LogiCoL: Logically-Informed Contrastive Learning for Set-based Dense Retrieval

Yanzhen Shen[1*], **Sihao Chen**[2*], **Xueqiang Xu**[3],
Yunyi Zhang[3], **Chaitanya Malaviya**[4], **Dan Roth**[4,5]
[1]Stanford University, [2]Microsoft, [3]University of Illinois Urbana-Champaign,
[4]University of Pennsylvania, [5]Oracle AI
yanzhen4@stanford.edu, danroth@seas.upenn.edu

Abstract

While significant progress has been made with dual- and bi-encoder dense retrievers, they often struggle on queries with logical connectives, a use case often overlooked yet important in downstream applications. In this paper, we introduce LogiCoL, a logically informed contrastive learning objective for dense retrievers. LogiCoL builds upon in-batch supervised contrastive learning and learns dense retrievers to respect the subset and mutually exclusive set relation between query results. We evaluated the effectiveness of LogiCoL in the entity retrieval task, where the model is expected to retrieve a set of Wikipedia entities that satisfy the implicit logical constraints of the query. We show that models trained with LogiCoL show improvements both in terms of retrieval performance and logical consistency in the results. We provide detailed analysis and insights to uncover why queries with logical connectives are challenging for dense retrievers and why LogiCoL is effective. Our codes and data are available at https://github.com/yanzhen4/LogiCoL.

1 Introduction

In recent years, dense retrievers have become a popular class of information retrieval (IR) methods, utilizing learned text embeddings to compute similarities between queries and documents (Karpukhin et al., 2020; Izacard et al., 2021). Dense retrievers have led to significant advances in knowledge-intensive natural language processing (NLP) applications (Chen et al., 2017; Kwiatkowski et al., 2019; Petroni et al., 2021).

However, despite the progress made, dense retrievers today still struggle in user queries with logical structure (Malaviya et al., 2023). In real-world applications, users often have complex retrieval needs that involve multiple preferences or constraints. As illustrated in Figure 1, user queries could contain different set operations, e.g. "*Species of orchids in Malaysia and / or / not in Thailand*". Although current dense retrievers are good at modeling the semantic relation between queries and documents, the logical relations within the query are often not well represented in the embedding space, leading to logically irrelevant or even contradictory retrieval results (Malaviya et al., 2023; Weller et al., 2024; Zhang et al., 2024).

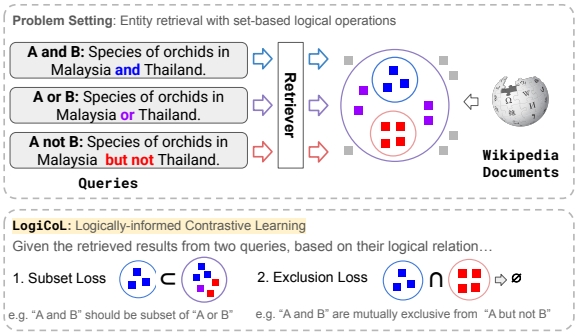

Figure 1: Overview of LogiCoL. We study retrieval tasks where the queries involve set operations. With queries that form subset or mutually exclusive relation with each other, we formulate the logical consistency between the result sets as loss terms, on top of the in-batch contrastive learning objective of dense retrievers.

To address this challenge, we propose LogiCoL, a **logi**cally-informed **co**ntrastive **l**earning framework that can be integrated with any dual- or bi-encoder dense retriever to enhance performance on queries with logical connectives. Our method is based on in-batch supervised contrastive learning (Khosla et al., 2020), where we sample queries that share atomic subqueries but with different logical connectives in the same mini-batch during training. For queries in which we expect the retrieval results to be subsets or mutually exclusive of each other, we train the model to respect the logical consistency between the result sets. The logical consistency constraints are expressed as two regularization terms based on t-norm (Li and Srikumar,

*Work was done while the first two authors were affiliated with the University of Pennsylvania.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 12114–12125
November 4-9, 2025

2019) in the learning objective, which model the subset and mutually exclusive relations between the retrieval results of queries.

We evaluate the effectiveness of LOGICOL on the QUEST dataset (Malaviya et al., 2023), where given a query, a model is expected to rank and retrieve the set of entities in Wikipedia that satisfy the requirements and constraints in the query. Our experimental results show that LOGICOL yields consistent improvement over off-the-shelf retrievers and supervised contrastive learning baselines in terms of retrieval performance and logical consistency of the retrieval results. By analyzing performance across different logical templates, we find that LOGICOL brings the most significant improvements on complex queries involving implicit intersection and negation operators. This is particularly important because prior embedding models already perform reasonably well on queries involving union, where relevance is loosely defined, but struggle with intersection and especially negation. Compared to contrastive learning without logical constraints, LOGICOL learns to separate queries that have different logical connectives but share a subset of ground truth documents, preventing logically distinct queries from collapsing to the same representation during model training.

To summarize, our contributions are three-fold.

- We propose LOGICOL, a logically-informed contrastive learning framework for retrieval with queries containing logical connectives.
- We conduct experiments on the task of set-based entity retrieval, where the model is expected to retrieve a set of entities in Wikipedia that satisfy the implicit logical constraints in the query.
- We provide detailed analysis and insights to uncover why queries with logical connectives are challenging for dense retrievers in general.

2 Motivation and Problem Setting

In this paper, we study retrieval tasks where the query contains logical connectives such as intersection (*AND*), union (*OR*), and negation (*NOT*). Specifically, we focus on the task of entity retrieval, where a model is expected to retrieve a set of entities in Wikipedia that satisfy the implicit logical constraints in the query (Malaviya et al., 2023).

With current dense retriever models, we observe that their query embeddings cannot distinguish and capture such logical relations well. To illustrate this, we visualize and compare the distributions

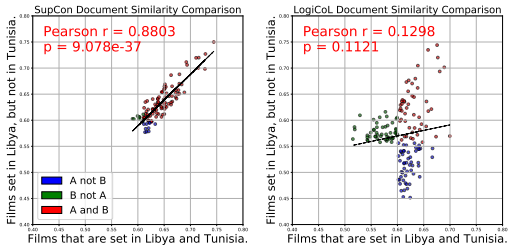

Figure 2: Document similarity correlation between queries of template *A AND B* vs. *A NOT B* of baseline (Left) and LOGICOL (Right). Each point corresponds to a candidate document, positioned by its similarity to the two queries, and it is also colored based on the different logical conditions it satisfied.

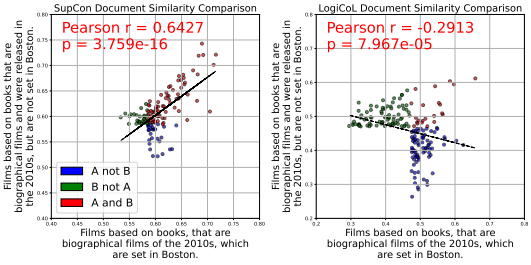

Figure 3: Document similarity correlation between queries of template *A AND B AND C* vs. *A AND B NOT C* of baseline (Left) and LOGICOL (Right).

of similarity scores between queries with different logical connectives and their top-100 retrieved documents. Figure 2 shows the similarity distribution for a query with template *A AND B* (e.g., *Films that are set in Libya and Tunisia*) versus *A NOT B* (e.g., *Films set in Libya, but not in Tunisia*). Similarly, in Figure 3, we compare queries of template *A AND B AND C* versus template *A AND B NOT C*. In both examples, the subqueries A, B, and C are identical across the compared query pairs; the only difference lies in the logical structure.

We observe that dense retrievers trained with the usual in-batch contrastive learning objectives cannot distinguish queries with different logical connectives. As shown in Figure 2 and 3, the distributions of query–document similarity scores for two queries with different logical connectives are highly correlated ($r = .88$ and $.64$, respectively). This indicates that the query embeddings for different queries are highly similar, despite the fact that queries results should be mutually-exclusive. Moreover, there is substantial overlap in the top-100 retrieved documents, even though the expected results are disjoint sets. This pattern holds across both examples, highlighting the inability of current embedding models to encode logical distinctions. In comparison, models trained with LOGICOL (§3)

show cleaner separation in the query embedding representations.

3 LOGICOL

In this section, we first describe the training data collection process (§3.1), and then the components of the LOGICOL learning objective (§3.2, §3.3).

3.1 Sampling Logically Related Queries

Our LOGICOL approach builds on top of in-batch supervised contrastive learning (Khosla et al., 2020), and models the logical consistency between the retrieval results from queries with subset or exclusion relations. Compared to the common practice of randomly sampling queries into mini-batches for contrastive learning (Karpukhin et al., 2020; Ni et al., 2021), LOGICOL requires each mini-batch to contain queries with shared atomic sub-queries but different logical connectives. For example, if a mini-batch contains the query "A and B", then we would want "A or B" to be in the same mini-batch as well.

We start from the QUEST dataset, which features queries with annotations of logical connectives and atomic sub-queries. However, only a small portion ($\sim$4%) of the queries in the dataset actually share atomic sub-queries with others. For this reason, we create an augmented version of QUEST, where variants of queries with different logical connectives are included. We denote this augmented dataset as QUEST+VARAIANTS. The types of logical connectives used in QUEST and QUEST+VARAIANTS are shown in Table 1. The ground truth document set for queries in QUEST+VARAIANTS is inferred from the set of ground truth entities for the atomic sub-queries. More details on the construction of QUEST+VARAIANTS can be found in Appendix A.1.

3.2 Contrastive Learning

During LOGICOL training, each mini-batch is constructed by sampling queries sharing the same set of atomic sub-queries. For each query, we randomly sample one ground truth document as part of the mini-batch. As different queries could share some ground truth documents, each query could have more than one ground truth document in the mini-batch, which the commonly adopted InfoNCE loss (Oord et al., 2018) does not support. For such reason, we leverage and modify the supervised contrastive loss (Khosla et al., 2020) as our base learning objective for LOGICOL.

$$\mathcal{L} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(q_i \cdot d_p / \tau)}{\sum_{j \in I \setminus \{i\}} \exp(q_i \cdot d_j / \tau)}$$

Here, $P(i)$ denotes the set of indices of all positive documents to the i-th query within the mini-batch, and $|P(i)|$ is its cardinality. $\mathcal{I} = \{1, \ldots, N\}$ denote the set of all queries and document indices. $\mathbf{q}_i \in \mathbb{R}^d$ denote the encoded representation of the i-th query in the mini-batch, and let $\mathbf{d}_i \in \mathbb{R}^d$ denote the encoded representation of the i-th document in the mini-batch.

3.3 Joint Constrained Learning

On top of the base contrastive learning objective, LOGICOL uses two regularization objectives to encourage the model to retrieve results that are consistent with the logical relations between queries. We specify two types of consistency requirements: exclusion relation consistency and subset relation consistency.

3.3.1 Exclusion Consistency

Given any two queries q_1 and q_2 within a mini-batch, if they exhibit an exclusion relation, then their retrieval results should be disjoint. Intuitively, the query such as *Orchids of Malaysia but not Thailand* and the query such as *Orchids of Thailand* should not retrieve overlapping documents.

To operationalize this constraint during training, we model the retrieval behavior of queries through their similarity distributions over documents, and require that any two queries with an exclusion relation should have divergent similarity distributions. Let q_i and q_j be two queries with an exclusion relation, and let $\mathbf{q}_i$, $\mathbf{q}_j$, and $\mathbf{d}_k$ denote their encoded representations and that of a document d_k in the same training mini-batch.

We compute the cosine similarity scores s_{q_i,d_k} and s_{q_j,d_k} between each query and document d_k and normalize them using a softmax to obtain probability distributions $\mathbf{s}_{q_i}$ and $\mathbf{s}_{q_j}$ over the mini-batch of documents:

$$\mathbf{s}_q = \text{softmax}([s_{q,d_1}, s_{q,d_2}, \ldots, s_{q,d_N}]).$$

To enforce divergence between the distributions of exclusion-related queries, we compute the symmetric Kullback–Leibler (KL) divergence between the two distributions:

$$\text{SymKL}(\mathbf{s}_{q_i}, \mathbf{s}_{q_j}) = \frac{1}{2}\left(\text{KL}(\mathbf{s}_{q_i} \,\|\, \mathbf{s}_{q_j}) + \text{KL}(\mathbf{s}_{q_j} \,\|\, \mathbf{s}_{q_i})\right) \quad (1)$$

where SymKL$(\cdot, \cdot)$ denotes the symmetric KL divergence, and KL$(p \,\|\, q)$ is the standard (asymmetric) KL divergence from distribution p to q.

We then train the model using the following margin-based loss function:

$$\mathcal{L}_E(q_i, q_j) = \max\left(\gamma_e - \text{SymKL}(\mathbf{s}_{q_i}, \mathbf{s}_{q_j}), 0\right),$$

where $\gamma_e > 0$ is a margin hyperparameter. The final exclusion loss averages over all exclusion-related query pairs within a mini-batch.

By maximizing the symmetric KL divergence between exclusion-related queries, we encourage the model to produce non-overlapping retrieval distributions, thereby enforcing exclusion consistency during training.

3.3.2 Subset Consistency

We define a subset relation between two queries (q_1, q_2) based on their logical structure. Specifically, $q_1 \subset q_2$ if the logical expression of q_1 implies that of q_2. For example, if $q_1 = A$ *AND* B and $q_2 = A$, then any document relevant to q_1 must also be relevant to q_2, since satisfying both A and B necessarily satisfies A. The subset relation between two queries can be interpreted as a logical implication: for any document d in the mini-batch, if d is relevant to q_1, then it should also be relevant to q_2. This implies that a high similarity score between q_1 and d must entail a correspondingly high similarity between q_2 and d. On the other hand, a case where $\text{sim}(q_1, d)$ is high but $\text{sim}(q_2, d)$ is low indicates a violation of this logical constraint.

To rewrite this constraint in a differentiable loss function, we use the product t-norm and transformation to the negative log space as in previous works (Li et al., 2019; Li and Srikumar, 2019), we define the subset loss:

$$\mathcal{L}_S = \sum_{(q_1,q_2)\in\mathcal{E}_S} \sum_{d\in\mathcal{D}} \max\Big(\log \text{sim}(q_1, d) - \log \text{sim}(q_2, d) + \gamma_s,\ 0\Big), \tag{2}$$

where γ_s is a margin hyperparameter.

Through this subset loss, we enforce the subset logical consistency, that is, for any d, it imposes a penalty when $\text{sim}(q_1, d)$ is high but $\text{sim}(q_2, d)$ remains comparatively low, thereby violating the implication that relevance to q_1 should entail relevance to q_2.

Templates	QUEST			QUEST+VARAIANTS		
	Train	Validation	Test	Train	Validation	Test
A	3482	56	260	3992	558	2286
$A \cap B$	168	54	271	430	278	1497
$A \cup B$	78	44	260	498	458	2532
$A \setminus B$	70	44	212	914	778	4226
$A \cap B \cap C$	190	40	236	114	52	294
$A \cap B \setminus C$	68	44	236	96	58	353
$A \cup B \cup C$	38	41	252	118	113	666
Total	4094	323	1727	6162	2295	11854

Table 1: Types of queries and dataset statistics for QUEST and QUEST+VARAIANTS.

3.3.3 Joint Learning Objective

After expressing the logical consistency requirements through additional margin-based loss terms, we combine all objectives into the following joint learning objective:

$$\mathcal{L}_{\text{joint}} = \mathcal{L} + \lambda_E \mathcal{L}_E + \lambda_S \mathcal{L}_S, \tag{3}$$

where $\mathcal{L}$ denotes the supervised contrastive loss, $\mathcal{L}_E$ denotes the exclusion consistency loss, and $\mathcal{L}_S$ denotes the subset consistency loss. The λ coefficients are non-negative hyperparameters to control the relative influence of each loss term.

3.4 Mixture of Random and Related Query Batching

To better understand the impact of changing the batch strategy compared to random batching for LOGICOL, we introduce a mixed batching strategy, where a mini-batch is comprised of both randomly sampled queries and groups of related queries. We use a hyperparameter α to control the proportion of random samples in the mixture. An analysis of the effect of α is discussed in §5.2.

4 Experiments

4.1 Dataset

We evaluate LOGICOL using the QUEST benchmark dataset (Malaviya et al., 2023). QUEST consists of 4,094 training queries and 1,727 testing queries spanning the domains of films, books, plants, and animals. The augmented QUEST+VARAIANTS contains 6,162 training queries and 11,854 testing queries. A detailed breakdown of the original and augmented Quest datasets across different templates is in Table 1.

4.2 Model Configurations

We initialize the transformer encoder layers with pre-trained weights from four types of sentence encoders: GTR (Ni et al., 2021), Contriever (Izacard et al., 2021), GTE (Li et al., 2023), and E5 (Wang

Backbone	Method	QUEST					QUEST+VARAIANTS				
		P@1	R@5	R@20	R@100	R@1000	P@1	R@5	R@20	R@100	R@1000
BM25		11.64	4.73	10.41	19.68	39.49	6.62	1.92	4.40	9.32	22.51
GTR-base	Zero-shot	8.98	3.41	7.07	13.46	30.95	5.63	1.38	3.13	6.66	18.82
	SupCon	19.75	8.08	16.66	31.71	60.34	12.38	3.45	7.45	15.63	38.04
	LOGICOL	**20.21**	**8.36**	**17.23**	**33.08**	**61.86**	**13.84**	**3.62**	**8.07**	**17.22**	**41.93**
GTE-base	Zero-shot	19.40	7.79	16.18	31.81	61.87	11.28	3.18	7.11	15.75	38.97
	SupCon	**25.77**	**11.16**	22.50	39.58	69.66	14.61	4.18	9.61	19.60	44.82
	LOGICOL	25.65	10.98	**22.60**	**40.69**	**71.24**	**15.78**	**4.50**	**10.14**	**21.46**	**48.87**
Contriever	Zero-shot	7.70	3.01	6.37	13.64	32.78	5.61	1.39	3.18	7.33	20.96
	SupCon	**21.31**	9.59	20.41	38.00	69.30	13.38	3.71	8.40	18.23	43.94
	LOGICOL	20.56	**9.76**	**20.45**	**39.38**	**71.46**	**13.84**	**4.03**	**9.50**	**20.71**	**49.16**
E5-base-v2	Zero-shot	18.18	7.75	16.29	30.97	60.19	11.45	3.09	6.90	14.99	37.70
	SupCon	**26.29**	11.16	22.47	40.04	70.32	15.26	4.48	9.58	19.82	45.49
	LOGICOL	24.32	**11.70**	**23.48**	**42.13**	**73.52**	**16.27**	**4.67**	**10.38**	**21.87**	**50.24**

Table 2: Performance comparison on QUEST and QUEST+VARAIANTS datasets across different backbone models and methods. We report *Precision@1*(P@1) and *Recall@k* (R@5/20/100/1000). Best scores per backbone model are bolded. LOGICOL achieves the best overall performance across all backbone models on both datasets.

et al., 2022). For each encoder, we use the base version of the model. In the case of E5, we specifically adopt its latest second version, E5-base-v2, to ensure stronger baseline performance.

To demonstrate the effectiveness of our framework, we use the original QUEST training queries and documents to fine-tune the sentence encoders using contrastive loss with random batching, which we denote as *SupCon*. We also include BM25 (Robertson et al., 2009) for reference.

4.2.1 Evaluation Metrics

Given an encoder's ranking of over 300k documents in the corpus for a given test query, we evaluate the results by Precision@1, denoted as P@1, and Recall@{5, 20, 100, 1000}, denoted as R@n, against the ground truth set of relevant documents.

4.3 Implementation Details

We train our models on four NVIDIA TITAN RTX GPUs for 10 epochs with an average effective batch size of 16. We adopt a constant learning rate for all four backbone models. For GTE (Li et al., 2023), Contriever (Izacard et al., 2021), and E5 (Wang et al., 2022), we used the recommended learning rate used in the original paper, which are 2e-5, 1e-5, and 1e-5. For GTR (Ni et al., 2021), we set the learning rate to be 1e-5, which significantly outperforms the original recommended learning rate 1e-3 using both *SupCon* and LOGICOL frameworks. In our joint constraint learning, we tune its hyperparameters based on performance on the development set. $\lambda_E = 0.1$ and $\lambda_S = 0.1$, and we set $\gamma_e = 0.2$ and $\gamma_e = 0.2$.

4.4 Experiment Results

Table 2 presents the evaluation results on both QUEST and QUEST+VARAIANTS. We observe that applying the LOGICOL fine-tuning framework consistently enhances performance across different backbone encoders, outperforming both the zero-shot baselines and *SupCon*. Notably, LOGICOL yields the most significant gains on Recall@1000, improving by approximately 2% to 3% over the fine-tuned baselines on the original test set, and by around 5% on the augmented test set. Given that the test sets contain a substantial number of queries (1,727 and 11,854, respectively), each associated with hundreds of ground-truth documents, these improvements provide strong evidence for the effectiveness of our framework.

We categorize queries into four types based on their implicit logical operators: None (i.e., single queries), Intersection, Negation, and Union. We then compare the performance of LOGICOL against the baseline across these categories. As shown in Table 3, LOGICOL consistently outperforms the baselines on atomic queries as well as complex queries involving intersection and negation operators. Although there is a slight drop in union queries, this can be attributed to the fact that the dense embeddings learned from the baseline contrastive learning schema treat all logical operators as the union operator, which leads

to marginally higher performance on the union queries and much lower performance on the intersection and negation queries in comparison to models trained using LOGICOL.

4.5 Ablation Study

The key technical novelty of LOGICOL is twofold: (1) we use groups of related queries as the smallest unit in our contrastive learning batch, enabling the model to learn the logical connectives within queries and logical relations between queries. (2) We apply joint constrained learning to enforce logical consistency among in-batch queries. We demonstrate the contributions of these techniques through a comprehensive ablation study. Specifically, we examine the following ablated variants:

- **SupCon**: Trained on the original QUEST data using random batching.
- **LOGICOL – GroupStrategy – Constraints**: Trained on QUEST+VARAIANTS training data using random batching.
- **LOGICOL – MixStrategy – Constraints**: Trained on QUEST+VARAIANTS using all grouped batches.
- **LOGICOL – Constraints**: Trained on QUEST+VARAIANTS using the mix strategy, but without applying joint constraint learning.

We compare these four ablated versions against LOGICOL. All variants are evaluated using the two strongest backbone models, Contriever and E5-base-v2, and we report performance in terms of Recall@100 (R@100) and Recall@1000 (R@1000). Our results in Table 4 reveal the following key observations: (1) **LOGICOL – GroupStrategy – Constraints** provides minimal improvement over **SupCon**, suggesting that simply increasing the number of training queries offers little benefit. (2) **LOGICOL – Constraints** improves performance over **LOGICOL – MixStrategy – Constraints** when using the E5-base-v2 backbone, but underperforms it when experimenting with Contriever. This suggests that solely using batches of queries that share atomic sub-queries has the risk of learning insufficient semantic information. (3) **LOGICOL - Constraints** consistently and significantly outperforms both **LOGICOL – GroupStrategy – Constraints** and **SupCon**, underscoring the advantage of our mixed group-based batching strategy that leverages both logical and semantic similarities. (4) Finally, LOGICOL further improves upon **LOGICOL – Constraints**, demonstrating that enforcing logical consistency through joint constrained learning provides consistent gains during training.

Backbone	Method	Performance across Query Templates (%)				
		None	Intersection	Negation	Union	All
GTR-base	SupCon	22.10	18.02	14.71	10.68	15.63
	LOGICOL	**23.61**	**19.34**	**17.10**	**11.35**	**17.22**
GTE-base	SupCon	26.97	23.24	18.04	**13.99**	19.60
	LOGICOL	**29.30**	**27.37**	**21.46**	12.45	**21.46**
Contriever	SupCon	26.06	21.94	15.71	**13.54**	18.23
	LOGICOL	**27.78**	**26.62**	**20.43**	12.23	**20.71**
E5-base-v2	SupCon	26.91	22.85	18.92	**13.93**	19.82
	LOGICOL	**29.01**	**27.20**	**21.48**	13.86	**21.87**

Table 3: Recall@100 across different backbone models on QUEST+VARAIANTS queries with different logical connectives.

Method	Contriever		E5-base-v2	
	R@100	R@1000	R@100	R@1000
SupCon	38.00	69.30	40.04	70.32
– GroupStrategy – Constraints	38.34	70.41	39.87	69.94
– MixStrategy – Constraints	38.05	70.33	40.46	72.14
– Constraints	39.21	71.39	41.73	73.26
LOGICOL	**39.38**	**71.46**	**42.13**	**73.52**

Table 4: Ablation versions of LOGICOL on QUEST queries. Rows below show versions with specific components removed.

4.6 Document Similarity Distribution

To further validate the effectiveness of our LOGICOL framework, we repeat the same analysis from Section §2, comparing document similarity distributions for logically mutually-exclusive queries under the same retrieval setting described in Section 3. In Figures 2 and 3, it shows that LOGICOL significantly reduces the similarity correlation between such queries in comparison to the baseline's result. The Pearson coefficients drop to 0.1298 and even -0.2913, indicating that the model no longer treats logically mutually-exclusive queries as semantically equivalent. The retrieved documents are also more cleanly separated, with minimal overlap in the top-100 results. These findings demonstrate that LOGICOL effectively encodes logical structure in the embedding space, enabling more faithful retrieval aligned with query intent.

5 Analysis and Discussion

5.1 Evaluating Logical Consistency in Retrieved Results

In this analysis, we aim to evaluate whether for queries with negation operators ($A \setminus B$ and $A \cap B \setminus C$) models trained using LOGICOL can better avoid retrieving logically excluded documents (B and C respectively). To measure this, we compute the average ranking positions of ground-truth documents

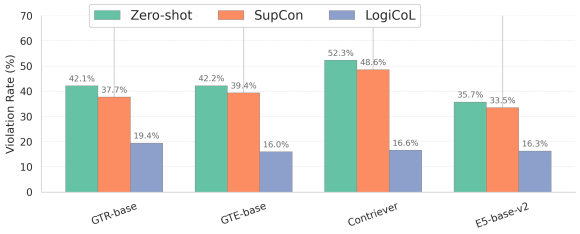

Figure 4: Logical consistency violation rate on queries with negation in comparison with *Zero-shot* and *SupCon* baselines on QUEST+VARAIANTS. Higher value indicates higher violation rate.

in the retrieval results AvgRank_q^+ and the average ranking positions of the logically excluded documents AvgRank_q^-. Then, if the logically excluded documents are ranked higher than its ground truth documents, which is $\text{AvgRank}_q^- < \text{AvgRank}_q^+$, then it is said that the retrieval result of this query has violated the logical consistency.

We define the overall violation rate across the test set as:

$$\text{ViolationRate} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbb{I}\left[\text{AvgRank}_q^- < \text{AvgRank}_q^+\right] \tag{4}$$

We show our result in Figure 4. *Zero-shot* and *SupCon* both have high violation rates, implying that current models fail to identify the exclusion relations, and this skill cannot be learned through simple supervised contrastive learning. On the other hand, training with LOGICOL framework significantly decreases the violation rate by over 20% for all backbone models, indicating they have successfully learned this skill.

5.2 Effect of Grouping Randomness on Embedding Coherence and Performance

To understand how including relevant queries in the same mini-batch affects the performance of the in-batch contrastive learning process, we study the effect of the hyperparameter α, which controls the proportion of random samples versus groups of related queries. We analyze its effect on both model performance and the coherence of embeddings. To measure the coherence of embeddings, within each group, we compute the pairwise similarity between queries that share the same atomic sub-queries base but differ in logical connectives. Formally, for a group $G = \{q_1, q_2, \ldots, q_n\}$ of queries sharing the same atomic sub-queries, we define the average embedding similarity as:

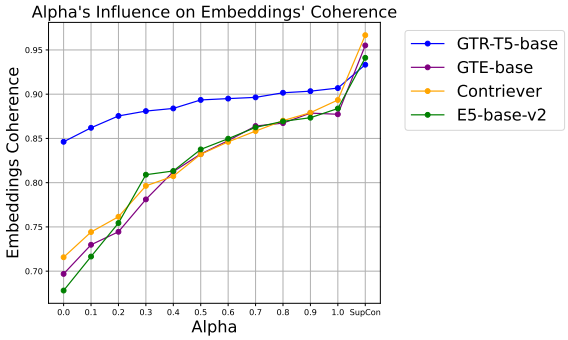

Figure 5: Coherence of embeddings (y-axis) of LOGICOL with different α values (x-axis). Larger α indicates lower randomness in batching strategy on QUEST+VARAIANTS. Coherence of *SupCon* baseline is put after the result of $\alpha = 1$ for reference (labeled as *SupCon*).

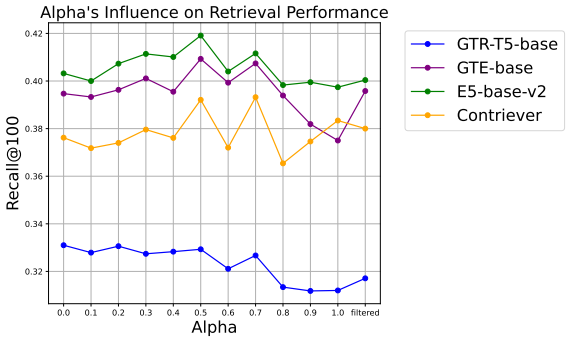

Figure 6: Retrieval Performance (Recall@100) of LOGICOL with different α values on QUEST+VARAIANTS. Performance of *SupCon* baseline is also put at the end.

$$\text{AvgGroupSim}(G) = \frac{2}{|G|(|G|-1)} \sum_{i<j} \text{sim}(f(q_i), f(q_j)) \tag{5}$$

where $f(q)$ denotes the embedding of query q, and $\text{sim}(\cdot,\cdot)$ is the cosine similarity between embeddings.

As shown in Figure 4, the performance experiences a slight drop when α increases from 0.8 to 1.0. This phase corresponds to augmenting the training data while still relying on the random batching strategy. However, as α decreases from 0.7 to 0.5—indicating a greater use of related queries groups—the performance significantly improves, outperforming the baseline. Beyond this point, as α continues to decrease, we observe a mild decline in performance.

These results suggest that while sampling logically related query groups enhances the model's ability to distinguish queries with different logical

structures, relying solely on this sampling method may hurt its ability to generalize over semantic similarity.

5.3 Why LogiCoL is Effective

As shown in Figure 6 and Table 3, models trained with *SupCon* struggle to capture the logical structure of complex queries, leading to poor performance, especially on intersection, negation, and union templates. In contrast, LogiCoL strikes a better balance between learning semantic coherence and the logical relations within queries. Furthermore, as shown in Figure 4, LogiCoL maintains logical consistency in retrieval with negation, avoiding entities explicitly excluded by the query. Overall, LogiCoL provides an effective contrastive learning framework that improves set-based retrieval.

6 Related Work

6.1 Dense Retrieval

Dense Passage Retriever (DPR) (Karpukhin et al., 2020) pioneered the approach of dense retrieval by encoding queries and passages into dense vector representations. Subsequently, Generalizable T5 Retriever (GTR) (Ni et al., 2021) uses a large pre-trained language model (T5) to encode texts, thus capturing richer semantic contexts. Contriever (Izacard et al., 2021) and E5 (Li et al., 2023) introduces an unsupervised and weakly supervised framework that trains on a large corpus. General Text Embedding (GTE) (Wang et al., 2022) adapts additional data sources on both pre-training and fine-tuning. SimCSE (Gao et al., 2021) introduces a framework that leverages data augmentation techniques to generate effective sentence embeddings through contrastive learning without additional labeled data. However, these methods focus solely on surface-level semantic similarity and fail to account for the underlying logical connectives present in complex queries.

6.2 Retrieval on Logical Queries

To examine the performance of these dense retrievers on logical queries, (Malaviya et al., 2023) creates the QUEST dataset, which consists of 3357 queries that map to a set of entities corresponding to Wikipedia documents. The BoolQuestions Zhang et al. (2024) dataset also contains logical queries and is constructed upon MSMarco (Nguyen et al., 2016). There have also been attempts to improve retrieval performance on queries with implicit logical operators. Mai et al. (2024) uses GPT-3.5 to generate over 150,000 additional training samples to fine-tune a BERT model (Devlin et al., 2019) with a customized contrastive loss. Krasakis et al. (2025) interpretable representations through Learned Sparse Retrieval (LSR). In contrast, our approach directly encodes logical structure through contrastive learning with logic-based groups and constraint-based objectives, without relying on expensive synthetic data or lexical heuristics.

6.3 Contrastive learning

Contrastive learning's efficacy relies heavily on the selection of negative samples, and several approaches Robinson et al. (2020); Zhang et al. (2025); Xiong et al. (2020) select and update hard negatives based on document embeddings distance or BM25 (Robertson et al., 2009) during training. To avoid the huge costs in continuously selecting hard negatives during training time, (Hofstätter et al., 2021; Sachidananda et al., 2023; Ma et al., 2024; Solatorio, 2024; Morris and Rush, 2024) have explored optimizing batch strategies beforehand. Most recently, Morris and Rush (2024) uses K-Means clustering on text embeddings to find the hardest possible batch to train the model. In contrast, the sampling strategy of our work is based on the logical structure of the queries instead of semantic similarity, and our mix strategy also enables the model to learn both semantics and logical distinctions.

Roth and Yih (2004, 2007) introduce the early frameworks for joint constrained learning by formulating entity and relation identification as a global inference problem. This paradigm has also been applied to the training of neural networks by (Li et al., 2019). This paradigm is further applied to information extraction tasks, such as temporal relation extraction Wang et al. (2020) and Emotion-Cause Pair Extraction Feng et al. (2023). We extend the usability of this framework from information extraction tasks to the text retrieval task by utilizing the subset and the exclusion relations between in-batch queries.

7 Conclusions

We present LogiCoL, a logically-informed contrastive learning framework designed to enhance the performance of dense retrievers on queries with logical connectives. LogiCoL introduces a novel

group batching strategy that enables the model to learn distinguishable representations for queries that are logically different yet semantically similar. In addition, we leverage the logical relationships among in-batch queries through a joint constrained learning objective, ensuring logical consistency in representation learning. Our experiments on the task of retrieving Wikipedia entities given complex logical queries demonstrate the effectiveness of LOGICOL over standard supervised contrastive learning. We further provide detailed analysis showing that LOGICOL significantly improves retrieval performance on queries involving negation. Finally, a hyperparameter study illustrates how tuning batching randomness influences the learning of semantic and logical information.

Limitations

Our work has the following limitations. First, due to limited computational resources, our experiments are conducted using moderate-sized bi-encoder models with approximately 110M parameters and an effective batch size of 32. Second, as existing benchmarks primarily focus on English, the cross-lingual generalizability of LOGICOL remains an open question.

Acknowledgments

We would like to thank Siru Ouyang and Yu Zhang for their valuable suggestions. This work was partially supported by the Intelligence Advanced Research Projects Activity (IARPA), via 2022-22072200003 and by ONR Contract N00014-19-1-2620. This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via 2022-22072200003. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, volume 1 (long and short papers)*, pages 4171–4186.

Huawen Feng, Junlong Liu, Junhao Zheng, Haibin Chen, Xichen Shang, and Qianli Ma. 2023. Joint constrained learning with boundary-adjusting for emotion-cause pair extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1118–1131.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Antonios Minas Krasakis, Andrew Yates, and Evangelos Kanoulas. 2025. Constructing set-compositional and negated representations for first-stage ranking. *arXiv preprint arXiv:2501.07679*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. 2019. A logic-driven framework for consistency of neural models. *arXiv preprint arXiv:1909.00126*.

Tao Li and Vivek Srikumar. 2019. Augmenting neural networks with first-order logic. *arXiv preprint arXiv:1906.06298*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Jiawei Ma, Po-Yao Huang, Saining Xie, Shang-Wen Li, Luke Zettlemoyer, Shih-Fu Chang, Wen-Tau Yih, and Hu Xu. 2024. Mode: Clip data experts via clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26354–26363.

Quan Mai, Susan Gauch, and Douglas Adams. 2024. Setbert: Enhancing retrieval performance for boolean logic and set operation queries. In *Proceedings of the 2024 8th International Conference on Natural Language Processing and Information Retrieval*, pages 162–167.

Chaitanya Malaviya, Peter Shaw, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2023. Quest: A retrieval dataset of entity-seeking queries with implicit set operations. *arXiv preprint arXiv:2305.11694*.

John X Morris and Alexander M Rush. 2024. Contextual document embeddings. *arXiv preprint arXiv:2410.02525*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the eighth conference on computational natural language learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8.

Dan Roth and Wen-tau Yih. 2007. Global inference for entity and relation identification via a linear programming formulation. *Introduction to statistical relational learning*, pages 553–580.

Vin Sachidananda, Ziyi Yang, and Chenguang Zhu. 2023. Global selection of contrastive batches via optimization on sample permutations. In *International Conference on Machine Learning*, pages 29542–29562. PMLR.

Aivin V Solatorio. 2024. Gistembed: Guided in-sample selection of training negatives for text embedding fine-tuning. *arXiv preprint arXiv:2402.16829*.

The Apache Software Foundation. 2025. Apache lucene: A high-performance, full-text search library. Latest release version 10.2.2, released June 20, 2025. Originally created by Doug Cutting in 1999.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. Joint constrained learning for event-event relation extraction. *arXiv preprint arXiv:2010.06727*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Orion Weller, Dawn Lawrie, and Benjamin Van Durme. 2024. Nevir: Negation in neural information retrieval. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2274–2287.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Yu Zhang, Yanzhen Shen, SeongKu Kang, Xiusi Chen, Bowen Jin, and Jiawei Han. 2025. Chain-of-factors paper-reviewer matching. In *Proceedings of the ACM on Web Conference 2025*, pages 1901–1910.

Zongmeng Zhang, Jinhua Zhu, Wengang Zhou, Xiang Qi, Peng Zhang, and Houqiang Li. 2024. Boolquestions: Does dense retrieval understand boolean logic in language? *arXiv preprint arXiv:2411.12235*.

A Appendix

A.1 Details On QUEST+VARAIANTS

Since not all the companion queries exist in the dataset, we develop the following automated pipeline to augment the original training set.

To mitigate this issue, we derived the missing base atomic queries from Wikipedia category names, which are hand-curated natural language labels assigned to groups of related documents in Wikipedia. We strictly follow the data construction procedure of Quest so that our extracted atomic queries match the original complex queries' decompositions.

After obtaining atomic queries, we then proceed to construct related complex queries from them. We limit our related complex queries to one of the 6 templates used in the original Quest dataset, as shown in Table 1. Then, for each new complex query $A \circ_1 B \circ_1 C$, we derived its ground truth document set from the ground truth document sets of its atomic queries A, B, and C, by applying the same operations on its answer document sets.

When combining these sub-queries and selected operators into a natural language, we first combine sub-queries through fixed templates, such as *{} and {}, but not {}*. Then, to ensure that the newly constructed queries closely resemble real-world queries, we also prompted an LLM, to paraphrase the templatically generated query to fluent natural language.

Besides augmenting the training dataset, we also augment the validation and testing datasets to provide a more comprehensive evaluation following the same rules.

To organize training batches, we cluster queries that share the same set of sub-queries $\{A, B, C\}$. Specifically, we identify two types of queries to include in the cluster. The first type consists of other queries that involve the same sub-query components $\{A, B, C\}$ but are connected with different logical operators; this encourages the model to differentiate between queries that are semantically similar but logically distinct. To avoid overpopulating our batch, we only add new complex queries that belong to one of Quest's original templates, which are $A \cap B \cap C$, $A \cap B \setminus C$, and $A \cup B \cup C$. The second type consists of the individual sub-queries A, B, and C; this promotes the model's ability to understand the relationship between a complex query and its components, recognizing how sub-queries and logical operators combine to form the overall query meaning.

Backbone	Method	QUEST			
		R@5	R@20	R@100	R@1000
GTR-base	SupCon	**8.08**	**16.66**	**31.71**	**60.34**
	SupCon + DTE	7.52	14.35	24.56	44.74
GTE-base	SupCon	**11.16**	**22.5**	**39.58**	**69.66**
	SupCon + DTE	9.45	18.68	30.83	52.02
Contriever	SupCon	**9.59**	**20.41**	**38.00**	**69.30**
	SupCon + DTE	8.18	17.15	30.15	53.30
E5-base-v2	SupCon	**11.16**	**22.47**	**40.04**	**70.32**
	SupCon + DTE	9.48	18.89	31.40	52.26

Table 5: Performance comparison between *SupCon* and *SupCon + DTE* on QUEST

For example, given the query in the previous example, *Orchids of Indonesia*, we include

- Other queries involving the same sub-queries but different logical structures
  - *Orchids of Indonesia and Malaysia and Thailand* ($A \cap B \cap C$)
  - *Orchids of Indonesia or Malaysia or Thailand* ($A \cup B \cup C$)
- The individual sub-queries themselves:
  - *Orchids of Indonesia* (A)
  - *Orchids of Malaysia* (B)
  - *Orchids of Thailand* (C)

A.2 Decompose-then-Ensemble Baseline

Traditional search systems like Apache Lucene (The Apache Software Foundation, 2025) handle complex queries with implicit set operators through a Decompose-then-Ensemble (DTE) strategy: complex queries are first decomposed into individual sub-queries, similarity scores are computed between each sub-query and all documents, then ensembled based on set operators to produce a combined similarity score for ranking. We compare this approach against naive dense retrieval, which encodes complex queries holistically.

For a complex query Q with sub-queries $Q_1, Q_2, \ldots, Q_n$ and operators $op_1, op_2, \ldots, op_{n-1}$, we compute similarity scores $s_i = \text{sim}(Q_i, D)$ for each document D and combine them as follows:

Binary operations:

$$\text{AND:} \quad \min(s_1, s_2) \tag{6}$$

$$\text{OR:} \quad \max(s_1, s_2) \tag{7}$$

$$\text{NOT:} \quad s_1 - s_2 \tag{8}$$

Ternary operations:

$$\text{AND-AND:} \quad \min(s_1, s_2, s_3) \tag{9}$$

$$\text{OR-OR:} \quad \max(s_1, s_2, s_3) \tag{10}$$

$$\text{AND-NOT:} \quad \min(s_1, s_2) - s_3 \tag{11}$$

We experiment with the same four types of sentence encoders and evaluate on QUEST using Recall @ {5, 20, 100, 100}. Our experimental results reveal that DTE consistently degrades performance across all backbone models and recall metrics. As shown in Table 5, SupCon substantially outperforms SupCon + DTE, with relative performance drops of 15-35% across configurations. These results demonstrate that holistic dense encoding could better preserve semantic relationships within queries.