Retrieving Support to Rank Answers in Open-Domain Question Answering

Zeyu Zhang, Alessandro Moschitti, and **Thuy Vu**
Amazon AGI, El Segundo, CA, USA
{zeyzhan, amosch, thuyvu}@amazon.com

Abstract

We introduce a novel Question Answering (QA) architecture that enhances answer selection by retrieving targeted supporting evidence. Unlike traditional methods, which retrieve documents or passages relevant only to a query q, our approach retrieves content relevant to the combined pair (q, a), explicitly emphasizing the supporting relation between the query and a candidate answer a. By prioritizing this relational context, our model effectively identifies paragraphs that directly substantiate the correctness of a with respect to q, leading to more accurate answer verification than standard retrieval systems. Our neural retrieval method also scales efficiently to collections containing hundreds of millions of paragraphs. Moreover, this approach can be used by large language models (LLMs) to retrieve explanatory paragraphs that ground their reasoning, enabling them to tackle more complex QA tasks with greater reliability and interpretability.

1 Introduction

Question Answering (QA) systems are increasingly expected to deliver accurate and reliable information; however, recent generative QA technologies have shown vulnerability to hallucinations. Consequently, methods to verify the factuality of generated answers have become essential. Previous research on automatic claim verification, such as the Fact Extraction and VERification (FEVER) challenge (Thorne et al., 2018), indicates that retrieving textual evidence—sentences supporting or refuting a claim—is critical to assess answer correctness. Previous works, e.g., (Bajaj et al., 2016; Zhang et al., 2022, 2021), typically apply standard lexical or neural retrieval techniques (Karpukhin et al., 2020) to collect such evidence. For instance, Dense Passage Retrieval (DPR) is commonly queried using the concatenation of a question q and an answer candidate a. Although straightforward and effective, this approach is inherently limited because DPR is trained primarily to retrieve documents relevant to the query terms. Thus, top-ranked documents typically exhibit high lexical overlap with q and a, but do not necessarily provide evidence confirming the correctness of a with respect to q.

q:	**When does season 6 of the next step start?**
a:	The series has been renewed for a sixth season of 26 episodes which will premiere in 2018.
s_1:	"The Next Step" is a Canadian teen drama series. This is a list of the characters, and who portrays them.
s_2:	On March 21, 2016, Frank van Keeken announced on Instagram that "The Next Step" would return for a fifth season, which premiered on May 26, 2017.
s_3:	The series has been renewed for a sixth season of 26 episodes which premiered in Canada on September 29, 2018.
s_4:	"The Next Step" is filmed at Filmport Presentation Centre, Toronto. Exterior and street shots were shot on location in Downtown Toronto.

Table 1: A question with answer candidates.

Table 1 illustrates this limitation by presenting a question, a candidate answer, and four potential supporting sentences (s_1–s_4). The query asks about the start date of Season 6 of *The Next Step*. The provided answer references the sixth season of a *series*. To logically confirm this answer, it is necessary to know that *The Next Step* is indeed a *series*. Sentence s_1 explicitly provides this critical information, despite also including irrelevant details. In contrast, sentences s_2 and s_4, although related to both q and a, fail to provide the essential missing link. Similarly, s_3 restates the answer without offering additional verification that *The Next Step* is a series. Crucially, s_2 and s_3 appear lexically *more relevant* to the query-answer pair, as they contain multiple overlapping terms such as *Canada*, *premiere*, and *sixth season*. This example clearly shows that effective evidence selection cannot rely solely on lexical or topical relevance; rather, it must explicitly capture the logical support relation between q and a.

This motivates retrieval architectures that move

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 35098–35105

beyond topical similarity to explicitly capture the support relation between a question, an answer, and evidence. This paper introduces an innovative QA architecture incorporating support retrieval and re-ranking models explicitly trained to identify relevant support relations. We formulate supervised methods to model these relations by training ranking functions on triples (q, a, s), where s represents the supporting text. Importantly, we demonstrate how neural retrieval can leverage this reranking training data by using a classification head analogous to DPR's dot-product structure ($\vec{Q} \cdot \vec{s}$), where $\vec{Q}$ encodes the query-answer pair (q, a) and $\vec{s}$ encodes the supporting evidence. In terms of efficiency, the computational complexity of our Dense Support Retrieval (DSR) module is the same as that of standard neural IR retrievers or equivalent verification systems. The only additional cost arises from verifying each candidate answer, as evidence retrieval is an inherent prerequisite for answer verification.

We evaluate the effectiveness of our support retrieval (i) implicitly, by measuring its impact on AS2, and (ii) explicitly, by calculating support recall. In doing so, our approach directly addresses answer and support granularity at the sentence or paragraph level, contributing to research on passage reranking and Answer Sentence Selection (AS2). Empirically, our Supervised Support Ranker (SSR) achieves state-of-the-art performance on the HotpotQA dataset (Yang et al., 2018), improving AS2 accuracy from 68.4% to 70.2% over the prior best approach (Zhang et al., 2023). Additionally, on the FocusQA dataset (Barlacchi et al., 2022), our Dense Support Retrieval (DSR) significantly improves retrieval performance compared to DPR, increasing support recall from 12% to 71.6%, and sets a new AS2 accuracy benchmark by improving the prior state-of-the-art from 28.7% to 33.1%.

Beyond efficiency and empirical gains, DSR enables fine-grained relational retrieval. By explicitly modeling explanatory relations between (q, a) pairs and supporting passages, our approach lays the foundation for specialized retrievers (e.g., causal, temporal, comparative) and provides a mechanism for grounding the reasoning of large language models (LLMs).

2 Background and Related Work

Retrieval-based QA systems, such as those pioneered by the TREC QA tracks (Voorhees and Tice, 1999), are conceptually straightforward. Given a question q, the system first retrieves the top-N relevant documents. These documents are then segmented into paragraphs or sentences, and an answer selector identifies the text most likely to contain the correct answer. Below, we summarize relevant advancements in neural retrieval and AS2.

Dense Passage Retrieval (DPR) Karpukhin et al. (2020) proposed DPR, an advanced neural retrieval method for QA systems. Unlike traditional sparse retrieval methods, which depend primarily on keyword matches, DPR leverages a dual-encoder architecture based on transformers. This approach generates dense vector representations of passages and queries, enabling more nuanced, semantic-based matching.

All passages in the corpus are encoded offline into dense vectors $\vec{p}$. Questions are encoded separately at inference time into vectors $\vec{q}$ within the same embedding space. The relevance of a passage p to a query q is determined by computing their similarity, typically as the scalar product $\vec{p} \cdot \vec{q}$. This similarity score is then used to rank passages efficiently.[1] Applications of DPR (Lewis et al., 2020; Borgeaud et al., 2022; Zhang et al., 2023) have demonstrated substantial improvements in retrieval effectiveness over traditional methods.

Answer Sentence Selection (AS2) The standard AS2 formulation is defined as follows: given a question $q \in \mathcal{Q}$ and a set of candidate answers $\mathcal{C}$, the goal is to learn a function $\pi : \mathcal{Q} \times \mathcal{C} \rightarrow \mathbb{R}$, where $\pi(q, c_i)$ outputs the likelihood that candidate $c_i \in \mathcal{C}$ correctly answers q.

Recent AS2 models rely on neural architectures to estimate π, such as convolutional neural networks (CNNs) proposed by Severyn and Moschitti (2015), CNNs with attention mechanisms like Compare-Aggregate (Yoon et al., 2019), and inter-weighted alignment networks (Shen et al., 2017). Bonadiman and Moschitti (2020) introduced several joint models surpassing earlier neural AS2 approaches. These, in turn, were further improved upon by transformer-based models such as TANDA (Garg et al., 2020). Current state-of-the-art approaches are ASR (Zhang et al., 2021) and DAR-DR (Zhang et al., 2023). Both employ transformer networks that leverage multiple candidate answers to estimate the

[1] For large repositories, exhaustive computation can be avoided through efficient KNN search (Johnson et al., 2019).

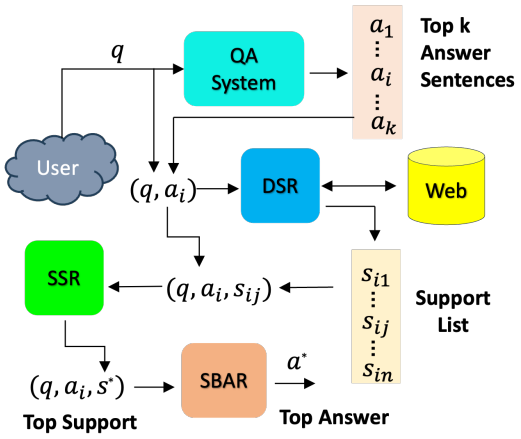

Figure 1: Our answer verification architecture.

probability of a given candidate c_i, computed as $\pi(c_i|c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_k)$.

Textual Entailment Finally, entailment methods test whether a hypothesis follows from a premise, while our support relation generalizes this idea to retrieving passages that justify a candidate answer with respect to a question. This broader framing situates our model within and beyond the entailment literature, and points toward future specialized retrievers (Bowman et al., 2015; Williams et al., 2018; Conneau et al., 2017; MacCartney and Manning, 2009).

3 Dense Support Retrieval and Ranking

We propose a novel question answering architecture based on: (i) Dense Support Retrieval (DSR), which resembles DPR but specifically targets the *support relation* rather than general relevance between queries and texts. This means the retrieved text explicitly supports the correctness of a candidate answer with respect to the query. (ii) A Support-based Answer Reranker (SBAR), which leverages retrieved support to select the best answer candidate. Fig. 1 illustrates our architecture. Given a user's question q, a standard QA system initially provides the top-k candidate answers a_i (we use answer sentences). To identify the best answer, we first retrieve n supports s_{ij} for each candidate a_i by applying DSR to (q, a_i). Next, we employ a Supervised Support Reranker (SSR) to select the most accurate support s_i. Finally, SBAR reranks the triplets (q, a_i, s_i) to determine the optimal answer a^*. Although previous work has utilized support retrieval for answer selection, e.g., (Zhang et al., 2023), our complete architecture—incorporating DSR, SSR, and SBAR—is novel.

Dense Support Retrieval (DSR) We build a query vector $\vec{Q}$ from the pair (q, a), while each supporting sentence s is encoded into an embedding $\vec{s}$. In the introduction, we highlighted the difference between general relevance and the specific concept of support necessary to infer answer correctness. Defining explicit rules for this support relation is challenging; thus, we train transformer-based neural models to automatically capture it using supervised data. Specifically, given the top supporting sentences $s_1, \ldots, s_n$ retrieved for query Q, we train DSR using the following ranking loss:

$$L(Q, s_1, \ldots, s_n) = -\log \frac{e^{\text{sim}(Q,s_j)}}{\sum_{i=1, i\neq j}^{n} e^{\text{sim}(Q,s_i)}}, \quad (1)$$

where $Q = (q, a)$, $\text{sim}(Q, s_i) = \vec{Q} \cdot \vec{s}_i$, s_j is the positive support, and all other s_i represent negative supports.

Supervised Support Reranker (SSR) We train the SSR component in a fully supervised manner using two alternative loss functions: (i) a cross-entropy loss, yielding the model **SSR**$_{CE}$; and (ii) a ranking loss, yielding the model **SSR**$_{RL}$, as defined in Eq.1. For the latter, we reuse the training instances generated for DSR, substituting $\text{sim}(Q, s_i)$ and $\text{sim}(Q, s_j)$ with the supervised score $\pi(q, a, s_i)$ and $\pi(q, a, s_j)$ respectively, as described in Sec.2.

Support-based Answer Reranker (SBAR) SBAR is a triplet-based classifier designed to perform the AS2 task using retrieved supports. Starting from a TANDA model, we fine-tune SBAR on triplets composed of question, candidate answer, and the corresponding gold support. Additionally, we introduce the variant **SBAR**$^-$, which is further trained using negative supports retrieved by DSR.

4 Experiments

We first validate our approach against state-of-the-art methods for AS2 and support ranking. Then, we analyze the impact of our proposed components, DSR and SSR.

4.1 Datasets

We conducted primary experiments using two open-domain question answering datasets:

HotpotQA by Yang et al. (2018) is a widely used multi-hop QA benchmark comprising approximately 100,000 crowd-sourced questions. The correct answers to these questions can only be inferred

	AS2			Support Ranking		
	P@1	MAP	MRR	P@1	MAP	MRR
TANDA	66.0	75.8	76.9	-	-	-
ASR	68.0	77.2	78.1	-	-	-
DAR	68.4	77.5	78.5	-	-	-
DAR-DR	68.3	77.3	78.3	-	-	-
SBAR*	81.5	86.0	86.6	-	-	-
SSR_{CE}+SBAR	70.2	78.6	79.2	62.7	67.6	67.6

Table 2: HotpotQA distractor setting results.

by jointly reasoning across multiple Wikipedia paragraphs. The dataset provides gold-standard answer phrases along with associated supporting paragraphs, enabling explicit identification of answers. We adapted HotpotQA to the AS2 task following (Zhang et al., 2023), splitting paragraphs into sentences and labeling sentences as correct answers if they contain the gold-standard answer phrases. We evaluate our models using the official distractor dev-set as our test set.

FocusQA by Barlacchi et al. (2022) is specifically designed for AS2 and is enriched with context information. Each question-answer pair in FocusQA is accompanied by four context types: title question context (TQC), paragraph question context (PQC), title answer context (TAC), and paragraph answer context (PAC). These contexts improve disambiguation of both questions and answers, closely aligning with our definition of supporting evidence. Barlacchi et al. (2022) demonstrated that utilizing such contextual information substantially enhances baseline AS2 performance. Table 6 in the appendix summarizes dataset statistics.

4.2 Experimental Setup

Metrics: The primary metric for QA systems is Accuracy, equivalent to Precision at 1 (P@1) in ranking tasks, which measures the percentage of correctly identified answers. Additionally, we report Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) to facilitate direct comparison with existing benchmarks. Finally, we evaluate support retrieval effectiveness using Hit-Rate@k (H@k), defined as the percentage of queries for which at least one relevant support appears within the top-k retrieved items.

4.3 HotpotQA Results

We use HotpotQA to evaluate our models in the standard AS2 benchmark and analyze the effect of SSR on SBAR. Since the distractor setting already includes gold supports among the candidates, we do not test DSR here. Previous work (Zhang et al., 2023) showed that retrieving additional supports in this setup does not improve performance.

As shown in Tab. 2, ASR and DAR outperform TANDA by incorporating sentence-level supports. DAR-DR, which includes additional retrieved supports, does not improve accuracy, confirming that distractor passages already contain sufficient information. SBAR*, using gold supports, provides an upper bound.

Our SSR_{CE}+SBAR model improves over ASR/DAR by 2.0% absolute and successfully selects the correct support 62.7% of the time. This highlights the value of supervised support reranking over semi-supervised or implicit methods.

	AS2			Support Rank.		
	P@1	MAP	MRR	H@1	H@5	H@200
TANDA	28.7	40.7	44.7	-	-	-
Gold Standard Contexts						
TQC	35.9	44.2	51.3	-	-	-
PQC	41.9	48.0	55.5	-	-	-
TAC	32.1	43.0	47.2	-	-	-
PAC	17.2	27.8	31.4	-	-	-
Retrieved and Ranked PQC						
DPR+SSR_{CE}+SBAR	22.2	34.9	38.7	5.0	9.1	12.0
DSR+SSR_{CE}+SBAR	29.0	39.2	44.7	20.2	35.6	43.1
DSR^2+SBAR	28.6	39.1	43.8	22.2	37.7	71.6
DSR^2+SSR_{CE}+SBAR	30.5	41.1	46.2	25.8	41.1	71.6
DSR^2+SSR_{CE}+$SBAR^-$	31.9	42.2	47.2	25.8	41.1	71.6
DSR^2+SSR_{RL}+$SBAR^-$	33.1	42.7	48.0	29.7	46.0	71.6

Table 3: Ranking and retrieval results on FocusQA.

4.4 FocusQA Results

FocusQA provides a realistic evaluation setting where candidate answers are not artificially constrained to include supporting evidence. Table 3 reports our results. Without external support, TANDA reaches 28.7%.

Using gold contexts reveals strong variation: paragraph-question context (PQC) yields 41.9%, title-question context (TQC) gives 35.9%, while paragraph-answer context (PAC) lowers performance to 17.2%. This indicates that helpful support bridges the question and answer, rather than merely repeating answer-related content.

For automatic support retrieval, DPR+SBAR underperforms TANDA (22.2%), confirming that general relevance is not sufficient for answer verification. Our Dense Support Retrieval (DSR), trained specifically to retrieve (q, a)-supportive pas-

sages, improves to 29.0%, and DSR^2, retrained on its own outputs, slightly outperforms it at 29.5%. SBAR$^-$, trained with hard negatives from DSR, increases robustness and achieves 30.9%. Replacing cross-entropy with a ranking loss in SSR$_{RL}$ improves support selection. When combining DSR^2, SBAR$^-$, and SSR$_{RL}$, we achieve 33.1%, a 4.4% absolute improvement over TANDA.

Retrieval and reranking metrics further support these findings. DSR raises Hit@200 from 12% (DPR) to 43.1%, while DSR^2 reaches 71.6%. SSR$_{RL}$ surpasses SSR$_{CE}$, improving Hit@1 from 25.8% to 29.7% and Hit@5 from 41.1% to 46.0%. These gains demonstrate that support-aware retrieval and ranking yield substantial improvements for answer sentence selection in realistic, non-synthetic settings.

5 Retrieval Approach, Efficiency, and LLM Grounding

This section discusses three central aspects of our approach: (i) the need for explanatory support beyond simple concatenation, (ii) the computational complexity of Dense Support Retrieval, and (iii) the potential for grounding large language models (LLMs).

5.1 Explanatory Support vs. Concatenation

Using (q, a) as a concatenated query does not guarantee explanatory support. Our contribution is to explicitly train a dense retriever with supervised (q, a, s) triples, where s is a passage that justifies the correctness of a for q. This ensures the model learns an explanatory relation rather than general relevance. Empirically, this distinction is crucial: on FocusQA, DSR improves support recall at Hit@200 from 12.0% (DPR) to 71.6%, and increases AS2 accuracy by +4.4 absolute points.

5.2 Computational Complexity

A concern with elevating retrieval granularity from q to (q, a) is the potential increase in computational complexity. In fact, the per-query complexity of DSR is identical to DPR: both encode passages offline and issue one dense vector for approximate nearest neighbor (ANN) search. The only increase comes from verifying each candidate answer separately, which is standard in all answer verification systems. Thus, DSR introduces no new algorithmic complexity, but belongs to the existing class of answer verification pipelines.

5.3 Potential for LLM Grounding

Retrieved supports can also serve as grounding signals for large language models (LLMs). Current LLM retrievers are trained for general relevance, not explanatory grounding. Our work is the first to train a dense retriever with (q, a, s) supervision at scale (130M paragraphs), showing that it is possible to retrieve explanatory evidence rather than merely related content.

We envision a broader paradigm of relational retrievers specialized for different reasoning needs: support retrievers to justify correctness, causal retrievers to establish causal links, temporal retrievers for event ordering, and comparative retrievers for judgments. An LLM could call such retrievers as specialized APIs depending on the type of reasoning required. For example, to answer "Did climate change cause the decline in Arctic fox populations?", an LLM might query: (i) a causal retriever, (ii) a temporal retriever, and (iii) a support retriever.

We believe our paper lays the groundwork for this vision, opening new directions for fine-grained, relational grounding of LLMs via retrieval.

6 Conclusion

We proposed a new QA architecture that uses supporting text to select the final answer. We show that our approach retrieves text supporting the correctness of a for q, with higher accuracy than standard retrieval systems, even when they use the concatenation of q and a as a query. This indicates that our model captures the relation between q and a, not just their combined bag of words. This finding is important, as it suggests we can efficiently retrieve answer supports—or other textual relations between two texts—from hundreds of millions of paragraphs using our neural support retriever.

In future work, we will apply our approach to other textual relations, such as entailment or causality. These results open new directions for scaling fine-grained semantic reasoning in retrieval-based QA systems. We also see our approach as a foundation for relational retrievers that can serve as grounding tools for LLM-based reasoning, improving both factuality and interpretability.

Limitations

We proposed an effective support retrieval architecture for open-domain question answering.

The textual information that we target is passages/sentences. This kind of knowledge is the one researchers are testing to improve grounding of Large Language Models (LLM). However, we haven't performed any experiment to demonstrate that our retrieved supports can help knowledge grounding. It may be possible that supporting information does not produces better effect than just related information.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Gianni Barlacchi, Ivano Lauriola, Alessandro Moschitti, Marco Del Tredici, Xiaoyu Shen, Thuy Vu, Bill Byrne, and Adrià de Gispert. 2022. FocusQA: Open-domain question answering with a context in focus. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5195–5208, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Daniele Bonadiman and Alessandro Moschitti. 2020. A study on efficiency, accuracy and document structure for answer sentence selection. *CoRR*, abs/2003.02349.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. TANDA: transfer and adapt pre-trained transformer models for answer sentence selection. In *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, February 7-12, 2020*, pages 7780–7788. AAAI Press.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Bill MacCartney and Christopher D. Manning. 2009. *Natural Language Inference*. Ph.D. thesis, Stanford University. Ph.D. Thesis.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *SIGIR'15*.

Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1179–1189, Copenhagen, Denmark. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

E. Voorhees and D. Tice. 1999. *The TREC-8 Question Answering Track Evaluation*, pages 77–82. Department of Commerce, National Institute of Standards and Technology.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. 2019. A compare-aggregate model with latent clustering for answer selection. *CoRR*, abs/1905.12897.

Zeyu Zhang, Thuy Vu, Sunil Gandhi, Ankit Chadha, and Alessandro Moschitti. 2022. Wdrass: A web-scale dataset for document retrieval and answer sentence selection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4707–4711.

Zeyu Zhang, Thuy Vu, and Alessandro Moschitti. 2021. Joint models for answer verification in question answering systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3252–3262, Online. Association for Computational Linguistics.

Zeyu Zhang, Thuy Vu, and Alessandro Moschitti. 2023. Double retrieval and ranking for accurate question answering. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1751–1762, Dubrovnik, Croatia. Association for Computational Linguistics.

A Example Appendix

A.1 Dataset details

The number of question and question-answer pairs in each of the splits of each of the datasets is shown in Table 4.

A.2 Implementation details

DSR implementation and training In the DSR model implementation, we leveraged eight Tesla V100 GPUs (32GB memory each) and a batch size of 128 for training over 40 epochs, utilizing dual BERT-Base models as the encoder. The Adam optimizer, with a learning rate of 2e-65, was employed. For inference, a large index was constructed, encapsulating around 130 million passages drawn from 54 million Common-Crawl documents[2]. We retrieve top 200 passages for each query. The dataset was curated from English web documents of the 5,000 most frequented domains, including Wikipedia, from 2019 and 2020 Common Crawl releases, filtering out pages with insufficient length or improper HTML structures. We also add all of positive paragraph question contexts of FocusQA to the index.

SSR implementation and training For SSR model training, we used the Adam optimizer with a 5e-6 learning rate, two Tesla A100 GPUs (40GB memory each), and a batch size of 256. The SSR model requires only one transformer model, a RoBERTa-Base, with a maximum sequence length of 512 and trained over 10 epochs.

TC implementation and training For the training of our Triplet Classifier (TC), we employed the Adam optimizer with a learning rate of 5e-6, using eight Tesla V100 GPUs (32GB memory each) and a batch size of 512 across 15 epochs. The TC model utilizes a single transformer model, RoBERTa-Base, concatenating the triplet as `[CLS] Question [SEP] Answer [SEP] Support Context`, with a maximum sequence length of 512.

Further Discussion Please note that we trained our model using a publicly available datasets, FocusQA. The latter was built using open domain questions, a general retrieval system, and a general index built on common crawl (the most representative set of web data available to the research community). Therefore, there is no reason for considering our results specific. On specific domains,

[2] commoncrawl.org

Dataset	Train		Dev.		Test	
	Question	Q/A Pairs	Question	Q/A Pairs	Question	Q/A Pairs
HoptotQA	86,447	3,538,844	4,000	164,500	7,405	306,487
FocusQA	3,276	49,386	800	12,498	1,960	48,056

Table 4: Number of questions and labeled question-answer pairs in the train, development, and test splits of the two QA corpora.

e.g., medical or law texts, our retrieval will be subject to degradation of performance but this is rather standard for any retrieval system. That is, it should not be considered a specific weakness of our approach.

We train DSR on FocusQA, using the standard train, dev and test sets (well-described in the FocusQA paper). In our paper, we describe the procedures for converting that data in training data for support retrieval.

Our model requires training data to be designed, but this only because we train a model for a different retrieval paradigm for the first time, i.e., answer support retrieval. Also standard neural retrieval models required specific training data to enable their high accuracy for passage/document retrieval. Our training data is not more difficult to acquire than standard one, for example, one can use FocusQA and HotpotQA procedures.

The nature of our supporting relation is not different from relevance relation, which has been traditionally used to train retrieval systems. Thus, it can generalize through different datasets, similar to relevance retrieval: there is no reason to suspect that the adaptation challenges of current retrieval systems are less or more harsh than our support retrieval system.