INSTRUCTOR: 自然语言指令创建广泛适用的文本嵌入模型的复现及其思考

朱昕睿

摘要

INSTRUCTOR 是生成文本嵌入的一种方法，该方法基于任务指令进行微调。传统的文本嵌入模型在应用到新任务或领域时通常会表现较差，需要进行特定任务或领域的微调。INSTRUCTOR 的目标是通过任务和领域描述来调整文本嵌入，使其适应不同的下游应用，而无需进行进一步的任务或领域特定微调。INSTRUCTOR 是一个单一的多任务模型，它根据文本输入和任务指令生成与任务和领域相关的嵌入。与之前的嵌入方法不同，INSTRUCTOR 将每个输入与其最终任务和领域的指令一起进行嵌入，从而将相同的输入针对不同的任务生成不同的嵌入向量。该方法的训练过程使用对比损失，通过最大化语义相关的文本对之间的相似性，同时最小化不相关的文本对之间的相似性。本文采用该方法在该文章提及的数据集中进行了测试与评估，并对该方法做出了思考与展望

关键词：文本嵌入；鲁棒性；对比损失；INSTRUCTOR；语义相关性

1 引言

文本嵌入将离散文本输入（例如句子、文档和代码）表示为可在许多下游任务中使用的固定大小的向量。这些任务包括语义文本相似性 [1, 4], 信息检索 [9, 11], 自动文本评估 [8], 即时检索以进行情境学习 [14] 等等。最近，我们看到学习文本嵌入方面取得了巨大进展 [5, 6, 12], 在预期任务或数据集上表现良好。

然而，大多数现有嵌入在应用于新任务或领域时可能会显着降低性能 [25], 例如，DPR [11] 的检索能力比文本相似性任务更强，而 SimCSE 则相反 [6]。此外，现有的嵌入在应用于相同类型的任务但应用于医学和金融等不同领域时通常表现不佳。解决这个问题的一种常见方法是进一步微调下游任务和域中数据集的嵌入，这通常需要大量带注释的数据 [7]。在本文中所采用的方法，假设可以使用任务和域描述将文本嵌入（即使对于相同的文本输入）调整为不同的下游应用程序，而无需进一步针对特定于任务或域的微调。

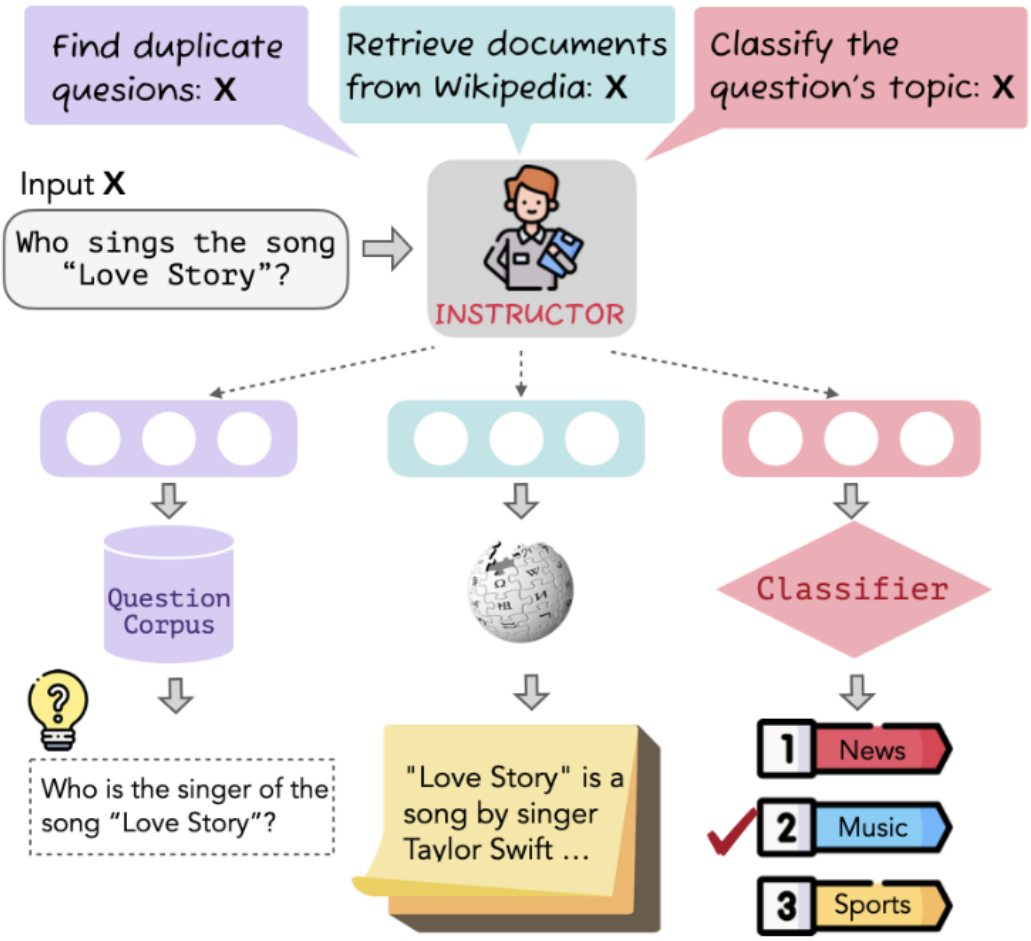

图 1. 在执行时，INSTRUCTOR 根据文本输入和任务指令生成嵌入

我们采用 INSTRUCTOR（Instruction-based Omnifarious Representations）方法，这是一个单一的多任务模型，可以在给定文本输入及其任务指令的情况下生成任务和领域感知的嵌入。它无需任何训练即可在大量下游嵌入任务中实现最先进的性能。我们方法的核心是基于指令的微调 [15, 27]：我们将每个输入及其最终任务嵌入在一起和领域指令，与之前仅接受文本输入的嵌入方法不同。INSTRUCTOR 将相同的输入嵌入到不同的向量中以实现不同的最终目标（例如，谁演唱了“Love Story”这首歌曲？在图 1中针对不同的任务嵌入到三个不同的向量中）。如图 2所示，INSTRUCTOR 在 MEDI 上进行训练，MEDI 是该文章 [24] 提出的的 330 个文本嵌入数据集，新添加了人工编写的任务指令注释。我们在所有数据集上使用对比损失来训练 INSTRUCTOR，从而最大化语义相关文本对之间的相似性，同时最小化不相关文本对。

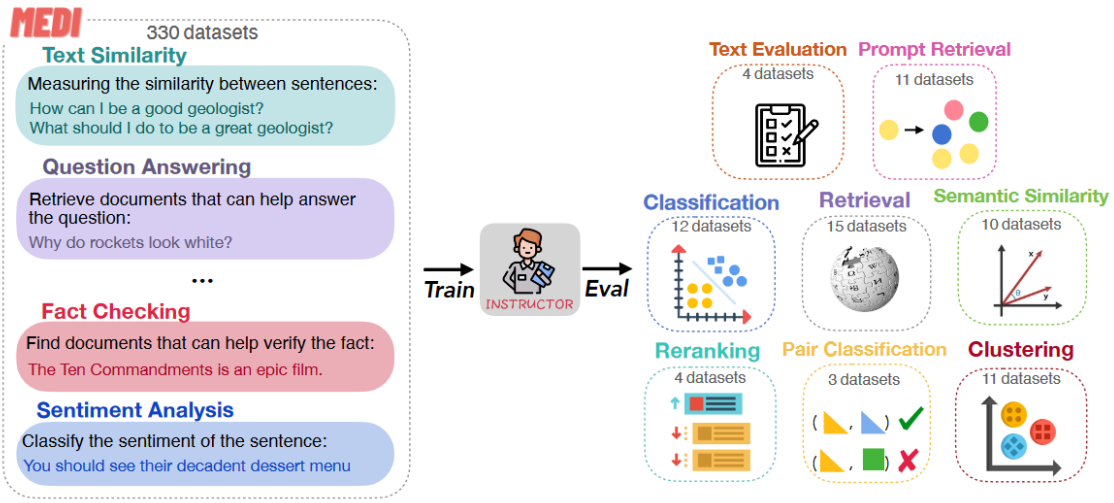

图 2. INSTRUCTOR 训练和评估流程

文章 [24] 在不同领域（例如金融、医学和新闻）和各种下游应用程序（总共 70 个嵌入评估数据集，其中 66 个在训练期间未见过）广泛评估 INSTRUCTOR，涵盖分类、语义文本相似性、信息检索、文本生成评估，以及上下文学习的提示检索。在 70 个不同的数据集上，INSTRUCTOR 的性能明显优于之前最先进的嵌入模型，平均性能提高了 3.4%。INSTRUCTOR 的性能也优于在没有任务指令的情况下训练的变体，这证明了指令对于创建任务感知嵌入的重要性。分析表明，指令微调解决了在不同数据集上训练单一模型的挑战。此外，他们证明 MEDI 的任务多样性使得 INSTRUCTOR 的性能对于指令中的释义特别稳健。总的来说，这些结果强烈表明指令微调应该广泛应用于文本嵌入，我们通过共享所有模型和代码来支持这一点。

2 相关工作

2.1 文本嵌入

文本嵌入在许多应用中都很有用，例如信息检索 [25]、文本相似性 [6]、上下文学习的提示检索 [23]、分类 [21] 等。许多先前的工作为不同的应用开发了不同的嵌入模型。例如，SBERT [21] 和 SimCSE [6] 仅应用于文本相似性和分类任务，而 DPR [11] 和 Contriever [10] 专注于信息检索。与仅在对称数据上训练的 Sentence-T5 或仅在非对称数据上训练的 GTR 不同，我们结合两组数据集并构建 MEDI，然后将其用于通过指令训练 INSTRUCTOR。穆尼尼霍夫等人 [17] 引入了大规模文本嵌入基准 (MTEB)，它可用于评估各种嵌入任务的嵌入模型，涵盖重排序、分类、信息检索、双文本挖掘、对分类、STS 和摘要。他们的基准测试表明，在一项任务上表现良好的模型可能在其他任务上表现不佳。现有嵌入模型的零样本传输能力较差，使得它们很难在只有很少标记数据可用的应用中使用。这促使我们开发一个适用于各种任务并对未见过的任务具有更好泛化能力的单一嵌入模型。最近提出了 E5 [26]，即弱监督对比预训练文本嵌入，它在 MTEB 基准上的各种任务中实现了强大的性能，与 INSTRUCTOR 相比，采用了更大的嵌入维度。

2.2 指令微调

最近的工作表明，指令微调语言模型可以在给定自然语言指令的情况下执行新任务 [15, 16, 26, 27]。尽管如此，指令微调仍有待在广泛适用的嵌入的背景下进行研究。在这项工作中，我们探索微调嵌入模型以遵循人类指令，其中指令指定了最终用例。并行工作证明指令可以促进信息检索 [2]，这与 INSTRUCTOR [24] 设计有关。他们使用指令构建了任务感知检索系统，并对检索任务进行了评估；INSTRUCTOR 构建了一个通用嵌入模型，其指令可应用于 8 个任务类别（图 2），包括检索、文本相似性、聚类和文本评估。

3 INSTRUCTOR

INSTRUCTOR 将输入与任务指令一起编码，从而提供可用于许多下游语言任务的特定于任务的表示，而无需任何额外的训练。在这里，我们介绍了 INSTRUCTOR 的架构（§ 3.1），介绍了我们如何执行基于多任务指令的微调（§ 3.2），并描述了我们如何收集和注释 MEDI 训练数据（§ 3.3）。默认情况下，我们将“任务”称为数据集，并在整篇论文中互换使用它们，而“任务类别”（例如检索）则包含许多任务。

3.1 嵌入架构

该作者 [24] 基于单一编码器架构构建 INSTRUCTOR [10]。继之前的工作之后，继续使用 GTR 模型作为骨干编码器（GTR-Base 用于 INSTRUCTOR-Base，GTRLarge 用于 INSTRUCTOR，GTR-XL 用于 INSTRUCTOR-XL）。GTR 模型从 T5 模型初始化，在网络语料库上进行预训练，并在信息搜索数据集上进行微调。GTR 模型系列中不同尺寸的可用性使我们能够探索指令微调嵌入模型的缩放行为。给定输入文本 x 和任务指令 I_x，INSTRUCTOR 对它们的串联 $I_x \oplus x$ 进行编码。然后，我们通过将均值池化应用于 x 中标记的最后隐藏表示来生成固定大小、特定于任务的嵌入 $E_I(I_x, x)$。

3.2 训练方法与损失函数定义

通过将各种任务制定为文本到文本问题来训练 INSTRUCTOR，该问题在给定输入 x 的情况下区分 (好/坏) 候选输出 $y \in \{y^+, y_i^-\}$，其中训练样本对应于元组 (x, I_x, y, I_y)，其中 I_x 和 I_y 分别是与 x 和 y 相关的指令。例如，在检索任务中，x 是一个查询，(好/坏) y 是来自某个文档集合的 (相关/不相关) 文档。对于文本相似性任务，输入和输出具有相似的形式，并且通常来自相同的源集合。对于分类任务，可以通过选择 y 作为与相同类别和不同类别相关的文本序列来形成训练样本，以表示好与坏的示例（有关对构建的详细信息请参见第 3.3 节）。输入和输出指令取决于任务。对于文本相似性等对称任务，输入和输出具有相同的形式和编码目标，指令是相同的。对于检索等非对称任务，输入是单句查询，输出是文档，指令反映了这种差异。

输入 x 的候选 y 的优度由相似度 s(x, y) 给出，即它们的 INSTRUCTOR 嵌入之间的余弦：

$$s\left(x, y\right) = \cos\left(E_I\left(I_x \oplus x\right), E_I\left(I_y \oplus y\right)\right) \tag{1}$$

继 Ni 等人之后 [18]，最大化正对 (x, y^+) 之间的相似性并最小化负对 $\left\{\left(x, y_i^-\right)\right\}_{i=1}^k$，其中 k 表示每个正对的负对数量。具体来说，损失函数如下：

$$L = \frac{e^{s\left(x,y^+\right)/\gamma}}{\sum_{y\in B} e^{s(x,y)/\gamma}} \tag{2}$$

其中 γ 是 softmax 温度，B 是 (x, y^+) 和 $\left\{\left(x, y_i^-\right)\right\}_{i=1}^k$ 的并集。进一步关注 Ni 等人的工作 [18]。我们计算交换 x 和 y 的相同损失，并将其添加到之前的损失中（即双向批量采样损失）。

3.3 MEDI：带有指令的多任务嵌入数据

现有的数据集不包含用于嵌入训练和指令的各种任务。因此，INSTRUCTOR [24] 构建了一个包含 330 个数据集的集合，其中包含跨不同任务类别和领域的指令：带指令的多任务嵌入数据 (MEDI)。

3.3.1 数据构成

通过将来自 SuperNaturalInstructions [18] 的 300 个数据集与来自为嵌入训练设计的现有集合的 30 个数据集相结合来构建 MEDI。

super-NI 数据集带有自然语言指令，但不提供正负对。通过使用 Sentence-T5 嵌入构建这些对 [18]，用 E( ) 表示。对于分类数据集，通过根据输入文本嵌入 $\cos\left(E\left(x_i\right), E\left(x_j\right)\right)$ 计算示例之间的成对余弦相似度。如果两个示例具有相同的类标签 $\left(y_j^+ = y_i\right)$，则使用与 x_j 高度相似的示例 x_i 创建正对，如果标签不同 $\left(y_j^- \neq y_i\right)$，则创建负对。对于输出标签是文本序列的其余任务，首先计算以下分数：

$$s_{pos} = \cos\left(E\left(x_i\right), E\left(x_j\right)\right) + \cos\left(E\left(y_i\right), E\left(y_j\right)\right) \tag{3}$$

$$s_{neg} = \cos\left(E\left(x_i\right), E\left(x_j\right)\right) - \cos\left(E\left(y_i\right), E\left(y_j\right)\right) \tag{4}$$

我们选择具有最高 s_{pos} 的示例对作为正对，选择具有最高 s_{neg} 的示例对作为硬负对。我们在训练中使用一个硬负例和批量采样负例。根据分析表明 [24]，由于不同的任务定义，来自 super-NI 的训练数据特别提高了评估中的指令鲁棒性。

其他 30 个嵌入训练数据集来自 Sentence Transformers 嵌入数据，KILT [20] 和 MedMCQA [19]。这 30 个数据集已经包含正对；其中一些，例如 MSMARCO [3] 和 Natural Questions [13]，也包含硬负对。继 Ni 等人之后 [18]，INSTRUCTOR [24] 在模型微调过程中使用四个负对（硬负数或批量负数）。由于所有这些数据集都没有指令，因此他们开发了一个统一的指令模板，并为每个数据集手动编写特定的提示，如下所述。

3.3.2 指令注释

MEDI 中的每个训练实例都是一个元组 (x, I_x, y, I_y)，其中自然语言指令 I_x 和 I_y 描述了 x 和 y 的嵌入如何用于任务。例如，在开放域 QA 中（例如表 1 中的自然问题），I_x 是“代表用于检索支持文档的维基百科问题；输入：,”，I_x 为“代表用于检索的维基百科文档；输入：。”

为了使 MEDI 中所有数据集的指令保持一致，他们 [24] 设计了统一的指令格式，由以下部分组成

- **文本类型**指定我们使用嵌入模型编码的输入文本的类型。例如，对于开放域 QA 任务，查询的输入类型是问题，而目标的输入类型是文档。
- **目标（可选）**描述如何在任务中使用输入文本的目标。例如，对于分类任务，任务目标是将句子分类到某个类别，而检索的任务目标是检索相关文档。因为并非所有句子都与特定任务相关（例如，STS 目标通用编码），所以这部分设为可选。
- **领域（可选）**描述任务域。例如，对于 NewsIR，任务的领域是新闻。因为并非所有任务都指定域（例如，STS 处理一般语句），所以这部分也是可选的。

最终指令采用以下格式："**代表任务目标的（域）文本类型:**"

4 复现细节

4.1 与已有开源代码对比

本文参考代码如下网址可见 https://github.com/xlang-ai/instructor-embedding

4.1.1 对 instruction 作用探讨

在 INSTRUCTOR [24] 的文章的分析消融实验中探讨了关于分析了 instruction 对训练数据多样性的重要性；测试了 INSTRUCTOR 对于 instruction 叙述变体的鲁棒性；探讨了 instruction 复杂程度与模型性能的关系等等要素。但是本文认为没有解释指导语是如何起作用的，是否是因为某一任务使用相同的指导语，而因为相同的指导语而拥有更多的相同 token 而导致 text embedding 的结果更加接近?

因此在本文中，我们将代码中的对于不同任务的指导语进行修改，使用大串相同字符进行指导，不同的任务指导语为不同的相同字符。

图 3. figure title

4.2 实验环境搭建

本次实验所需环境如下:

python=3.7

transformers==4.20.0

datasets>=2.2.0

pyarrow==8.0.0

jsonlines

numpy
requests>=2.26.0
scikit_learn>=1.0.2
scipy
sentence_transformers>=2.2.0
torch
tqdm
rich

5 实验结果分析

INSTRUCTOR 的结果以及三个基准的基线：MTEB、Billboard 和提示检索如图 4。我们对相同尺寸的 INSTRUCTOR 和 GTR 模型进行了正面对比。我们还包括其他代表性模型的性能以供参考，但并不用于直接比较。INSTRUCTOR 在所有三个基准测试中平均达到最佳性能。与初始化 INSTRUCTOR 的 GTR-Large（335M）相比，指令微调在 MTEB、Billboard 和提示检索方面分别提高了 5.7%、18.3% 和 5.7% 的性能。具体来说，在所有任务类别中，INSTRUCTOR (335M) 在文本评估 (18.3%)、分类 (10.1%) 和聚类任务 (8.9%) 方面比 GTR-Large 表现出了巨大的改进。特别值得注意的是，与之前最先进的模型 Sent-T5-XXL 相比，INSTRUCTOR 的性能（平均 58.4 vs. 56.5），尽管 INSTRUCTOR 的参数少了一个数量级（335M vs. 4.8B）。

Benchmark	**MTEB**								**Billboard**	**Prompt**	**Avg.**
Task category	Retri.	Rerank	Cluster	Pair.	Class.	STS	Sum.	Avg.	Text Eval.	Retri.	
# datasets	15	4	11	3	12	10	1	56	3	11	70
Small Models for reference (<500M)											
SimCSE (110M)	21.9	47.5	33.4	73.7	67.3	79.1	23.3	48.7	29.4	58.3	48.2
coCondenser (110M)	33.0	51.8	37.6	81.7	64.7	76.5	29.5	52.4	31.5	59.6	51.8
Contriever (110M)	41.9	53.1	41.1	82.5	66.7	76.5	30.4	56.0	29.0	57.3	53.2
GTR-Large (335M)	47.4	55.4	41.6	85.3	67.1	78.2	29.5	58.3	31.2	59.8	55.1
INSTRUCTOR (335M)	**47.6**	**57.5**	**45.3**	**85.9**	**73.9**	**83.2**	**31.8**	**61.6**	**36.9**	**63.2**	**58.4**
Relative gain (%)	+0.4	+4.5	+8.9	+0.7	+10.1	+6.4	+7.8	+5.7	+18.3	+5.7	+5.9
Large Models for reference(≥500M)											
Sent-T5-XXL (4.8B)	42.2	56.4	43.7	85.1	**73.4**	82.6	30.1	59.5	33.9	61.5	56.5
GTR-XXL (4.8B)	48.1	56.7	42.4	86.1	67.4	78.4	30.6	58.9	32.0	60.8	55.8
SGPT-NLI (5.8B)	32.3	52.3	37.0	77.0	70.1	80.5	30.4	53.7	29.6	57.9	51.9
GTR-XL (1.5B)	48.0	56.0	41.5	86.1	67.1	77.8	30.2	58.4	32.0	60.4	55.5
INSTRUCTOR-XL (1.5B)	**49.3**	**57.3**	**44.7**	**86.6**	73.2	**83.1**	**32.0**	**61.8**	**34.1**	**68.6**	**58.8**
Relative gain (%)	+2.7	+2.3	+7.7	+0.6	+9.1	+6.9	+6.0	+5.8	+6.6	+13.6	+5.9

图 4. 原文实验结果

在本文中仅仅测试评估了 Billboard 中的文本评估的嵌入结果，测试结果与文章相同，在修改后的测试中，反而质量出现了下降。

6 总结与展望

尽管 INSTRUCTOR 显着提高了基线 GTR 性能，但由于计算限制，我们在模型微调过程中只能使用四个反例。然而，反面例子已被证明在对比学习中发挥着重要作用 [22]。未来的

工作能够扩大微调过程中使用的反例数量，并研究挖掘 hard negative 的各种方法。此外，将多任务指令微调应用于 GTR-XXL（4.8B 参数），这也是未来探索的领域。INSTRUCTOR 的核心是指令设计。虽然 INSTRUCTOR [24] 目前的统一的格式已证明有效，但并不一定是最优解在未来的研究可以探索其他指导要素以进一步提高绩效。

参考文献

[1] Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. SemEval-2012 task 6: A pilot on semantic textual similarity. In Eneko Agirre, Johan Bos, Mona Diab, Suresh Manandhar, Yuval Marton, and Deniz Yuret, editors, **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics.

[2] Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen tau Yih. Task-aware retrieval with instructions, 2022.

[3] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension dataset, 2018.

[4] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, 2017.

[5] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data, 2018.

[6] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings, 2022.

[7] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks, 2020.

[8] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022.

[9] 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.

[10] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering, 2021.

[11] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering, 2020.

[12] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors, 2015.

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

[14] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3?, 2021.

[15] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context, 2022.

[16] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*, 2021.

[17] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark, 2023.

[18] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers, 2021.

[19] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann, editors, *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR, 07–08 Apr 2022.

[20] 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. Kilt: a benchmark for knowledge intensive language tasks, 2021.

[21] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[22] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021.

[23] Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Selective annotation makes language models better few-shot learners, 2022.

[24] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. 2022.

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

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

[27] Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections, 2021.