Large language model augmented exercise retrieval for personalized language learning

Austin Xu[*]
Georgia Institute of Technology

Klinton Bicknell
Duolingo

Will Monroe
Duolingo

Abstract

We study the problem of zero-shot multilingual exercise retrieval in the context of online language learning, to give students the ability to explicitly request personalized exercises via natural language. Using real-world data collected from language learners, we observe that vector similarity approaches poorly capture the relationship between exercise content and the language users use to express what they want to learn. This semantic gap between queries and content dramatically reduces the effectiveness of general-purpose retrieval models pretrained on large-scale information retrieval datasets. We leverage the generative capabilities of large language models to bridge the gap by synthesizing hypothetical exercises based on the user's input, which are then used to search for relevant exercises. Our approach, which we call mHyER, outperforms several strong baselines, such as Contriever, on a novel benchmark created from publicly available Tatoeba data.

1 Introduction

Modern personalized education systems typically leverage the power of machine learning models to estimate user skill levels[1–7] and adaptively serve exercises to students [8–10]. Adaptivity, while a critical part of any personalized education system, is a *passive* form of personalization from the student's point of view: While exercises are tailored to an estimate of the student's skill level, this customization occurs behind the scenes, with no opportunity for students to specify particular characteristics of exercises. In this paper, we study a complementary form of *user initiated* personalization in the context of *online language learning*. In particular, students (referred to as "users") are given the ability to *explicitly* request learning content from a personalized education system, which returns relevant exercises from a fixed catalog for the user to do.

Online language learning is a natural setting for user initiated personalization, as people learn languages for very personal reasons: Some learn for fun, while others have specific goals, such as preparing for an international trip or developing language skills for business. Different reasons for learning lead to different needs for exercise content: Someone learning to write in a business setting may want extra practice with grammar or politeness, whereas the user learning for a vacation may want exercises about hotels or transportation. With the goal of allowing language learners to tailor an online learning experience to their own needs, we formalize the task of **exercise retrieval for user directed language learning** and evaluate machine learning models for this task. The goal of this task (Figure 1) is to retrieve relevant exercises from a set of existing exercises based on a user's input. In this setting, collecting relevance labels (i.e., pairs of user inputs and relevant exercises) is particularly challenging, as users will typically be presented with only a small number of exercises for any given input. As a result, we consider the *zero-shot* setting, where we do not have access to relevance labels for training. Concretely, we make the following contributions:

[*]Work done as an intern at Duolingo. Contact: axu@gatech.edu

NeurIPS'23 Workshop on Generative AI for Education (GAIED).

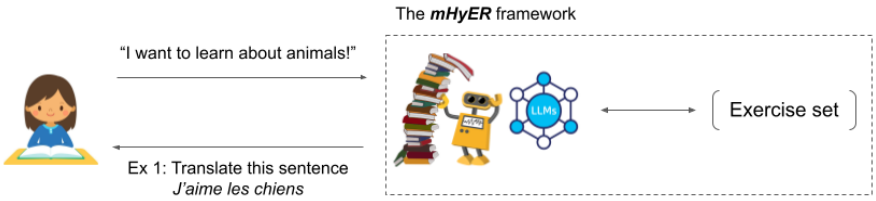

Figure 1: Exercise retrieval for user directed language learning and our proposed solution, multilingual Hypothetical Exercise Retrieval (mHyER). At a high level, users are allowed to provide *any* natural language input, and the goal is to retrieve exercises relevant to that input. Our method utilizes large language models to perform zero-shot exercise retrieval.

- We present our zero-shot retrieval approach, mHyER, in Section 2. We highlight a core challenge of this task and show that mHyER overcomes the pitfalls of direct vector similarity search.
- With no existing benchmarks for this task, we create the first benchmark in exercise retrieval with publicly available Tatoeba data and evaluate our method against several strong dense retrieval baselines in Section 3. Empirically, mHyER outperforms relevant baselines by a significant margin.

Related work. Exercise retrieval is naturally connected to the broad field of information retrieval, and in particular, dense retrieval [11, 12]. Dense retrieval focuses on retrieving documents based on similarity measured in a learned representation space. Zero-shot retrieval, or retrieval without training on task-specific relevance information, is of particular relevance to our task. Such methods typically rely on a supervised pretraining stage [13–15], where models are trained on large scale retrieval datasets, such as MS MARCO [16]. However, such supervised pretraining ultimately depends on the existence of suitable labeled datasets, which are not always readily available [17]. The rise of large language models (LLMs) with strong zero/few-shot performance in new domains has resulted in a line of research integrating LLMs into the retrieval pipeline. Such approaches typically rely on some combination of specialized prompting and synthesizing retrieval datasets to retrain retrieval models [18–21]. Our approach takes particular inspiration from HyDE [22], which utilizes a LLM to synthesize a hypothetical document, which is used then used with a pretrained encoder to retrieve documents via nearest neighbors. A fundamental step in any retrieval method is the representation space used for similarity comparisons. For the task of exercise retrieval, we focus on learning sentence embeddings, where pretrained language models such as BERT [23] or RoBERTa [24] serve as strong foundations. Contrastively learning sentence representations [17, 25–28] has become especially popular due to its simplicity and strong empirical performance. Of particular interest to the language learning setting is multilingual contrastive learning [29].

2 Problem setup and method

2.1 Exercise retrieval for user directed language learning

The goal of exercise retrieval for user directed language learning is to retrieve relevant exercises for a user given a text input from the user describing what they want to learn. In particular, we assume that user is taking a language learning course, which consists of two languages: the "first language" (i.e., a language they already know—not necessarily the user's native language) and the "second language" (i.e., the language they are learning), which we refer to as L1 and L2, respectively. The user completes *exercises*, which are drawn from a fixed set of N exercises $\mathcal{E} = \{e_1, \ldots, e_N\}$ that are at an appropriate level for the user. We can view this set of N exercises as samples from an unknown *exercise distribution*, which captures characteristics (style, length, content, etc) of exercises. Each exercise consists of two sentences $e_i^{(\texttt{L1})}$ and $e_i^{(\texttt{L2})}$. The user does the exercise by translating $e_i^{(\texttt{L1})}$ to the L2, with $e_i^{(\texttt{L2})}$ used as an example of a correct translation. The user provides some input t, and our objective is to retrieve the K (unique) exercises that are the most relevant based on input t in a zero-shot manner. That is, without using any labeled relevance data for training, we want to retrieve K unique exercises $e_1^\star, \ldots, e_K^\star$ that maximize the relevance distribution conditioned on user input t:

$$e_1^\star, \ldots, e_K^\star = \underset{\substack{e_1, \ldots e_K \in \mathcal{E} \\ e_i \neq e_j \quad \forall i,j}}{\arg\max} \prod_{i=1}^{K} p(e_i | T = t). \tag{1}$$

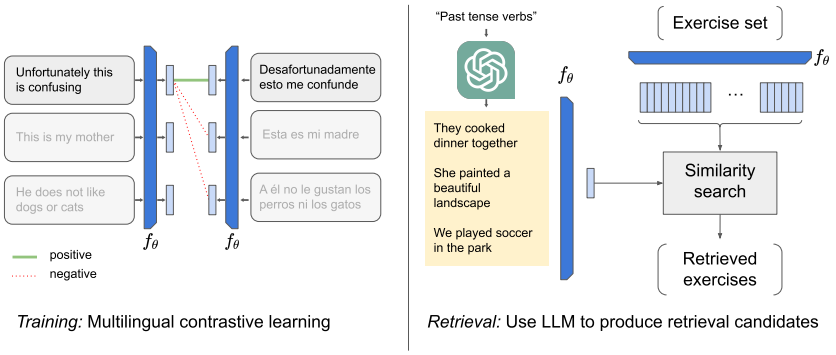

Figure 2: mHyER consists of two stages. Contrastive finetuning (left) is employed as a training stage to optimize our semantic similarity space for multilingual exercises. Then at retrieval time (right), a large language model is employed to synthesize hypothetical retrieval candidates. These retrieval candidates are then used in direct similarity search to retrieve exercises.

User inputs. The core of the personalized experience in this problem setting is allowing users to provide an input describing what they want to learn with *no restrictions on input content*, resulting in large number of potential input types. For example:

- **Topics:** Users request that the content of exercises matches a particular topic. Inputs such as "words about animals" or "countries" are such examples.
- **Grammar:** Because learning grammatical structure is a large component of language learning, grammar-focused inputs, such as "non-present tenses", are another common input type.
- **Culture:** Users can request to review culture-specific aspects of language, such as idioms, slang, or region-specific quirks. For example, a user learning Spanish may want to learn about "voseo", a region-specific grammatical concept in South America.
- **Learning process:** Users have requests that refer to elements of the process of learning a language, such as words that are hard to spell or pronounce.

As we discuss in Section 2.3, user inputs of these types result in a fundamental distributional gap between how users express their learning objectives and the content of the exercises.

2.2 Method

Baseline: direct search with similarity spaces. For text-based retrieval, most existing methods consist of a model f_θ (parametrized by θ) that maps natural language inputs (from the space of all text inputs $\mathcal{T}$) to some d-dimensional vector space: $f_\theta : \mathcal{T} \to \mathbb{R}^d$. Such models, also referred to as *encoders*, are typically neural networks trained such that texts with similar content are more similar in the representation space under some measure, like cosine similarity. That is, if $t_1, t_2 \in \mathcal{T}$ are similar in content, then $\text{sim}(f_\theta(t_1), f_\theta(t_2))$ is large (and positive). This similarity space suggests a natural approach for retrieving exercises: Pass each exercise e_i through the model f_θ to obtain $f_\theta(e_i)$.[2] Then, when a user provides an input t, pass t through the model to obtain $f_\theta(t)$ and return the K exercises with largest cosine similarity $\text{sim}(f_\theta(e_i), f_\theta(t))$. As we see in Section 2.3, direct similarity search often retrieves sentences featuring "language about language", which are often irrelevant to the user's input. This leads us to leverage the generative abilities of LLMs, as we discuss next.

mHyER. We propose mHyER, which after a multilingual contrastive training stage, retrieves exercises in a two-step manner. First, we sample a set of N_c hypothetical exercises from the exercise distribution *conditioned on the user input*. We call these sampled exercises our *retrieval candidates*. In principle, we do not have access to this exact distribution, but we can efficiently approximate sampling via LLM. Second, we use the retrieval candidates to perform similarity search via K-nearest neighbors. mHyER is inspired by two complementary methods: the multilingual contrastive learning approach of [29], and the HyDE retrieval method of [22], which uses synthesized retrieval candidates. We now discuss both the training and retrieval stages in greater detail.

[2]We slightly abuse notation here and write $f_\theta(e_i)$ to mean either $f_\theta(e_i^{(\text{L1})})$ or $f_\theta(e_i^{(\text{L2})})$. The choice to compare against the representation of the L1 or L2 sentence is explored in Section 3.

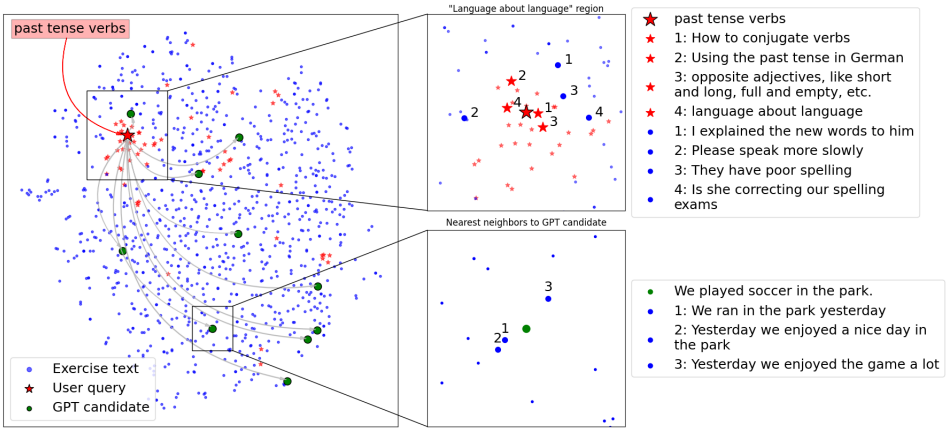

Figure 3: TSNE visualization of exercise, user input, and GPT-4-synthesized retrieval candidate representations in the representation space of a trained BERT sentence encoder (left). User inputs concentrate in the language about language region (top right), making direct similarity search sub-optimal. Retrieval candidates bridge the distribution gap between user inputs and exercise text and are close in similarity to exercises that meet the user's specifications (bottom right).

Stage 1: Learning a multilingual similarity space. While we operate in a setting where no explicit user relevance data is provided, the exercise sentence pairs $(e_i^{(\text{L1})}, e_i^{(\text{L2})})$ can be utilized to improve the similarity space. Naturally, we want the L1 and L2 sentences to be considered similar in the representation space. As a result, we take inspiration from [29] and utilize multilingual contrastive learning, an unsupervised approach that aims to learn a representation where similar items (called *positive* pairs) are closer together and dissimilar items (called *negative* pairs) are far apart. For exercise e_i, the contrastive loss $\mathcal{L}_i$ with a mini-batch of N_B sentence pairs is

$$\mathcal{L}_i = -\log \frac{\exp\left(\text{sim}\left(f_\theta(e_i^{(\text{L1})}), f_\theta(e_i^{(\text{L2})})\right)/\tau\right)}{\sum_{j=1}^{N_B} \exp\left(\text{sim}\left(f_\theta(e_i^{(\text{L1})}), f_\theta(e_j^{(\text{L2})})\right)/\tau\right)}, \tag{2}$$

where τ is the user-set temperature parameter and $\text{sim}(\cdot,\cdot)$ is the cosine similarity. In this work, rather than train a sentence encoder from scratch, we follow the commonly accepted practice of initializing our encoder with existing BERT sentence encoder checkpoints, and employing contrastive learning to finetune these checkpoints on exercise data.

Stage 2: Sampling retrieval candidates and exercise retrieval. A core component of mHyER is sampling from the exercise distribution conditioned on the user input. While we cannot sample directly from this distribution, we can approximate sampling with a LLM. In particular, we prompt the LLM with a fixed description of the exercise distribution and instruct the LLM to synthesize *hypothetical* exercises based on this description and based on a user's input. Crucially, we can synthesize exercises *without requiring any labeled examples*. To retrieve exercises, the LLM synthesizes K_h hypothetical exercises, which we denote $\tilde{e}_1, \ldots, \tilde{e}_{K_h}$. We then encode these hypothetical exercises via f_θ to obtain K_h vectors $f_\theta(\tilde{e}_1), \ldots, f_\theta(\tilde{e}_{K_h})$. To retrieve exercises, we retrieve the K exercises that have the highest similarity score compared to the average of the K_h vectors: $\frac{1}{K_h}\sum_{i=1}^{K_h} f_\theta(\tilde{e}_i)$. We use GPT-4 in this work, but in practice, any LLM of sufficient capacity can be used.

2.3 Bridging a fundamental distribution gap with mHyER

In an effort to better understand the task of retrieving exercises from user inputs, we crowdsourced a small dataset of user inputs from Duolingo users. We then contrastively finetune mBERT with roughly 40,000 real exercises from Duolingo, spanning 3 different language pairs. To get a sense of how contrastively learned similarity spaces reflect user inputs and exercise text, we visualize our collected data, along with a subsample of the exercises, via TSNE in Figure 3. This visualization reveals **a fundamental distribution gap between user inputs and exercise text**: How users describe what they want to learn occupies a distinct part of the representation space explicitly using words or phrases about language (e.g., "verbs", "past tense", "spelling"). We refer to this region as the "language about language" region. As a result, direct similarity search yields exercises that similarly

Table 1: Evaluation results on the Tatoeba Tags dataset. mHyER$_{[\texttt{model}]}$ indicates that contrastive finetuning was employed with [model] as the initial checkpoint. +[dataset] denotes that [dataset] data was used for contrastive finetuning. In all cases, mHyER outperforms relevant baselines dramatically, with large gains coming from finetuning on *out-of-distribution* data (Duo-OOD).

		English		English (L2) from Spanish (L1)				Spanish (L2) from English (L1)			
		AUC	P@15	AUC L1	AUC L2	P@15 L1	P@15 L2	AUC L1	AUC L2	P@15 L1	P@15 L2
Unsup. pretraining	BERT	0.495	0.032	0.481	0.428	0.019	0.020	0.492	0.505	0.044	0.020
	mBERT	0.468	0.037	0.446	0.487	0.038	0.040	0.469	0.442	0.039	0.019
	Contriever	0.536	0.161	0.542	0.523	0.112	0.073	0.529	0.549	0.165	0.087
	mContriever	0.571	0.064	0.438	0.503	0.051	0.063	0.559	0.564	0.061	0.027
	SimCSE	0.646	0.115	0.535	0.559	0.069	0.054	0.635	0.610	0.127	0.068
	mHyER$_{\text{mBERT}}$+en-from-es	0.722	0.225	0.686	0.701	0.227	0.208	0.710	0.696	0.243	0.242
	mHyER$_{\text{mBERT}}$+es-from-en	0.717	0.223	0.697	0.693	0.219	0.211	0.702	0.706	0.237	0.244
	mHyER$_{\text{mBERT}}$+Duo-OOD	0.752	0.211	0.734	0.738	0.215	0.206	0.739	**0.757**	0.225	0.242
	mHyER$_{\text{Contriever}}$+Duo-OOD	**0.768**	0.239	0.644	**0.780**	0.106	0.232	**0.749**	0.659	0.265	0.099
	mHyER$_{\text{mContriever}}$+Duo-OOD	0.729	**0.258**	**0.748**	0.723	**0.267**	**0.264**	0.713	0.744	**0.271**	**0.294**
Sup. pretraining	Contriever	0.541	0.164	0.491	0.492	0.120	0.086	0.530	0.492	0.180	0.105
	mContriever	0.575	0.104	0.548	0.510	0.126	0.108	0.560	0.581	0.112	0.101
	mHyER$_{\text{Contriever}}$+Duo-OOD	**0.775**	0.246	0.668	**0.797**	0.102	0.240	**0.760**	0.692	**0.268**	0.108
	mHyER$_{\text{mContriever}}$ +Duo-OOD	0.738	**0.255**	**0.761**	0.734	**0.260**	**0.264**	0.722	**0.752**	0.255	**0.280**

contain words explicitly about language. As shown in Figure 3, the input "past tense verbs" is most similar to exercises about language (e.g., "I explained the new words to him"). Figure 3 also highlights how synthesizing retrieval candidates helps bridge this distributional gap by "translating" the user's input (which is typically expressed in "language about language") to a hypothetical in-distribution exercise whose content satisfies the user input.

3 Datasets and experimental results

3.1 Tatoeba Tags dataset: Data and evaluation metrics.

Data. We construct a retrieval dataset from Tatoeba, a public database of sentences and translations that are tagged with grammatical concepts, language specific concepts, or topics. For example, the sentence "The brown bear is an omnivore" is tagged with "animals" and the sentence "That way I kill two birds with one stone" is tagged with "idiomatic expression". We treat each tag as a user input, and deem an exercise relevant if it has been tagged accordingly. We form 3 benchmarks for evaluation, collectively referred to as the Tatoeba Tags dataset: The English benchmark (139 tags and 89,392 English-only sentences), the Spanish from English benchmark (114 tags in Spanish and 49,258 Spanish-English sentence pairs), and the English from Spanish benchmark (108 tags in English and 46,837 English-Spanish sentence pairs). The Spanish from English and English from Spanish benchmark differ in sentences and number of tags due to considerations specific to learning direction; see Appendix A for details as well as the dataset curation process.

Metrics. We utilize Precision@K, which is a common metric in information retrieval that reports the fraction of the K retrieved exercises that are relevant. To compute Precision@K, we retrieve K sentences per user input (i.e., tag) and record the fraction of the K retrieved sentences tagged with the user input tag. Because the tagging of Tatoeba sentences is not exhaustive, the *absolute* values of reported Precision@K may be low, but *relative* performance still indicates how methods would perform if tagging was comprehensive. In light of this, we also view the problem as a more general problem of binary classification, where the goal is to predict whether an exercise is relevant or irrelevant, and report area under the ROC curve (AUC).

3.2 Experimental results

We evaluate mHyER against BERT and mBERT [23], as well as the following BERT-based models: Contriever, mContriever [17], and SimCSE [25]. In particular, we use the BERT $_{\texttt{base}}$ (110 million

Table 2: Ablation results on the Tatoeba Tags dataset. We experiment by removing either the contrastive finetuning stage or the retrieval candidate synthesis stage. `+GPT` indicates that retrieval candidates were used with no contrastive finetuning, whereas `+Duo-OOD` indicates that direct similarity search was performed after contrastively finetuning on `Duo-OOD`. In a vast majority of cases, both stages boost performance, with retrieval candidates generally contributing more in performance gains.

		English		English (L2) from Spanish (L1)				Spanish (L2) from English (L1)			
		AUC	P@15	AUC L1	AUC L2	P@15 L1	P@15 L2	AUC L1	AUC L2	P@15 L1	P@15 L2
Unsup. pretraining	mContriever	0.571	0.064	0.438	0.503	0.051	0.063	0.559	0.564	0.061	0.027
	mContriever +GPT	0.676	0.237	0.613	0.663	0.213	0.213	0.643	0.602	0.245	0.217
	mContriever +Duo-OOD	0.665	0.096	0.670	0.665	0.119	0.106	0.656	0.657	0.090	0.077
	$\text{mHyER}_{\text{mContriever}}$+Duo-OOD	**0.729**	**0.258**	**0.748**	**0.723**	**0.267**	**0.264**	**0.713**	**0.744**	**0.271**	**0.294**
Sup. pretraining	mContriever	0.575	0.104	0.548	0.510	0.126	0.108	0.560	0.581	0.112	0.101
	mContriever +GPT	0.731	0.250	0.642	0.724	0.238	0.243	0.706	0.636	**0.263**	0.258
	mContriever +Duo-OOD	0.672	0.106	0.678	0.677	0.128	0.120	0.662	0.661	0.113	0.091
	$\text{mHyER}_{\text{mContriever}}$+Duo-OOD	**0.738**	**0.255**	**0.761**	**0.734**	**0.260**	**0.264**	**0.722**	**0.752**	0.255	**0.280**

parameters) variant of each of the above methods. These methods achieve strong unsupervised performance in a variety of information retrieval and semantic text similarity tasks. Above, mBERT and mContriever were trained on multilingual data, while all other methods were trained on only English text. For a complete description of the baselines, please see Appendix B. We also experiment with supervised Contriever and mContriever, which are finetuned on MS MARCO [16]. We consider two retrieval settings: *Unsupervised pretraining*, where we start with a BERT checkpoint that has been pretrained in an unsupervised manner, and *supervised pretraining*, where we start with a BERT checkpoint that has been pretrained on MS MARCO [16], a large scale retrieval dataset that covers different tasks, such as passage ranking and keyphrase extraction. **We emphasize that mHyER is trained exclusively without labeled exercise relevance data**. Within each setting, we can retrieve exercises in two distinct ways: synthesizing retrieval candidates in the from language (L1) and doing similarity search on the L1 exercise texts, or synthesizing retrieval candidates in the learning language (L2) and performing similarity search on the L2 exercise texts. As a result, we report AUC and precision@15 in both the L2 and L1 settings. See Appendix C for more experimental details.

The evaluation results are presented in Table 1. We contrastively finetuned mBERT on the Spanish from English benchmark (denoted `es-from-en`) and the English from Spanish benchmark (denoted `en-from-es`), as well as the 40,000 *out-of-distribution* Duolingo sentence pairs mentioned in Section 2.3, which we refer to as Duo-OOD. In particular, we observe that finetuning on this dataset outperforms finetuning on in-distribution data. This surprising observation leads us to finetune Contriever and mContriever checkpoints with Duolingo data in both the unsupervised and supervised settings. In the unsupervised setting, we once again observe poor performance from direct similarity search baselines and sizable increases in performance when using mHyER: **up to 39% increases in AUC and more than doubling the performance of precision@15 for the best mHyER over the best direct similarity approach.** We observe similar gains in the supervised setting. Methods that use Contriever (pretrained only on English data) typically perform better when retrieving in English, whereas methods using mContriever typically perform better in multilingual settings.

Ablation study. The two key steps in mHyER are the multilingual contrastive pretraining stage and synthesizing retrieval candidates to use for retrieval. To characterize the relative contributions of each stage, we create variants of mHyER performing direct similarity search after contrastive pretraining or retrieving with GPT-synthesized retrieval candidates with a non-finetuned encoder (i.e., HyDE [22]). As shown in Table 2, the combination of both stages yields the best performance in the vast majority of cases. Utilizing only synthesized retrieval candidates results in the larger increases in precision compared to contrastive finetuning, the opposite is true for AUC. This suggests that the two steps drive performance increases in complementary ways: Contrastive finetuning changes the similarity space such that relevant exercises are closer to user inputs at a *global* level, resulting in increases in AUC (which measures a global ranking of predicitions). However, direct similarity search still cannot overcome distributional gaps, and hence, increases in precision@15 are low relatively. On the other hand, synthesizing GPT candidates directly improves retrieval quality, resulting in higher retrieval quality, but does not change representations, resulting in relatively lower increases in AUC.

References

[1] Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4:253–278, 1994.

[2] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. *Advances in Neural Information Processing Systems*, 28, 2015.

[3] Shalini Pandey and George Karypis. A self-attentive model for knowledge tracing. *International Educational Data Mining Society*, 2019.

[4] Dongmin Shin, Yugeun Shim, Hangyeol Yu, Seewoo Lee, Byungsoo Kim, and Youngduck Choi. Saint+: Integrating temporal features for ednet correctness prediction. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, pages 490–496, 2021.

[5] Ghodai Abdelrahman and Qing Wang. Knowledge tracing with sequential key-value memory networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2019.

[6] Shiwei Tong, Qi Liu, Wei Huang, Zhenya Hunag, Enhong Chen, Chuanren Liu, Haiping Ma, and Shijin Wang. Structure-based knowledge tracing: An influence propagation view. In *2020 IEEE International Conference on Data Mining*, pages 541–550. IEEE, 2020.

[7] Liangbei Xu and Mark A Davenport. Dynamic knowledge embedding and tracing. *International Educational Data Mining Society*, 2020.

[8] Zhengyang Wu, Ming Li, Yong Tang, and Qingyu Liang. Exercise recommendation based on knowledge concept prediction. *Knowledge-Based Systems*, 210:106481, 2020.

[9] Shuyan Huang, Qiongqiong Liu, Jiahao Chen, Xiangen Hu, Zitao Liu, and Weiqi Luo. A design of a simple yet effective exercise recommendation system in k-12 online learning. In *International Conference on Artificial Intelligence in Education*, pages 208–212. Springer, 2022.

[10] Peng Cui and Mrinmaya Sachan. Adaptive and personalized exercise generation for online language learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 10184–10198, 2023.

[11] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, 2019.

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

[13] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[14] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2345–2360, 2022.

[15] Yue Yu, Chenyan Xiong, Si Sun, Chao Zhang, and Arnold Overwijk. Coco-dr: Combating the distribution shift in zero-shot dense retrieval with contrastive and distributionally robust learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1462–1479, 2022.

[16] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[17] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research*, 2022.

[18] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2387–2392, 2022.

[19] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh International Conference on Learning Representations*, 2022.

[20] Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. Improving passage retrieval with zero-shot question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3781–3797, 2022.

[21] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*, 2022.

[22] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 1762–1777, 2023.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2019.

[24] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019.

[25] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 6894–6910, 2021.

[26] Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. Diffcse: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218, 2022.

[27] Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. ESimCSE: Enhanced Sample Building Method for Contrastive Learning of Unsupervised Sentence Embedding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3898–3907, 2022.

[28] Qinyuan Cheng, Xiaogui Yang, Tianxiang Sun, Linyang Li, and Xipeng Qiu. Improving contrastive learning of sentence embeddings from ai feedback. *arXiv preprint arXiv:2305.01918*, 2023.

[29] Yaushian Wang, Ashley Wu, and Graham Neubig. English contrastive learning can learn universal cross-lingual sentence embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9122–9133, 2022.

Acknowledgements

We thank Ali Malik, Stephen Mayhew, and Mark Davenport for their helpful comments, pointers, and feedback.

A Dataset details

To form the benchmarks, we collect all tags corresponding to the benchmark, filter out tags and sentences containing profanity, and then perform benchmark specific processing. We then keep only the tags with more than 20 sentences and download the corresponding sentences. The benchmark specific processing is done to better align the benchmark with how users would interact with real-world language learning courses. For example, a user's input will most likely be in the L1, meaning the tags must be translated to the appropriate language. Not only does the language of the tags matter, but the content of the tags as well. Some tags only make sense as user inputs for one course direction, but not the other. For example, a Spanish speaker learning English would not input "voseo" (a Spanish grammatical concept), nor would an English speaker learning Spanish input "British English".

B Baseline details

BERT and mBERT were trained in a self-supervised manner by using masked language modeling and next sentence prediction objectives, with the only difference being the training data (only English for BERT and a multilingual corpus for mBERT). Contriever and mContriever propose two new approaches in contrastively tuning BERT: (1) utilizing an inverse-cloze task and independent cropping as means of forming positive pairs and (2) utilizing a Momentum encoder to ensure better representation of negative items; please see [17] for specific details. Contriever is initialized with BERT and trained on CCNet and Wikipedia data, whereas mContriever was initialized with mBERT and trained on multiple languages in CCNet. We also consider supervised variants of Contriever and mContriever, which are finetuned on the MS MARCO, a large scale retrieval dataset. SimCSE considers dropout as a "minimal augmentation" and forms positive pairs in the contrastive loss by passing the same sentence through the encoder with different random dropout parameters. Starting with BERT, SimCSE is trained on Wikipedia data.

C Additional experiment details

For all experiments, we take the `[CLS]` representation as the sentence representation, except when working with Contriever/mContriever, where we use their custom mean pooling; see `https://huggingface.co/facebook/contriever` for further details. In all cases, we train mHyER using the setup of [29] (adapted from [25]) for 1 epoch with step size 0.001 and temperature parameter $\tau = 0.05$.