possible orderings lead to a lower probability that the orders will be identical, resulting in a greater similarity difference. This is reflected in Figure 4a, where the *sim* generally shows little difference, while in Figure 4b, the *sim* difference is more pronounced.

The greater the *sim* difference, the wider the variation in performance. As shown in Figure 4a (QASC), the *sim* values are generally greater than 80, resulting in comparable Recall@10 scores. This means that Recall@10 is quite similar for RPA (RoBERTa), AORO (RoBERTa), and AORO (RoBERTa on DeBERTa), as seen in Table 2. The most significant improvement is observed with AORO (DeBERTa), where Recall increases by 1.0. However, in the case of MultiRC (Table 3), the maximum *sim* between the optimization orders is $sim(\hat{F}^{\text{AORO}}_{\text{RoBERTa}}, \hat{F}^{\text{AORO}}_{\text{DeBERTa}}) = 76.4$ (Figure 4b). Here, the F1 score of AORO (DeBERTa) increases by 1.9 compared to AORO (RoBERTa on DeBERTa), indicating a more substantial improvement.

5.4. Regularities in Reasoning Orders

There is a notable empirical pattern in AORO: the similarity value $sim(\hat{F}^{\text{AORO}}_{\text{RoBERTa}}, \hat{F}^{\text{AORO}}_{\text{DeBERTa}})$ is consistently large both in QASC (89.7) and MultiRC (74.6), as depicted in Figure 4. After the trained retrieval model automatically selects the reasoning order by itself (AORO), the orders generated by the two PTMs with different structures consistently exhibit higher similarities. This observation suggests that, within the conducted experiments, **the preferred reasoning order could be relatively insensitive to the choice of PTM.** Moreover, high *sim* values are observed across datasets, indicating that **the preferred order may also be robust to dataset variation.** This consistency can be referred to as regularities in reasoning orders.

In this section, training-free methods are employed to generate reasoning orders and calculate their similarities with $\hat{F}^{\text{Excellent}}$ (as referenced by $\hat{F}^{\text{AORO}}_{\text{DeBERTa}}$) to explore the relevant factors underlying these empirical regularities. The four training-free methods include TF-IDF, RPA, GloVe, and spaCy, which can be categorized into two groups:

- **TF-IDF, RPA, GloVe:** First, Q and $\hat{f}$s are processed in three steps: (i) they are split using the `word_tokenize` from NLTK (Natural Language ToolKit) as tokenizer, (ii) stop words are removed, and (iii) stems are extracted using the Porter Stemmer. This results in lists of words that are used for similarity calculations. The ranking process involves the following steps: (i) identifying the fact $\hat{f}_i$ that has the highest sentence similarity with Q, (ii) removing overlapping words between Q and $\hat{f}_i$, and (iii) using the remaining words as a new Q to compute similarity with the remaining $\hat{f}$s. This ranking process continues iteratively until no $\hat{f}$s remain.
- **spaCy:** When using spaCy's similarity method, it is important to note that once the sentence is split into individual words, spaCy cannot compute inter-sentence similarity. As a result, the new Q is created by combining the old Q and the retrieved $\hat{f}$.

The TF-IDF score for the i-th fact in $\hat{F}$ is calculated as follows:

$$\text{TF-IDF}_i = \sum_{word \in Q} \text{TF}(word, \hat{f}_i) \times \text{IDF}_{word} \tag{10}$$

Here, TF refers to the Term Frequency of $word$ in $\hat{f}_i$, and IDF_{word} is the standard Inverse Document Frequency, which remains constant despite the reconfiguration of Q. Thus, a higher TF-IDF value for $\hat{f}$ indicates that it contains more important words that are present in Q, leading to a higher ranking. Additionally, the dependence on GloVe to calculate scores in AIR is illustrated in Figure 5.

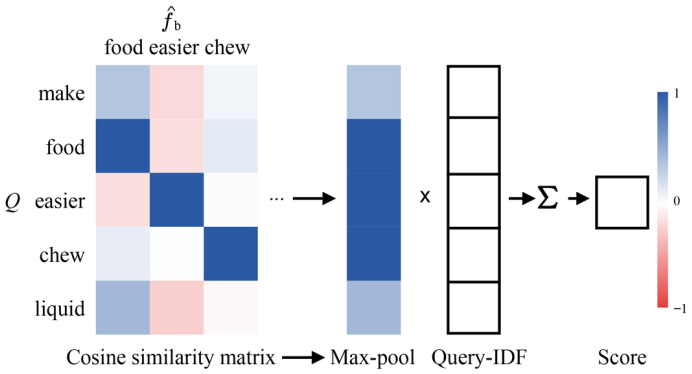


Figure 5. The process in AIR utilizes GloVe to calculate similarity scores.

Specifically, the cosine similarity matrix is computed based on the word vectors from GloVe, and the final Query-IDF $\text{IDF}^{\text{GloVe}}_{word}$ is a modified version of IDF_{word}, defined as follows:

$$\text{IDF}^{\text{GloVe}}_{word} = \log \frac{|Q + \hat{F}| - |word \in Q, \hat{F}| + 0.5}{|word \in Q, \hat{F}| + 0.5}. \tag{11}$$

In this equation, $|Q + \hat{F}|$ denotes the total number of texts of Q and $\hat{F}$, and $word \in Q, \hat{F}$ refers to the texts containing the specified $word$.

After determining the reasoning order using a training-free method, $sim(\hat{F}^{\text{Training-free}}, \hat{F}^{\text{Excellent}})$ is calculated, followed by using DeBERTa to evaluate the performance based on $\hat{F}^{\text{Training-free}}$. The results are presented in Table 4.

Table 4. $sim(\hat{F}^{\text{Training-free}}, \hat{F}^{\text{Excellent}})$ and the critical metrics for training with DeBERTa.

Training-Free Method	QASC *sim*/Recall@10 Both Found	MultiRC *sim*/F1
AORO (DeBERTa)	100/54.3	100/68.3
TF-IDF	82.2/52.0 (↓2.3)	67.2/64.5 (↓3.8)
RPA	83.1/52.7 (↓1.6)	67.1/64.6 (↓3.7)
GloVe	78.0/51.6 (↓2.7)	65.7/64.3 (↓4.0)
spaCy	74.1/51.7 (↓2.6)	59.7/63.5 (↓4.8)

In Table 4, the least reduction in $\hat{F}^{\text{Excellent}}$ is achieved by RPA, with *sim*s of 83.1 and 67.1 on QASC and MultiRC, respectively, followed by TF-IDF. In contrast, GloVe and spaCy demonstrate relatively poorer performance. Although zero reduction of $\hat{F}^{\text{Excellent}}$ was not achieved, it appears that lexicon-level statistical methods currently yield better order-optimization results than word vectors. Therefore, it is not yet feasible to fully replace the automatic optimization process of AORO with a training-free approach. Further exploration of the regularities in reasoning orders is still necessary.

5.5. Case Studies

This section analyzes cases from the QASC and MultiRC datasets to evaluate the results of reasoning-order optimization, as presented in Table 5.

In the QASC case, the naive reasoning order is $\hat{F} = Q \xrightarrow{\text{P}} \hat{f}_1 \xrightarrow{\text{P}} \hat{f}_2$. After optimization, all methods adopt the new order $Q \xrightarrow{\text{P}} \hat{f}_2 \xrightarrow{\text{P}} \hat{f}_1$, aligning with the observation that all similarity scores on QASC are greater than 80. However, as noted in Section 5.3, when $|\hat{F}|$ is small, the differences in *sim* scores are minimal. The consistently high similarity scores and uniform orders obscure the analysis of the strengths and weaknesses of the different methods. To illustrate the variations more effectively, the subsequent analysis focuses on a MultiRC sample with $|\hat{F}| \geq 4$, which includes the query Q and its four corresponding ground-truth facts.

In the MultiRC sample, both $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$ and $\hat{F}^{\text{AORO}}_{\text{DeBERTa}}$ follow the order $Q \xrightarrow{\text{P}} \hat{f}_1 \xrightarrow{\text{P}} \hat{f}_4 \xrightarrow{\text{P}} \hat{f}_2 \xrightarrow{\text{P}} \hat{f}_3$, while the training-free methods, TF-IDF and RPA, also produce the same order ($sim = 100$). In contrast, GloVe and spaCy produce entirely different orders, resulting in similarity scores of 0. The notable discrepancies in similarity appear to stem from the specific sample. Indeed, when analyzing all samples with $|\hat{F}| \geq 4$, it is observed that as the number of reasoning steps increases, the similarities for TF-IDF (50.0) and RPA (51.2) significantly surpass those of GloVe (45.5) and spaCy (42.4). This suggests that the training-free methods TF-IDF and RPA are more effective in optimizing the inference order for long sequences.

Table 5. Analysis of QASC and MultiRC cases. Possible reasoning connections between search sources and targets are included.

QASC	Q **with 2 Ground-Truth Facts**
Q	Query: What are invaluable for soil health? Choice: annelids.
$\hat{f}_1$	Fact: Annelids are worms such as the familiar earthworm.
$\hat{f}_2$	Fact: Earthworms are invaluable for soil health.
	$\hat{F}^{\text{AORO}}$ (? $\xrightarrow{\text{Reasoning connection}}$?)
$\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$	$Q \xrightarrow{\text{P}} \hat{f}_2 \xrightarrow{\text{P}} \hat{f}_1$ ($Q \xrightarrow{\text{What are "invaluable for soil health?" "Earthworms."}} \hat{f}_2$, $Q + \hat{f}_2 \xrightarrow{\text{Are "earthworms" related to "annelids?"}} \hat{f}_1$)
$\hat{F}^{\text{AORO}}_{\text{DeBERTa}}$	$Q \xrightarrow{\text{P}} \hat{f}_2 \xrightarrow{\text{P}} \hat{f}_1$ (same with $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$)
	$\hat{F}^{\text{Training-free}}$
TF-IDF	$Q \xrightarrow{\text{P}} \hat{f}_2 \xrightarrow{\text{P}} \hat{f}_1$ (same with $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$)
RPA	$Q \xrightarrow{\text{P}} \hat{f}_2 \xrightarrow{\text{P}} \hat{f}_1$ (same with $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$)
GloVe	$Q \xrightarrow{\text{P}} \hat{f}_2 \xrightarrow{\text{P}} \hat{f}_1$ (same with $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$)
spaCy	$Q \xrightarrow{\text{P}} \hat{f}_2 \xrightarrow{\text{P}} \hat{f}_1$ (same with $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$)
MultiRC	Q **with 4 Ground-Truth Facts**
Q	Query: Why did the writer think he was going crazy? Choice: He heard a sound and saw a light in a vacuum.
$\hat{f}_1$	Fact: Hotel California My first thought: I was going crazy.
$\hat{f}_2$	Fact: Twenty-four hours of silence (vacuum, remember); was I hallucinating noises now?
$\hat{f}_3$	Fact: It was a fine bell, reminiscent of ancient stone churches and the towering cathedrals I'd seen in documentaries.
$\hat{f}_4$	Fact: And accompanying the bell, I saw a light.
	$\hat{F}^{\text{AORO}}$ (? $\xrightarrow{\text{Reasoning connection}}$?)
$\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$	$Q \xrightarrow{\text{P}} \hat{f}_1 \xrightarrow{\text{P}} \hat{f}_4 \xrightarrow{\text{P}} \hat{f}_2 \xrightarrow{\text{P}} \hat{f}_3$($Q \xrightarrow{\text{Did he "think" he is "going crazy"? Yes.}} \hat{f}_1, Q \xrightarrow{\text{Did he "see a light?" Yes.}} \hat{f}_4, Q \xrightarrow{\text{Why "thought" "crazy?" "Hallucinating noises."}} \hat{f}_2, Q + \hat{f}_4 \xrightarrow{\text{Are "sounds" or "noises" related to "bell?"}} \hat{f}_3$)
$\hat{F}^{\text{AORO}}_{\text{DeBERTa}}$	$Q \xrightarrow{\text{P}} \hat{f}_1 \xrightarrow{\text{P}} \hat{f}_4 \xrightarrow{\text{P}} \hat{f}_2 \xrightarrow{\text{P}} \hat{f}_3$ (same with $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$)
	$\hat{F}^{\text{Training-free}}$ / For samples whose $\lvert\hat{F}\rvert \geq 4$, $sim(\hat{F}^{\text{Training-free}}, \hat{F}^{\text{AORO}}_{\text{DeBERTa}})$
TF-IDF	$Q \xrightarrow{\text{P}} \hat{f}_1 \xrightarrow{\text{P}} \hat{f}_4 \xrightarrow{\text{P}} \hat{f}_2 \xrightarrow{\text{P}} \hat{f}_3$ (same with $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$)/50.0
RPA	$Q \xrightarrow{\text{P}} \hat{f}_1 \xrightarrow{\text{P}} \hat{f}_4 \xrightarrow{\text{P}} \hat{f}_2 \xrightarrow{\text{P}} \hat{f}_3$ (same with $\hat{F}^{\text{AORO}}_{\text{RoBERTa}}$)/51.2
GloVe	$Q \xrightarrow{\text{P}} \hat{f}_2 \xrightarrow{\text{P}} \hat{f}_3 \xrightarrow{\text{P}} \hat{f}_4 \xrightarrow{\text{P}} \hat{f}_1$/45.5
spaCy	$Q \xrightarrow{\text{P}} \hat{f}_4 \xrightarrow{\text{P}} \hat{f}_3 \xrightarrow{\text{P}} \hat{f}_1 \xrightarrow{\text{P}} \hat{f}_2$/42.4

In terms of content, the MultiRC sample demonstrates that **AORO** (DeBERTa) **initiates each reasoning process from Q and systematically analyzes and verifies each detail within it, addressing Q and ultimately clarifying the relationships among these details.** The reasoning steps undertaken by AORO are as follows:

- Confirm the query `going crazy` ($Q \xrightarrow{\text{P}} \hat{f}_1$).
- Validate the choice `see a light` ($Q \xrightarrow{\text{P}} \hat{f}_4$).
- Investigates the cause of `crazy`, identifying it as `hallucinating noises` ($Q \xrightarrow{\text{P}} \hat{f}_2$).
- Combines Q and $\hat{f}_4$ to establish the relationship that `sound` comes from `bell` ($Q + \hat{f}_4 \xrightarrow{\text{P}} \hat{f}_3$).

This sequence aligns with a logical thinking order: **first confirming the query, then answering the question, and finally providing additional details**. Looking back at the QASC sample, it can be observed that AORO (DeBERTa) follows a similar approach: it initially seeks the conceptual definition of the query (clarifying what it is) before supplementing the remaining content.

6. Conclusions

This paper introduces AORO, a method designed to automatically optimize the reasoning order for multi-hop QA. AORO operates by iteratively selecting the most appropriate target at each inference step, constructing an optimized reasoning order for each sample. Its performance is validated on the QASC and MultiRC datasets. The results demonstrate that the method improves upon strong baselines with the same PTM, while the synergy with advanced PTMs yields further gains of up to 1.6 points in Recall@10 and 3.7 points in F1 score. Additionally, the conducted experiments reveal that the optimized reasoning orders generated by AORO, based on various pre-trained models, exhibit a high degree of similarity across datasets. This suggests that the reasoning process may follow a stable, model- and dataset-agnostic principle. While the implementation of training-free methods to approximate the optimized order still requires improvement, a consistent reasoning pattern is observed: verifying the authenticity of the question content, answering the question, and providing supplementary details. Future work will aim to utilize more advanced models to achieve even better reasoning orders and to develop more effective training-free methods for approximating these orders. Future work will also seek to explore the fundamental factors that determine the optimal reasoning order.

Author Contributions: Conceptualization, S.L., Z.C., K.B. and Z.J.; Data curation, S.L. and Z.C.; Formal analysis, S.L. and Z.C.; Funding acquisition, S.L.; Investigation, Z.C.; Methodology, S.L., Z.C., K.B. and Z.J.; Project administration, S.L.; Resources, Z.C.; Software, Z.C.; Supervision, S.L.; Validation, S.L. and Z.J.; Visualization, S.L. and Z.C.; Writing—original draft, S.L. and Z.C.; Writing—review and editing, S.L. and Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under Grant No. 62406087, Natural Science Foundation of Shandong Province under Grant No. ZR2024QF139, and State Key Laboratory of Processors (ICT, CAS) under Grant No. CLQ202406.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Talmor, A.; Herzig, J.; Lourie, N.; Berant, J. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; Volume 1 (Long and Short Papers); Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4149–4158. [CrossRef]
2. Abdalla, M.; Kasem, M.; Mahmoud, M.; Yagoub, B.; Senussi, M.; Abdallah, A.; Kang, S.; Kang, H. ReceiptQA: A Question-Answering Dataset for Receipt Understanding. *Mathematics* **2025**, *13*, 1760. [CrossRef]
3. Sha, Y.; Feng, Y.; He, M.; Liu, S.; Ji, Y. Retrieval-Augmented Knowledge Graph Reasoning for Commonsense Question Answering. *Mathematics* **2023**, *11*, 3269. [CrossRef]
4. Xiao, G.; Liao, J.; Tan, Z.; Yu, Y.; Ge, B. Hyperbolic Directed Hypergraph-Based Reasoning for Multi-Hop KBQA. *Mathematics* **2022**, *10*, 3905. [CrossRef]

5. Tang, Y.; Ng, H.T.; Tung, A.K.H. Do multi-hop question answering systems know how to answer the single-hop sub-questions? In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, 19–23 April 2021; Merlo, P., Tiedemann, J., Tsarfaty, R., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 3244–3249. [CrossRef]
6. Feldman, Y.; El-Yaniv, R. Multi-hop paragraph retrieval for open-domain question answering. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; Korhonen, A., Traum, D.R., Màrquez, L., Eds.; Volume 1: Long Papers; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 2296–2309. [CrossRef]
7. Das, R.; Godbole, A.; Kavarthapu, D.; Gong, Z.; Singhal, A.; Yu, M.; Guo, X.; Gao, T.; Zamani, H.; Zaheer, M.; et al. Multi-step entity-centric information retrieval for multi-hop question answering. In Proceedings of the 2nd Workshop on Machine Reading for Question Answering, MRQA@EMNLP 2019, Hong Kong, China, 4 November 2019; Fisch, A., Talmor, A., Jia, R., Seo, M., Choi, E., Chen, D., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 113–118. [CrossRef]
8. Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.W.; Salakhutdinov, R.; Manning, C.D. Hotpotqa: A dataset for diverse, explainable multi-hop question Answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 2369–2380. [CrossRef]
9. Lee, H.; Yang, S.; Oh, H.; Seo, M. Generative multi-hop retrieval. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; Goldberg, Y., Kozareva, Z., Zhang, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 1417–1436. [CrossRef]
10. Ho, X.; Nguyen, A.D.; Sugawara, S.; Aizawa, A. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), 8–13 December 2020; Scott, D., Bel, N., Zong, C., Eds.; International Committee on Computational Linguistics: Barcelona, Spain, 2020; pp. 6609–6625. [CrossRef]
11. Lee, K.; Hwang, S.; Han, S.; Lee, D. Robustifying multi-hop QA through pseudo-evidentiality training. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, 1–6 August 2021; Zong, C., Xia, F., Li, W., Navigli, R., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 6110–6119. [CrossRef]
12. Su, D.; Xu, P.; Fung, P. QA4QG: Using question answering to constrain multi-hop question generation. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23–27 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 8232–8236. [CrossRef]
13. Li, S.; Li, X.; Shang, L.; Jiang, X.; Liu, Q.; Sun, C.; Ji, Z.; Liu, B. Hopretriever: Retrieve hops over wikipedia to answer complex questions. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, 2–9 February 2021; AAAI Press: Washington, DC, USA, 2021; pp. 13279–13287. [CrossRef]
14. Cao, Z.; Liu, B.; Li, S. RPA: Reasoning path augmentation in iterative retrieving for multi-hop QA. In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, 7–14 February 2023; Williams, B., Chen, Y., Neville, J., Eds.; AAAI Press: Washington, DC, USA, 2023; pp. 12598–12606. [CrossRef]
15. Asai, A.; Hashimoto, K.; Hajishirzi, H.; Socher, R.; Xiong, C. Learning to retrieve reasoning paths over wikipedia graph for question answering. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020; OpenReview.net: Alameda, CA, USA, 2020. Available online: https://openreview.net/forum?id=SJgVHkrYDH (accessed on 19 May 2020).
16. Das, R.; Dhuliawala, S.; Zaheer, M.; McCallum, A. Multi-step retriever-reader interaction for scalable open-domain question answering. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019; OpenReview.net: Alameda, CA, USA, 2019. Available online: https://openreview.net/forum?id=HkfPSh05K7 (accessed on 9 May 2019).
17. Xiong, W.; Li, X.L.; Iyer, S.; Du, J.; Lewis, P.S.H.; Wang, W.Y.; Mehdad, Y.; Yih, S.; Riedel, S.; Kiela, D.; et al. Answering complex open-domain questions with multi-hop dense retrieval. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021; OpenReview.net: Alameda, CA, USA, 2021. Available online: https://openreview.net/forum?id=EMHoBG0avc1 (accessed on 6 May 2021).
18. Zhang, J.; Zhang, X.; Yu, J.; Tang, J.; Tang, J.; Li, C.; Chen, H. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022, Dublin, Ireland, 22–27 May 2022; Muresan, S., Nakov, P., Villavicencio, A., Eds.; Volume 1: Long Papers; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 5773–5784. [CrossRef]

19. Bai, Y.; Lv, X.; Li, J.; Hou, L.; Qu, Y.; Dai, Z.; Xiong, F. SQUIRE: A sequence-to-sequence framework for multi-hop knowledge graph reasoning. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; Goldberg, Y., Kozareva, Z., Zhang, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 1649–1662. [CrossRef]
20. Piekos, P.; Malinowski, M.; Michalewski, H. Measuring and improving bert's mathematical abilities by predicting the order of reasoning. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, 1–6 August 2021; Zong, C., Xia, F., Li, W., Navigli, R., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 383–394. [CrossRef]
21. Khot, T.; Clark, P.; Guerquin, M.; Jansen, P.; Sabharwal, A. QASC: A dataset for question answering via sentence composition. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, 7–12 February 2020; AAAI Press: Washington, DC, USA, 2020; pp. 8082–8090. [CrossRef]
22. Bengio, Y.; Louradour, J.; Collobert, R.; Weston, J. Curriculum Learning. In Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, QC, Canada, 14–18 June 2009; Danyluk, A.P., Bottou, L., Littman, M.L., Eds.; ICML: San Diego, CA, USA, 2009; pp. 41–48. [CrossRef]
23. Fang, Y.; Sun, S.; Gan, Z.; Pillai, R.; Wang, S.; Liu, J. Hierarchical graph network for multi-hop question answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November 2020; Webber, B., Cohn, T., He, Y., Liu, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 8823–8838. [CrossRef]
24. Mavi, V.; Jangra, A.; Jatowt, A. A survey on multi-hop question answering and generation. *arXiv* **2022**. [CrossRef]
25. Wu, C.; Hu, E.; Zhan, K.; Luo, L.; Zhang, X.; Jiang, H.; Wang, Q.; Cao, Z.; Yu, F.; Chen, L. Triple-fact retriever: An explainable reasoning retrieval model for multi-hop QA problem. In Proceedings of the 38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, 9–12 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1206–1218. [CrossRef]
26. Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; Deng, L. MS MARCO: A human generated machine reading comprehension dataset. In Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 9 December 2016; Besold, T.R., Bordes, A., d'Avila Garcez, A.S., Wayne, G., Eds.; Ser. CEUR Workshop Proceedings; CEUR-WS.org: Aachen, Germany, 2016; Volume 1773. Available online: https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf (accessed on 10 December 2023).
27. Yang, Y.; Yih, W.; Meek, C. Wikiqa: A challenge dataset for open-domain question answering. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, 17–21 September 2015; Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., Marton, Y., Eds.; The Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 2013–2018. [CrossRef]
28. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The long-document Transformer. *arXiv* **2020**. [CrossRef]
29. Ainslie, J.; Onta nón, S.; Alberti, C.; Cvicek, V.; Fisher, Z.; Pham, P.; Ravula, A.; Sanghai, S.; Wang, Q.; Yang, L. ETC: Encoding long and structured inputs in transformers. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November 2020; Webber, B., Cohn, T., He, Y., Liu, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 268–284. [CrossRef]
30. Khashabi, D.; Chaturvedi, S.; Roth, M.; Upadhyay, S.; Roth, D. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, LA, USA, 1–6 June 2018; Walker, M.A., Ji, H., Stent, A., Eds.; Volume 1 (Long Papers); Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 252–262. [CrossRef]
31. Yadav, V.; Bethard, S.; Surdeanu, M. Unsupervised alignment-based iterative evidence retrieval for multi-hop question answering. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020; Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 4514–4525. [CrossRef]
32. Yadav, V.; Bethard, S.; Surdeanu, M. Alignment over Heterogeneous Embeddings for Question Answering. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Volume 1 (Long And hort Papers); Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 2681–2691. [CrossRef]

33. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, 25–29 October 2014; Moschitti, A., Pang, B., Daelemans, W., Eds.; A Meeting of SIGDAT, a Special Interest Group of the ACL; ACL: Brooklyn, NY, USA, 2014; pp. 1532–1543. [CrossRef]
34. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; Volume 1 (Long and Short Papers); Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4171–4186. [CrossRef]
35. Yadav, V.; Bethard, S.; Surdeanu, M. If you want to go far go together: Unsupervised joint candidate evidence retrieval for multi-hop question Answering. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, 6–11 June 2021; Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., Zhou, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 4571–4581. [CrossRef]
36. Zhang, J.; Zhang, H.; Zhang, D.; Liu, Y.; Huang, S. End-to-end beam retrieval for multi-hop question answering. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, 16–21 June 2024; Duh, K., Gómez-Adorno, H., Bethard, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 1718–1731. [CrossRef]
37. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *arXiv* **2019**. [CrossRef]
38. Xu, W.; Deng, Y.; Zhang, H.; Cai, D.; Lam, W. Exploiting reasoning chains for multi-hop science question answering. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event/Punta Cana, Dominican Republic, 16–20 November 2021; Moens, M., Huang, X., Specia, L., Yih, S.W., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 1143–1156. [CrossRef]
39. Khattab, O.; Potts, C.; Zaharia, M.A. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. In Proceedings of the Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, Virtual, 6–14 December 2021; Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W., Eds.; NeurIPS: San Diego CA, USA, 2021; pp. 27670–27682. Available online: https://proceedings.neurips.cc/paper/2021/hash/e8b1cbd05f6e6a358a81dee52493dd06-Abstract.html (accessed on 20 December 2021).
40. Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. Superglue: A stickier benchmark for general-purpose language understanding systems. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R., Eds.; NeurIPS: San Diego CA, USA, 2019; pp. 3261–3275. Available online: https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html (accessed on 20 December 2020).
41. Yadav, V.; Bethard, S.; Surdeanu, M. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop 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 2019, Hong Kong, China, 3–7 November 2019; Inui, K., Jiang, J., Ng, V., Wan, X., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 2578–2589. [CrossRef]
42. He, P.; Liu, X.; Gao, J.; Chen, W. Deberta: Decoding-enhanced bert with disentangled attention. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021; OpenReview.net: Alameda, CA, USA, 2021. Available online: https://openreview.net/forum?id=XPZIaotutsD (accessed on 7 May 2021).
43. You, Y.; Li, J.; Reddi, S.J.; Hseu, J.; Kumar, S.; Bhojanapalli, S.; Song, X.; Demmel, J.; Keutzer, K.; Hsieh, C. Large batch optimization for deep learning: Training BERT in 76 minutes. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020; OpenReview.net: Alameda, CA, USA, 2020. Available online: https://openreview.net/forum?id=Syx4wnEtvH (accessed on 30 May 2020).
44. Xiong, L.; Xiong, C.; Li, Y.; Tang, K.; Liu, J.; Bennett, P.N.; Ahmed, J.; Overwijk, A. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021; OpenReview.net: Alameda, CA, USA, 2021. Available online: https://openreview.net/forum?id=zeFrfgyZln (accessed on 8 May 2021).
45. Sammut, C.; Webb, G.I. (Eds.) *TF–IDF*; Springer: Boston, MA, USA, 2010; pp. 986–987. [CrossRef]

46. Ferguson, J.; Hajishirzi, H.; Dasigi, P.; Khot, T. Retrieval Data Augmentation Informed by Downstream Question Answering Performance. In Proceedings of the Fifth Fact Extraction And VERification Workshop (FEVER), Dublin, Ireland, 26 May 2022; pp. 1–5. Available online: https://aclanthology.org/2022.fever-1.1/ (accessed on 28 May 2022).
47. Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; Huang, X. Pre-trained models for natural language processing: A survey. *arXiv* **2020**. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.