Link, Synthesize, Retrieve: Universal Document Linking for Zero-Shot Information Retrieval

Dae Yon Hwang[1,2*] **Bilal Taha**[2,3] **Harshit Pande**[1*] **Yaroslav Nechaev**[1*]
[1] Amazon AGI [2] University of Toronto [3] Vector Institute
daeyon.hwang@alumni.utoronto.ca bilal.taha@mail.utoronto.ca
pandeconscious@gmail.com {dhwang, hppnd, nechaey}@amazon.com

Abstract

Despite the recent advancements in information retrieval (IR), zero-shot IR remains a significant challenge, especially when dealing with new domains, languages, and newly-released use cases that lack historical query traffic from existing users. For such cases, it is common to use query augmentations followed by fine-tuning pre-trained models on the document data paired with synthetic queries. In this work, we propose a novel Universal Document Linking (UDL) algorithm, which links similar documents to enhance synthetic query generation across multiple datasets with different characteristics. UDL leverages entropy for the choice of similarity models and named entity recognition (NER) for the link decision of documents using similarity scores. Our empirical studies demonstrate the effectiveness and universality of the UDL across diverse datasets and IR models, surpassing state-of-the-art methods in zero-shot cases. The developed code for reproducibility is included in the supplementary material. [1]

1 Introduction

In information retrieval (IR), zero-shot learning is an essential problem that emerges when dealing with a new language or domain with little to no availability of the associated queries. Traditional IR methods primarily utilized sparse retrieval, while recent methods revolve around dense retrieval (DR), demonstrating the promising result (Neelakantan et al., 2022). Yet, using pre-trained DR directly on zero-shot cases results in substantial performance degradation, requiring dedicated fine-tuning (Izacard et al., 2021; Zhang et al., 2021).

One strategy for fine-tuning without relying on query traffic involves expanding the queries based on existing queries or documents with rule-based methods or language models (LMs) to obtain additional context in unseen domains (Wang et al., 2023; Jagerman et al., 2023; Weller et al., 2024). RM3 (Abdul-Jaleel et al., 2004) and AxiomaticQE (Yang and Lin, 2019) are classical ways to expand the queries with additional relevant terms while the recent studies indicate that large LMs (LLMs) can produce sophisticated synthetic data (Schick and Schütze, 2021), often resulting in better transfer learning than human-curated datasets (Liu et al., 2022). While LLMs like Gemini (Team et al., 2023) generate superb synthetic queries for fine-tuning, devising a cost-effective way for IR remains challenging without additional recipes like dimensionality reduction (Hwang et al., 2023b).

To address the limitations of document-to-query generation, we propose a novel algorithm called Universal Document Linking (UDL), which offers an intuitive yet effective solution for zero-shot.

Table 1: Synthetic queries augmented by UDL.

Document	Augmented query before UDL	Augmented query by UDL
In case of allergic rhinitis, you are still in group of subjects who can receive AstraZeneca's Covid-19 vaccine.	Subject of astrazeneca vaccination	Covid-19 vaccination for allergic rhinitis
With allergic rhinitis, according to regulations of the Ministry of Health, you can still receive the Covid-19 vaccine normally.	Regulations of the Ministry of Health on allergic rhinitis	
Google Finance gives you free information.	Google finance cost	Which company gives the free quotes?
Sure, Yahoo Finance does this for FREE.	Is yahoo finance free?	
Most predict dire consequences if GHGs continue to rise through the 21st century, which is what seems most likely.	Does GHG increase?	What is the future of climate change?
There may be some tipping points that will accelerate climate change but we do not know when each of these will become a problem.	Acceleration of climate change	
Public health is a key issue– the state has a role in stopping people harming themselves – they may be harming themselves but the cost often falls on government through public healthcare, and therefore on all taxpayers. Smoking also harms others through passive smoking.	Why are we banning smoking?	Do governments have the right to ban smokers?
Paternalistic Personal autonomy has to be the key to this debate. If people want to smoke – and the owner of the public place has no issue with that – it is not the role of the state to step in. All that is required is ensuring that smokers are educated about the risks so that they can make an informed decision.	Why the education needs for smoking	

[*] Work was done outside of Amazon
[1] https://github.com/eoduself/UDL

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 18971–18982
November 12-16, 2024

This method links similar documents, aiding in the generation of synthetic queries spanning multiple documents. The UDL algorithm relies on selecting a similarity model based on term entropy and determining the link decisions using named entity recognition (NER) models. This approach facilitates the link decisions tailored to each dataset's unique characteristics, highlighting the universality of our method. Moreover, UDL is flexible to be combined with other query augmentations which reveals the high extensibility. With UDL, small LM can outperform LLM with a low cost. Table 1 presents examples demonstrating how UDL generates additional relevant queries that would not be generated by its absence. In this work, we make two main contributions: **(1)** Exploring the document linking for query augmentation with empirical studies which was not investigated previously, and **(2)** Introducing the UDL algorithm and demonstrating its effectiveness across diverse query augmentations, IR models, and datasets with varying tasks.

2 Motivation

Figure 1 illustrates the overall flow of fine-tuning a retrieval model in zero-shot scenario, where actual queries do not exist during fine-tuning. Instead, we use documents to generate synthetic queries, which aids the IR model in learning the distribution of the unseen domain (Thakur et al., 2021).

According to Hwang et al. (2023a) and our initial findings (Table 11), merely increasing the size of synthetic data doesn't consistently improve results. This is because query augmentation associates a synthetic query with a single document, whereas queries in datasets can be linked to multiple documents. Our insight from this led us to develop a method to link similar documents for the generation of synthetic queries that cover multiple documents.

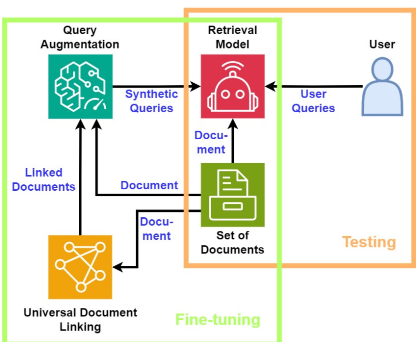

Figure 1: Overall zero-shot case. IR model is fine-tuned with synthetic queries, then interacted with user queries.

Algorithm 1 Universal Document Linking

Data: A set of documents in each dataset
Result: Linked documents
Parameters: Thresholds in similarity model γ and score δ, decision of similarity model D_M and score D_T, pre-trained general NER N_g and specialized NER N_s

Step A. Decision of Similarity Model

1. Measure TF-IDF in all documents
2. Calculate $Entropy$ for each term in TF-IDF across documents
3. **if** $D_M = \frac{\#\ of\ terms\ in\ Entropy > 1}{\#\ of\ terms\ in\ Entropy \leq 1} > \gamma$ **then**
   Use pre-trained LM as similarity model

else
   Use TF-IDF as similarity model
end

Step B. Decision of Similarity Score

1. **if** *candidate documents not in English* **then**
   Translate to English

end

2. Eliminate the special characters in candidates
3. $D_T = \begin{cases} \delta \text{ , if } K_{N_g} \times V_{N_s} > K_{N_s} \times V_{N_g} \\ 1 - \delta \text{ , otherwise} \end{cases}$

K: Number of keywords from NER
V: Vocabulary size of NER

Step C. Link Documents

1. Measure the cosine-similarity between candidate documents using a model from **A**
2. **if** *cosine-similarity > score from* ***B*** **then**
   Link documents

end

3 Universal Document Linking

Algorithm 1 outlines the procedural steps in the UDL. In the first step, denoted as **A**, the appropriate similarity model is selected for each dataset. We explore term frequency-inverse document frequency (TF-IDF) and pre-trained LM to derive document embeddings. Notably, TF-IDF considers lexical similarity, which is valuable for identifying unique features (e.g., disease like COVID), while pre-trained LM provides semantic similarity, aiding in contextual understanding. To determine the suitable similarity model, we initially compute TF-IDF scores for all documents, followed by calculating D_M based on the Shannon entropy of terms using TF-IDF. Entropy values greater than 1 (i.e., numerator in D_M) describe high uncertainty since random variables have approximately uniform distribution in multiple classes. This concept is extended to the term entropy (Equation (1)) where we calculate the entropy for each term across documents.

To accommodate the D_M for the massive documents, we introduce the γ value where articles and relatively common terms are mostly distributed in entropy greater 1 as expected (see Table 12). Documents with an overwhelming presence of these terms are not desirable for TF-IDF since it can obscure the unique characteristics of documents, affecting link decisions. In such cases, considering semantically similar documents using pre-trained LM proves to be a more viable alternative.

After defining the similarity model, we proceed to determine the criteria in step **B** for deciding whether candidate documents should be linked. Each dataset contains varying levels of domain-specific terminology, which must be taken into account during document linking. To address this, we initially translated non-English documents into English using Google Translator [2] to handle multilingual cases. After removing special characters, we compute D_T based on the number of keywords extracted from NER models that are pre-trained on general (N_g) and specialized documents (N_s) while considering the vocabulary size of each NER for unbiased comparison. Note that a large size of vocabulary can have a higher chance of capturing broad keywords. The entity coverage is detailed in Table 9, where N_g effectively identifies keywords in documents related to the natural conversation and question-answering (QA), while N_s adequately finds keywords from professional jargon like medical and scientific claims.

Based on this analysis, a higher value of D_T indicates that a dataset is more similar to a group of general documents, enabling the linking of diverse documents without concerns of domain-specific jargon, resulting in a lower score (i.e., δ). Conversely, a lower D_T value suggests that a dataset consists of specialized documents, which benefits from linking similar documents that share domain-specific jargon, resulting in higher scores (i.e., $1-\delta$). Thus, general and specialized documents are considered opposites. In Section 4, we tested the UDL across multiple datasets from different domains (e.g., QA, scientific documents) to show its applicability without requiring a specific NER for each domain. This was confirmed with the selected NERs but our UDL could be readily extendable to any other NER.

[2] https://github.com/ssut/py-googletrans

Table 2: Query augmentations with Distilled-BERT. Performances (SD) are from NFCorpus, SciFact, ArguAna.

Method	N@10	R@100	# Parameters
Off-the-shelf	40.7 (0.0)	67.5 (0.0)	-
Cropping (Izacard et al., 2021)	38.8 (0.4)	68.3 (0.5)	-
RM3 (Abdul-Jaleel et al., 2004)	41.7 (0.4)	70.2 (0.4)	-
AxiomaticQE (Yang and Lin, 2019)	43.4 (0.5)	69.7 (0.3)	-
Summarization (Zhang et al., 2020)	43.3 (0.6)	69.4 (0.2)	569M
Flan (Chung et al., 2024)	44.3 (0.3)	70.4 (0.3)	248M
OpenLLaMA (Geng and Liu, 2023)	47.0 (0.4)	72.5 (0.5)	3B
QGen (Raffel et al., 2020)	46.3 (0.5)	71.9 (0.4)	109M
UDL + RM3	44.0 (0.4)	71.6 (0.5)	109M
UDL + AxiomaticQE	44.5 (0.3)	71.4 (0.5)	109M
UDL + Summarization	45.1 (0.4)	71.7 (0.4)	678M
UDL + Flan	45.2 (0.6)	72.1 (0.5)	357M
UDL + OpenLLaMA	48.2 (0.2)	73.1 (0.3)	3.1B
UDL + QGen	**49.5 (0.3)**	**73.6 (0.4)**	218M
Mapping + QGen	47.6 (0.4)	72.6 (0.5)	218M
TF-IDF + QGen	47.7 (0.5)	72.9 (0.5)	218M
LM (Song et al., 2020) + QGen	48.2 (0.3)	72.7 (0.3)	218M
Fixed score (0.4) + QGen	46.9 (0.4)	72.1 (0.4)	218M
Fixed score (0.6) + QGen	47.8 (0.2)	72.5 (0.4)	218M

Finally, in step **C**, we calculate the cosine similarity between documents based on the model from step **A** and establish links when the similarity surpasses a score from step **B**.

4 Results and Discussions

Experimental Setup The details of the experimental setup are covered in Appendix A, where we empirically set two hyperparameters in UDL as γ=0.7 and δ=0.4, and reported the averaged NDCG@k (N@k) and Recall@k (R@k), along with the standard deviation (SD). For reproducibility, the training framework is covered in Appendix B, and the code is included in the supplementary material. Steps of fine-tuning are as follows: **(1)** Classifying linked and unlinked documents based on UDL, taking into account the order of the linked ones. **(2)** Feeding them as the inputs to the models and generating the synthetic queries with the same process as the original approach (e.g., model or prompt-based generations). **(3)** Fine-tuning the IR models based on generated queries.

Research Questions We aim to address four research questions (RQs): **RQ1.** What is the most suitable query augmentation method in zero-shot IR? **RQ2.** How does UDL enhance zero-shot IR? **RQ3.** How well does UDL generalize? **RQ4.** Is UDL competitive with state-of-the-art (SOTA)?

Main Results Table 2 shows averaged results based on different query augmentations where we generated the same number of queries for each method. The overall trend of LM-based approaches outperforming simpler methods persists when UDL is added. However, a relatively parameter-efficient

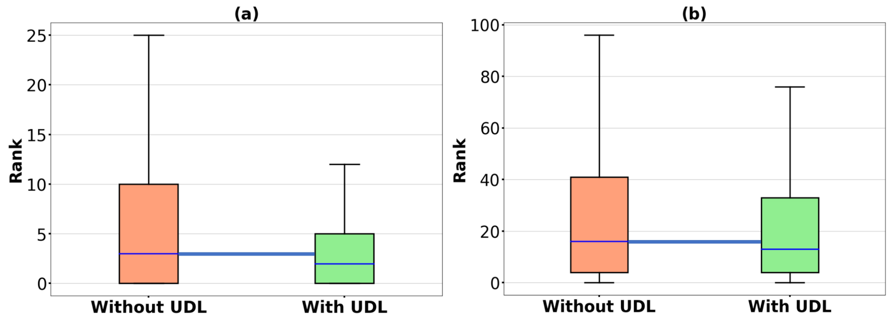

Figure 2: Distribution of rank of correctly classified queries when k=100 in NFCorpus, SciFact, ArguAna. (a) Single linked query-document. (b) Multiple linked query-documents. Blue line: Median value.

combination of UDL+QGen (218M) showed the best performance overall (**RQ1**), outperforming UDL+OpenLLaMA (3.1B). This promises significant savings of computational resources at scale. From our initial investigation, we found that OpenLLaMA tends to become more verbose after incorporating UDL, which may increase the risk of hallucination. In contrast, QGen generates more concise queries that are likely more accurate and relevant to the document. Additionally, we did not modify the LLM prompts based on UDL in this work, which presents a valuable future direction to optimize the prompts to better cover linked ones.

Furthermore, we ablated the document merging mechanism of UDL by generating the synthetic queries from each document individually and mapping them to documents found by the linking procedure (Mapping+QGen in Table 2). While this still outperformed the corresponding baseline (QGen), it performed worse than complete UDL. This suggests that generating queries from the merged documents improves model generalization by introducing harder queries with increased ambiguity compared to the original. Indeed, Table 1 anecdotally shows that resulting queries fit both linked documents and are generally less specific. Besides, the linking mechanism itself provides a more exhaustive way of identifying positive query-document pairs, improving the performance (**RQ2**). Figure 2 illustrates this behavior: Distributions with UDL are more compact, have fewer outliers, and allocate higher ranks for relevant documents.

Lastly, we investigated the influence of decisions in UDL separately. We compared the results between fixed similarity models (i.e., TF-IDF or LM+QGen) and flexible ones (i.e., UDL+QGen) where the latter excels. Also, we tested the results by fixing the similarity scores (i.e., Fixed score (0.4) or Fixed score (0.6)+QGen) and LM where flexible scores from UDL enhances the performance. Therefore, our evolved approach with flexible choices of the similarity models and scores promises the results.

Hyperparameters Choice Figure 3 shows the grid search for UDL's hyperparameters using NFCorpus, SciFact, and ArguAna yielding γ=0.7 and δ=0.4 as most optimal. (see Tables 14 and 15 for detailed results). We also checked the quality between synthetic queries and the offered train queries in used datasets. Detail of logic is shown in Algorithm 2 where 93% of synthetic queries generated from linked documents in UDL have sufficient quality as the train set to map the relevant documents.

Does UDL generalize? Table 3 compares the results of off-the-shelf models to those that have been fine-tuned across various models and English datasets. Interestingly, fine-tuning with QGen does not always improve the results, especially in high-performance models (e.g., All-MPNet). This suggests that synthetic queries can potentially decrease domain adaptation. Generally, we observe further improvements with UDL, except for SCIDOCS with All-MPNet. In such cases, UDL remains su-

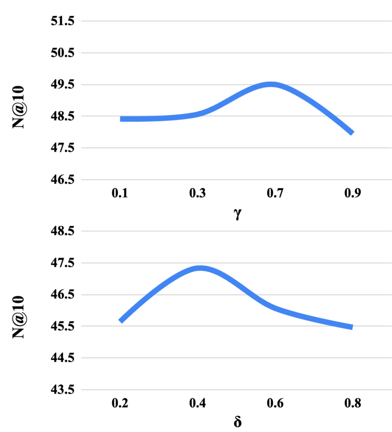

Figure 3: Grid search for γ and δ.

Table 3: Performances in English datasets. †: In-domain result since Quora was exposed for pre-training before fine-tuning with UDL. SD is always lower than 0.7. QGen and UDL+QGen have same number of generated queries.

Model	Data / Metric / Method	NFCorpus N@10	NFCorpus R@100	SciFact N@10	SciFact R@100	ArguAna N@10	ArguAna R@100	SCIDOCS N@10	SCIDOCS R@100	Climate-FEVER N@10	Climate-FEVER R@100	TREC-COVID N@10	TREC-COVID R@100	Quora N@10	Quora R@100
All-MPNet	Off-the-shelf	33.3	33.9	65.6	94.2	46.5	98.7	**23.8**	**55.0**	22.0	54.5	51.3	10.6	87.5†	99.6†
	QGen	33.1	31.3	65.2	91.6	53.3	98.8	19.1	44.4	23.8	54.9	59.8	10.8	86.0†	99.2†
	UDL + QGen	**35.9**	**34.9**	**67.1**	**94.8**	**61.0**	**99.5**	22.5	51.3	**24.1**	**55.4**	**69.5**	**12.2**	**88.1**†	**99.7**†
Distilled-BERT	Off-the-shelf	25.6	23.3	53.8	84.6	42.6	94.6	13.3	29.7	20.2	44.6	47.8	7.2	85.5	98.9
	QGen	29.0	27.1	59.6	90.1	50.3	98.5	14.4	33.1	22.0	52.3	56.9	9.8	84.5	98.7
	UDL + QGen	**31.2**	**30.8**	**61.5**	**90.7**	**55.8**	**99.2**	**16.6**	**40.5**	**22.3**	**52.8**	**61.7**	**10.9**	**85.8**	**99.1**
SGPT	Off-the-shelf	21.7	23.3	54.3	85.7	41.1	94.6	11.7	26.9	20.8	45.5	57.2	9.3	81.7	97.8
	QGen	24.1	23.8	56.8	88.9	47.4	96.9	12.6	29.8	21.1	48.0	61.6	9.5	83.9	98.6
	UDL + QGen	**24.6**	**26.0**	**57.4**	**90.0**	**52.0**	**99.1**	**15.3**	**37.1**	**21.5**	**48.4**	**64.5**	**10.6**	**85.0**	**99.0**
M-Distilled USE	Off-the-shelf	20.0	24.2	39.0	74.7	48.7	97.1	9.3	27.5	13.0	37.5	23.9	3.5	82.4	98.4
	QGen	24.8	24.7	48.9	81.9	47.9	97.3	13.5	32.0	16.3	40.0	57.0	10.6	83.4	98.6
	UDL + QGen	**26.9**	**27.9**	**49.9**	**84.1**	**49.1**	**98.5**	**15.1**	**38.3**	**16.7**	**42.7**	**62.0**	**11.5**	**84.3**	**99.0**

Table 4: Performances in non-English datasets where SD is always lower than 0.7.

Model	Data / Metric / Method	ViHealthQA N@10	ViHealthQA R@100	GermanQuAD N@10	GermanQuAD R@100
M-Distilled USE	Off-the-shelf	9.3	21.6	33.4	67.0
	QGen	22.2	33.8	31.7	65.8
	UDL + QGen	**23.0**	**34.8**	**34.7**	**69.0**
V-SBERT	Off-the-shelf	13.8	27.6	-	-
	QGen	22.9	33.6	-	-
	UDL + QGen	**23.8**	**34.8**	-	-
V-SimeCSE	Off-the-shelf	10.9	23.4	-	-
	QGen	22.5	33.4	-	-
	UDL + QGen	**23.4**	**34.6**	-	-
G-Electra	Off-the-shelf	-	-	25.0	53.5
	QGen	-	-	28.1	59.7
	UDL + QGen	-	-	**30.6**	**60.8**
G-XLM-R	Off-the-shelf	-	-	8.3	24.7
	QGen	-	-	36.0	70.5
	UDL + QGen	-	-	**36.6**	**71.2**

perior to naive fine-tuning. Table 4 demonstrates the results of UDL compared to the off-the-shelf models in Vietnamese and German datasets. The findings show the superiority of UDL when applied to non-English languages which confirms the flexibility of UDL. Table 5 covers the results in MA-Amazon (Reddy et al., 2022) with our approach and compares them with SOTA. This dataset contains interactions between user search queries and product information, along with relevance labels, making it well-suited for evaluating the extensibility of our method in real-world scenarios. Similar to the previous experiments, QGen improves the zero-shot performances where it is further enhanced consistently with the UDL approach. Therefore, our UDL is still generalized properly in potential real-world implementations. Even if SOTA models have bigger sizes and access to real user queries for pre-training and fine-tuning, the combination of UDL and QGen outperforms them significantly. Note that SOTA models consist of larger parameters and utilize the 482K unique documents for pre-training and 17K query-document pairs for fine-tuning. This confirms both the cost-effectiveness and resource-effectiveness of the UDL to achieve

Table 5: Performances in shopping query dataset where SD in Distilled-BERT is always under 0.4. SOTA results are exported from Sun et al. (2023).

Model	Method	Data	N@50	R@100	R@500	# Parameters
Distilled-BERT	Off-the-shelf	Document	39.0	57.8	73.5	66M
	QGen		43.5	65.2	80.6	
	UDL + QGen		**44.6**	**66.8**	**82.5**	
BIBERT	Pre-training + Fine-tuning	Query + Document	40.1	61.4	78.1	~109M
MTBERT			40.0	61.4	78.4	
MADRAL			40.4	61.7	78.5	
ATTEMPT			41.0	62.3	79.2	

Table 6: Comparison with SOTA in zero-shot scenarios. UDL: Fine-tuning All-MPNet with UDL.

Model	BM25	TAS-B	Contr-iever	SPLA-DE++	ANCE	COCO-DR	DRA-GON+	UDL
N@10	40.5	38.2	40.8	44.8	35.6	45.3	43.8	**46.7**
R@100	50.1	51.6	54.5	53.7	46.7	53.9	53.4	**58.0**

better performance than SOTA. Thus, we can verify that UDL works well across multiple datasets, languages, and models (**RQ3**).

A comparison between SOTA and QGen with UDL in English datasets is shown in Table 6 where all IR models have approximately 100M parameters for each encoder. Notably, All-MPNet with UDL wins others, demonstrating the superiority of UDL (**RQ4**). In the case of UDL implementation, some of the SOTA models were exposed to the documents of the target dataset during pre-training, but our method achieved better results. Lastly, we focused on directly fine-tuning with UDL, which could be extended to other applications like document expansion. This highlights the versatility of UDL for various tasks and models.

5 Conclusions

We propose a novel UDL to mitigate the limitations of conventional fine-tuning of IR models in zero-shot. UDL uses entropy and NER to tailor a linking method for each dataset with diverse tasks. Our comprehensive experiments show the effectiveness of UDL across various datasets and models.

6 Limitations

The proposed UDL offers significant advantages as an application. However, there are three possible limitations to consider. Firstly, while we consistently surpassed naive fine-tuning, there is an inherent limit to the enhancements. The performance of the retrieval model is influenced by the quality of synthetic queries. In general, the advanced pseudo-query generation methods manage multiple documents more effectively, indicating a valuable future direction to combine UDL with competitive pseudo-query generation approaches for further improvement. It also highlights the importance of selecting appropriate query augmentation strategies early in the project. Secondly, there is potential to introduce dynamic criteria, such as γ and δ in UDL, which were empirically defined in this study. Adjustments could be made for each candidate document, tailored to the similarities between documents and their types. Lastly, our comprehensive evaluation of UDL spanned ten datasets with diverse domains and languages (see Tables 3 - 5). There is a scope to extend this to larger documents and other languages, which was challenging due to computational resource constraints. These identified limitations present valuable research directions for those considering the proposed UDL in their applications.

References

Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. 2004. Umass at trec 2004: Novelty and hard. *Computer Science Department Faculty Publication Series*, page 189.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *Advances in Information Retrieval*, pages 716–722, Cham. Springer International Publishing.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

ClearNLP. 2015. Constituent-to-dependency conversion. [Accessed: 2024-06-12].

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

Together Computer. 2023. Redpajama-data: An open source recipe to reproduce llama training dataset. [Accessed: 2024-06-12].

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Common Crawl. 2007. Common crawl. [Accessed: 2024-06-12].

Kornél Csernai. 2017. First quora dataset release: Question pairs. [Accessed: 2024-06-12].

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims. *arXiv preprint arXiv:2012.00614*.

Christiane Fellbaum. 2005. Wordnet and wordnets. In Alex Barber, editor, *Encyclopedia of Language and Linguistics*, pages 2–665. Elsevier.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From distillation to hard negative sampling: Making sparse neural ir models more effective. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 2353–2359.

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

Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama. [Accessed: 2024-06-12].

GENIA. 2007. Genia 1.0. [Accessed: 2024-06-12].

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.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. [Accessed: 2024-06-12].

Dae Yon Hwang, Yaroslav Nechaev, Cyprien de Lichy, and Renxian Zhang. 2023a. GAN-LM: Generative adversarial network using language models for downstream applications. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 69–79, Prague, Czechia. Association for Computational Linguistics.

Dae Yon Hwang, Bilal Taha, and Yaroslav Nechaev. 2023b. EmbedTextNet: Dimension reduction with weighted reconstruction and correlation losses for efficient text embedding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9863–9879, Toronto, Canada. Association for Computational Linguistics.

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*.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*.

Ehsan Kamalloo, Nandan Thakur, Carlos Lassance, Xueguang Ma, Jheng-Hong Yang, and Jimmy Lin. 2023. Resources for brewing beir: Reproducible reference models and an official leaderboard. *arXiv preprint arXiv:2306.07471*.

Weize Kong, Swaraj Khadanga, Cheng Li, Shaleen Kumar Gupta, Mingyang Zhang, Wensong Xu, and Michael Bendersky. 2022. Multi-aspect dense retrieval. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3178–3186.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2022. *Pretrained transformers for text ranking: Bert and beyond*. Springer Nature.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*.

Alisa Liu, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2022. Wanli: Worker and ai collaboration for natural language inference dataset creation. *arXiv preprint arXiv:2201.05955*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Timo Möller, Julian Risch, and Malte Pietsch. 2021. Germanquad and germandpr: Improving non-english question answering and passage retrieval. *arXiv preprint arXiv:2104.12741*.

Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*.

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. PhoBERT: Pre-trained language models for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1037–1042.

Nhung Thi-Hong Nguyen, Phuong Phan-Dieu Ha, Luan Thanh Nguyen, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. 2022. Spbertqa: A two-stage question answering system based on sentence transformers for medical texts. In *International Conference on Knowledge Science, Engineering and Management*, pages 371–382. Springer.

OntoNotes. 2013. Ontonotes release 5.0. [Accessed: 2024-06-12].

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Chandan K Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping queries dataset: A large-scale esci benchmark for improving product search. *arXiv preprint arXiv:2206.06588*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. *arXiv preprint arXiv:2104.07540*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*.

Xiaojie Sun, Keping Bi, Jiafeng Guo, Xinyu Ma, Yixing Fan, Hongyu Shan, Qishen Zhang, and Zhongyi Liu. 2023. Pre-training with aspect-content text mutual prediction for multi-aspect dense retrieval. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4300–4304.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2021. Trec-covid: constructing a pandemic information retrieval test collection. In *ACM SIGIR Forum*, volume 54, pages 1–12. ACM New York, NY, USA.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

Orion Weller, Kyle Lo, David Wadden, Dawn Lawrie, Benjamin Van Durme, Arman Cohan, and Luca Soldaini. 2024. When do generative query and document expansions fail? a comprehensive study across methods, retrievers, and datasets. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1987–2003, St. Julian's, Malta. Association for Computational Linguistics.

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*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Peilin Yang and Jimmy Lin. 2019. Reproducing and generalizing semantic term matching in axiomatic information retrieval. In *Advances in Information Retrieval*, pages 369–381, Cham. Springer International Publishing.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. 2019. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*.

Yue Yu, Chenyan Xiong, Si Sun, Chao Zhang, and Arnold Overwijk. 2022. Coco-dr: Combating distribution shifts in zero-shot dense retrieval with contrastive and distributionally robust learning. *arXiv preprint arXiv:2210.15212*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International conference on machine learning*, pages 11328–11339. PMLR.

Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. 2021. Mr. TyDi: A multi-lingual benchmark for dense retrieval. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 127–137, Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Setup

Databases We tested ten datasets where the summary of the database is shown in Table 7: NFCorpus (Boteva et al., 2016) has automatically extracted relevance judgments for medical documents. SciFact (Wadden et al., 2020) consists of expert-annotated scientific claims with abstracts and rationales. ArguAna (Wachsmuth et al., 2018) contains the pairs of argument and counterargument from the online debate. SCIDOCS (Cohan et al., 2020) has seven document-level tasks from citation prediction, document classification, and recommendation. Climate-FEVER (Diggelmann et al., 2020) consists of real-world claims regarding climate-change with manually annotated evidence sentences from Wikipedia. TREC-COVID (Voorhees et al., 2021) contains the COVID-related topics with a collection of literature articles where biomedical experts measure the relevancy between articles and topics. Quora (Csernai, 2017) is built for identifying the duplicate question which is necessary for a scalable online knowledge-sharing platform. GermanQuAD (Möller et al., 2021) is high-quality and human-labeled German dataset which includes the self-sufficient questions with all relevant information. ViHealthQA (Nguyen et al., 2022) consists of health-interested QA in Vietnamese. Multi-Aspect Amazon ESCI Dataset (MA-Amazon) (Reddy et al., 2022) has user queries for product search and long lists of product information like title, description, brand, color with four relevance labels.

Models In this work, we considered the diverse sets of models where the summary of them is covered in Table 8: For query augmentation, we tested five pre-trained models: PEGASUS (Summarization) (Zhang et al., 2020), T5-Base (QGen) (Raffel et al., 2020) for English datasets, mT5-Base (QGen) (Xue et al., 2020) for Vietnamese and German databases, Flan T5-Base (Flan) (Chung et al., 2024), OpenLLaMA (Geng and Liu, 2023; Computer, 2023; Touvron et al., 2023).

For retrieval task, eight pre-trained retrieval models are experimented: M-Distilled USE (Yang et al., 2019), All-MPNet (Song et al., 2020), Distilled-BERT (Sanh et al., 2019), SGPT (Muennighoff, 2022), V-SBERT (Nguyen and Nguyen, 2020), V-SimeCSE (Gao et al., 2021), G-Electra (Clark et al., 2020), G-XLM-R (Conneau et al., 2020).

For pre-trained LM in similarity model, we employed three pre-trained models: All-MPNet (Song et al., 2020) for English datasets, V-SBERT (Nguyen and Nguyen, 2020) for Vietnamese database, G-BERT (Chan et al., 2020) for German dataset.

For comparison, ten SOTA models are investigated: TAS-B (Hofstätter et al., 2021), Contriever (Izacard et al., 2021), SPLADE++ (Formal et al., 2022), ANCE (Xiong et al., 2020), COCO-DR (Yu et al., 2022), DRAGON+ (Lin et al., 2023), BIBERT (Lin et al., 2022), MTBERT (Kong et al., 2022), MADRAL (Kong et al., 2022), ATTEMPT (Sun et al., 2023).

Table 7: Details of datasets used where we only cover the size of test set which is our point of interest. Note that ViHealthQA did not report the licenses in the paper or a repository.

Dataset	Language	Size of Test Set		License
		# Queries	# Document	
NFCorpus	English	323	3K	CC-BY-SA-4.0
SciFact	English	300	5K	CC-BY-NC-2.0
ArguAna	English	1K	8K	CC-BY-SA-4.0
SCIDOCS	English	1K	25K	CC-BY-4.0
Climate-FEVER	English	1K	5M	CC-BY-SA-4.0
TREC-COVID	English	50	171K	CC-BY-SA-4.0
Quora	English	10K	523K	CC-BY-SA-4.0
GermanQuAD	German	2K	2M	CC-BY-4.0
ViHealthQA	Vietnamese	2K	9K	-
MA-Amazon	English	8K	164K	Apache-2.0

Table 9 describes the details of NER models used in this work. NER model trained with general sources (N_g) covers the diverse types of general entities while NER model trained with specialized sources (N_s) addresses the various types of medical and scientific entities mostly related to the jargon.

UDL Details For the UDL, we tested three different methods (Concatenation, Summarization, Random permutation of the order) to link the two closest documents where we empirically selected Concatenation at last (Table 16). We generated three synthetic queries for each linked and unlinked documents, noting that there is a limitation to improvements based on size (Table 11). To decide the similarity model, we considered scikit-learn [3] for TF-IDF, while All-MPNet (Song et al., 2020), V-SBERT (Nguyen and Nguyen, 2020), and G-BERT (Chan et al., 2020) were used for English, Vietnamese, and German datasets in pre-trained LM. The spaCy (Honnibal et al., 2020) is utilized to import the N_g ($en\_core\_web\_trf$ [4]) and N_s ($en\_core\_sci\_scibert$ [5]). As shown in Tables 14 and 15, we empirically decided the hyperparame-

[3] `https://scikit-learn.org/stable/`
[4] `https://spacy.io/models/en`
[5] `https://allenai.github.io/scispacy/`

Table 8: Details of models used. Some models did not clearly report the licenses in the paper or a repository.

Model	Language	Number of Parameters	License
PEGASUS (Summarization)	English	569M	Apache-2.0
T5-Base (QGen)	Multilingual	109M	Apache-2.0
mT5-Base (QGen)	Multilingual	390M	Apache-2.0
Flan T5-Base (Flan)	Multilingual	248M	Apache-2.0
OpenLLaMA	Multilingual	3B	Apache-2.0
M-Distilled USE	Multilingual	135M	Apache-2.0
All-MPNet	English	109M	Apache-2.0
Distilled-BERT	English	66M	Apache-2.0
SGPT	English	125M	MIT
V-SBERT	Vietnamese	135M	-
V-SimeCSE	Vietnamese	135M	-
G-Electra	German	110M	-
G-XLM-R	German	278M	MIT
G-BERT	German	109M	MIT
TAS-B	English	66M	Apache-2.0
Contriever	English	109M	CC-BY-NC-4.0
SPLADE++	English	139M	Apache-2.0
ANCE	English	124M	Apache-2.0
COCO-DR	English	109M	MIT
DRAGON+	English	109M	CC-BY-NC-4.0
BIBERT	English	~109M	-
MTBERT	English	~109M	-
MADRAL	English	~109M	-
ATTEMPT	English	~109M	Apache-2.0

Table 9: Details of NER models used.

	General NER (N_g)	Specialized NER (N_s)
Types of Entities	*General:* Numerals, Date, Event, Objects, Countries, Language, Person, Quantity Monetary, Time, Companies, Mountain ranges ...	*Medical:* Organism, Gene, Chemical, Pathological formation, Cell, Tissue ... *Scientific:* Task, Method, Metric, Material, Professional and Generic terms ...
Sources	OntoNotes 5 (OntoNotes, 2013) ClearNLP (ClearNLP, 2015) WordNet 3.0 (Fellbaum, 2005) RoBERTa-Base (Liu et al., 2019)	OntoNotes 5 (OntoNotes, 2013) Common Crawl (Crawl, 2007) GENIA 1.0 (GENIA, 2007) SciBERT (Beltagy et al., 2019)
Vocabulary Size	50K	785K
License	MIT	CC-BY-SA-3.0

ters (γ=0.7, δ=0.4) to get the promising results. For datasets with more than 1M documents, we considered a maximum 30K documents during query augmentations and UDL to meet the resource constraints, except for MA-Amazon where we used 60K documents. We trained the retrieval model three times with different random seeds to account for random initialization. Currently, our suggested algorithm, UDL, will follow the MIT license.

B Notes on Reproducibility

Total Computational Budget and Infrastructure used For UDL and fine-tuning the retrieval models, we employed the Intel(R) Xeon(R) CPU @ 2.20GHz and NVIDIA A100. All of them used RAM 80GB and we trained three times with different seeds to get the averaged results. For decision of similarity model, TF-IDF required about 34 seconds and LM needed about 174 seconds for 10K documents. For decision of similarity score, it took about 787 seconds for 10K documents. The query augmentation for 10K documents took about 6699 seconds for summarization, 2970 seconds for Flan, 12542 seconds for OpenLLaMA and 721 seconds for QGen. Other augmentations like random cropping and RM3 are fast enough to be negligible. Fine-tuning is affected heavily by the size of the model and synthetic queries. For example, it took about 20 seconds when training a 135M parameters model with 11K queries and 4K documents. Note that, these computational costs do not affect the inference time during retrieval. In all experiments, we mainly utilized the BEIR environment (Thakur et al., 2021; Kamalloo et al., 2023) to evaluate the retrieval performances.

Table 10: Hyperparameters in UDL.

Parameter	Setting
γ	0.7
δ	0.4
Max features in TF-IDF	36000
Epoch	1
Learning Rate	2e-5
Weight Decay	1e-2

Table 11: The effect of size of synthetic queries generated from QGen. Retrieval model is Distilled-BERT.

Metrics	NFCorpus 1 synthetic queries	3 synthetic queries	9 synthetic queries
N@1	35.9	**36.9**	36.2
N@10	27.9	**29.0**	28.4
N@100	25.0	25.8	**26.1**
R@1	4.3	**4.5**	4.3
R@10	13.2	**13.6**	13.4
R@100	26.0	**27.1**	26.3

Hyperparameters In Table 10, we cover all the hyperparameters considered in this work which are based on the empirical results. During fine-tuning, we used *MultipleNegativesRankingLoss* [6] with *AdamW (warmup scheduler=10% of train set)* (Loshchilov and Hutter, 2017). During the evaluation, *cosine-similarity* is utilized to retrieve the documents given queries.

[6] https://www.sbert.net/docs/package_reference/losses.html

Table 12: Examples of terms from TF-IDF according to the Shannon Entropy.

Shannon Entropy	Examples of Terms
Greater than 1	the, this, an, a, yes, no, is, was, has, have, old, new ...
Less than 1	hala, storms, ipad, sari, coax, intermediate, pulse, peculiarities, swearing, enlisting, endures, fervour ...

Table 13: Decisions of similarity model and type of document from UDL in each dataset.

Dataset	Decisions of the UDL	
	Model	Type of Document
NFCorpus	LM	Specialized
SciFact	TF-IDF	Specialized
ArguAna	LM	General
SCIDOCS	LM	Specialized
Climate-FEVER	TF-IDF	General
TREC-COVID	TF-IDF	Specialized
Quora	LM	General
GermanQuAD	TF-IDF	General
ViHealthQA	LM	Specialized
MA-Amazon	LM	General

C Term Entropy in UDL

Equation (1) explains the term entropy measurement used in UDL.

$$E(X) = -\sum_{i=1}^{N} P(X_i) \log_2 P(X_i) \quad (1)$$

where E is the entropy, X is the term, $P(X_i)$ is the distribution of terms across documents, N is the number of documents.

D Ablation Study

Detailed Investigation of UDL Table 11 shows the limitation of improvement after increasing the size of synthetic queries which confirms the importance of UDL. Table 12 shows the examples of term entropy where article and relatively common words have entropy greater than 1 while the professional and relatively uncommon words have entropy less than 1. Table 13 covers the overall decisions of UDL in each dataset. Tables 14 and 15 reveal the details of ablation studies for hyperparameters in UDL. Table 16 explains the results depending on the different merging methods in UDL. Compared with random permutation, concatenation gives better results which reveals the importance of the order of sentences. Compared with summarization, concatenation shows better results which confirms the importance of the original structure of sentences.

Algorithm 2 Quality Checking

Data: Train queries and documents in each dataset and synthetic queries
Result: Sufficient quality of synthetic queries to map the used documents
Parameters: Queries in train set $Q = \{q_1 \ldots q_n\}$, synthetic queries $\hat{Q} = \{\hat{q}_1 \ldots \hat{q}_m\}$, documents used for generating synthetic queries and mapped by train queries $Doc = \{doc_1 \ldots doc_k\}$

1. Find train queries mapping the linked documents in UDL: q_i, doc_a, doc_b
2. Measure cosine-similarity in pairs of q_i-doc_a, q_i-doc_b: $Score(q_i, doc_a)$, $Score(q_i, doc_b)$
3. Measure cosine-similarity in pairs of $\hat{q}_j$-doc_a, $\hat{q}_j$-doc_b where $\hat{q}_j$ is generated from linked doc_a-doc_b: $Score(\hat{q}_j, doc_a)$, $Score(\hat{q}_j, doc_b)$
4. **if** $Score(q_i, doc_a) < Score(\hat{q}_j, doc_a)$ & $Score(q_i, doc_b) < Score(\hat{q}_j, doc_b)$ **then**
   $\hat{q}_j$ properly maps both documents

else
  if $Score(q_i, doc_a) < Score(\hat{q}_j, doc_a)$ **then**
    $\hat{q}_j$ appropriately maps doc_a
  end
  if $Score(q_i, doc_b) < Score(\hat{q}_j, doc_b)$ **then**
    $\hat{q}_j$ appropriately maps doc_b
  end
end

Quality of Synthetic Queries Algorithm 2 reveals the overall logic of quality checking based on the offered train set in NFCorpus and SciFact. We first found train data which covers same documents considered as linking in UDL. Then, we measured the cosine-similarity between the train query and relevant documents, and compared this with the cosine-similarity between the generated synthetic query and those same documents. If generated query has higher scores, this argues that our generated data has enough quality to link the single/multiple documents.

From our analysis, 93% of generated queries properly maps both documents where it increases up to 99% for single document. Thus, most of queries generated from linked documents in UDL have the sufficient quality to map the relevant documents without additional quality control.

Table 14: Different similarity models for UDL. Retrieval model is Distilled-BERT and similarity score is 0.6 for NFCorpus, Scifact and 0.4 for ArguAna. $\gamma = 0.7$ is our final decision.

Metrics	NFCorpus				SciFact				ArguAna			
	$\gamma = 0.1$	$\gamma = 0.3$	$\boldsymbol{\gamma = 0.7}$	$\gamma = 0.9$	$\gamma = 0.1$	$\gamma = 0.3$	$\boldsymbol{\gamma = 0.7}$	$\gamma = 0.9$	$\gamma = 0.1$	$\gamma = 0.3$	$\boldsymbol{\gamma = 0.7}$	$\gamma = 0.9$
N@1	37.7	37.6	**39.0**	35.8	49.2	49.0	**50.4**	49.6	29.2	30.1	**30.3**	27.7
N@10	30.5	30.4	**31.2**	28.9	60.1	60.1	**61.5**	61.1	54.6	55.2	**55.8**	53.9
N@100	28.4	28.5	**28.9**	25.2	65.1	**65.2**	64.9	64.1	57.9	**59.2**	**59.2**	55.4
R@1	4.3	4.3	**4.4**	3.9	46.8	46.5	**48.1**	48.0	29.0	29.5	**30.3**	27.7
R@10	14.2	14.3	**14.7**	13.2	75.2	72.5	**73.3**	73.2	84.0	84.3	**85.1**	78.8
R@100	30.1	30.3	**30.8**	27.8	88.4	88.2	**90.7**	90.2	99.1	98.7	**99.2**	98.4

Table 15: Different similarity scores for UDL. Retrieval model is Distilled-BERT and similarity model is fixed to TF-IDF. $\delta = 0.4$ is our final choice.

Metrics	NFCorpus				SciFact				ArguAna			
	$\delta = 0.2$	$\boldsymbol{\delta = 0.4}$	$\delta = 0.6$	$\delta = 0.8$	$\delta = 0.2$	$\boldsymbol{\delta = 0.4}$	$\delta = 0.6$	$\delta = 0.8$	$\delta = 0.2$	$\boldsymbol{\delta = 0.4}$	$\delta = 0.6$	$\delta = 0.8$
N@1	37.4	**39.2**	36.7	37.2	44.0	**50.4**	47.3	47.3	25.6	**26.8**	25.6	25.4
N@10	28.1	**29.0**	28.6	28.1	57.9	**61.5**	59.3	58.8	50.9	**51.5**	50.3	49.5
N@100	25.3	**26.3**	26.1	26.0	60.8	**64.9**	63.2	62.6	54.6	**55.7**	54.6	54.1
R@1	4.4	**4.6**	3.8	4.0	41.8	**48.1**	44.9	44.8	25.6	**26.8**	25.6	25.1
R@10	12.8	12.9	**13.4**	13.2	71.2	73.3	**73.9**	71.4	79.3	**80.1**	79.3	77.0
R@100	25.9	**27.3**	26.6	26.1	88.3	**90.7**	89.6	90.1	97.4	**98.4**	97.9	97.2

Table 16: Results according to the merging approaches in UDL. Random permutation: Concatenate two documents and then, randomly mix up the order. Summarization: Using Flan T5-Base (Chung et al., 2024), summarize each document separately and then, concatenate them. Title is always attached directly.

Metrics	NFCorpus			SciFact			ArguAna		
	Concatenation	Random Permutation	Summarization	**Concatenation**	Random Permutation	Summarization	**Concatenation**	Random Permutation	Summarization
N@1	**39.0**	37.5	38.6	**50.4**	47.3	48.3	**30.3**	29.6	23.4
N@10	**31.2**	30.0	29.6	**61.5**	58.9	59.4	**55.8**	54.8	45.9
N@100	**28.9**	28.4	28.0	**64.9**	62.6	63.4	**59.2**	58.1	51.5
R@1	**4.4**	4.0	4.3	**48.1**	44.9	45.9	**30.3**	30.0	23.4
R@10	**14.7**	14.2	13.5	**73.3**	72.5	72.0	**85.1**	83.9	73.7
R@100	**30.8**	30.1	30.0	**90.7**	89.2	90.3	**99.2**	98.7	98.0