1. 引言

招聘场景中的文本（如职位描述或简历）区别于其他类型文本，不需要从头到尾阅读，通常采用扫视阅读，因此在模型架构设计时，应更关心局部关系和细粒度特征的捕捉。

在过去，以SentenceBERT[1] 为代表的Embedding模型，通过引入一些目标函数对BERT[2]类模型生成的句子 Embedding 进一步训练，从而使其在语义上有意义。更大的贡献在于，通过 Siamese Encoder 的方式，将每个句子映射到同一个语义空间，且使用余弦相似度度量句子的语义相似性，从而大幅降低了计算开销，使得在具有大规模数据的招聘场景中，实现比较精准的人岗匹配成为可能。因此，多数表征学习模型在当前时代都是基于 BERT 和 RoBERTa[3]，它们将多个 Encoder 堆叠构成仅编码器模型，在数十亿个单词上进行预训练[4]。

随着 ChatGPT 的发布，LLMs 的研究开始了突破性的发展，如何从从解码器中提取语义上有意义的嵌入并将其应用于语义匹配场景[5]，逐渐受到了研究者们的关注。首次进行这一尝试的是 OpenAI[6]，他们基于一个预训练的 Transformer 模型，去除了 Encoder 部分和最后的词表映射头，仅保留 Decoder 模型结构和参数，仅执行单次前向传播进行训练。它们证明了，基于预训练的仅解码器模型进行大批次对比学习和大规模训练的简单方法，就能够从 LLM-based Embedding 模型中得到有意义的句子嵌入，且一定程度上验证了 scaling law，虽然这种趋势并不如 LLMs 中那么明显[7]。

在这种范式之下，逐渐衍生出了两种技术路线：以 LLM2Vec[8]为代表的双向注意力模型和以 Qwen3-Embedding 系列[9]为代表的纯因果注意力模型。前者认为，带有注意力掩码的因果模型限制了它们生成丰富上下文表征的能力，因此他们基于预训练的 LLM 之后的训练工作都是基于双向注意力。后者则保持生成与嵌入的一致性，在 Embedding 训练中依然保持因果注意力，且取 last token Embedding 为句子表征。本文暂不讨论两种技术路线，我们仅基于后者的 Qwen3-Embedding-8B 模型展开了领域训练的实验探索与理论分析，通过一系列的对比与消融实验，我们证明了：基于 Decoder-Only 的 LLM 进行再训练和进一步领域微调的范式，在招聘场景的人岗匹配中是有效的。

2. 方法

领域训练可以看作是迁移学习的一种应用，尽可能保留通用模型的能力、且将其在领域中做适应性训练，以进一步提升其领域性能。这种方法往往能够实现少量领域数据的微调训练即可大幅提升领域性能。

LoRA 微调

测试集为我们构造的三种任务：岗岗匹配（JD2JD）、人岗匹配（JD2CV）、人人匹配（CV2CV），用于考察模型在招聘领域的表现。

3. 实验与结果

在 CRE 的上一个版本（CRE0.5.0）中，我们采用的是 BERT-large 模型结构的模型，通过 CNN 投影层，结合领域数据对模型的最后一层参数进行领域微调，并证明了 CNN 投影层能够提升模型对细粒度特征的感知能力，从而优化了人岗匹配的效果。

3.1. 末层微调

我们尝试沿用 CRE0.5.0 的微调思路，对 Qwen3-Embedding-8B 模型的末层进行微调。

在领域数据中，每个查询有 7 个硬负例。我们采用批内负样本的对比学习，为了更好地利用硬负样本，我们采用了跨 GPU 批次的 InfoNCE 损失，温度系数（temperature）设置为 0.1，将批内其他样本的正例与负例均视为当前查询的负例，全局 batch 设为 32，学习率设为 1e-5，梯度累计为 1，使用 4 个 GPU 并行训练 5 个 epoch，训练中的句子 Embedding 为 last token Embedding（与 Qwen3-Embedding-8B 保持一致）。

Epoch	JD2JD	JD2CV	CV2CV	AVG
base	**60.98**	**62.77**	**52.14**	**58.63**
1	39.71	50.79	43.42	44.64
2	41.38	49.58	41.55	44.17
3	44.14	41.95	35.05	40.38
4	43.31	43.45	29.95	38.90
5	41.27	35.09	30.12	35.49

表 1. 最后一层微调的实验结果

训练后，对模型做领域性能评估，微调最后一层的结果如表 1 所示（微调最后两层、三层的结果均类似，此处不再展示），表中的数值指余弦相似度的斯皮尔曼秩相关系数。从结果中可以发现，从微调开始起，模型的领域性能即大幅下降，这表明训练过程与模型严重不适配，因此，我们分析了 Encoder-Only 模型和以 Qwen3-Embedding-8B 为代表的 LLM-based Embedding 模型的特性。

Encoder-Only 模型各个层的功能相对较为独立（如浅层学习词法，中层学习句法，高层学习语义和全局的整合），这为 Encoder-Only 模型的领域微调仅训练末层或投影层即可有效提供了一定的理论支持。

而 Qwen3 LLM 技术报告[10]指出，Qwen3-8B-base（即 Qwen3-Embedding-8B 模型的基座模型）为纯预训练模型，包括：一阶段世界知识预训练、二阶段使用合成的高质量推理数据继续预训练和三阶段高质量长上下文语料扩展预训练，这三阶段的预训练都是下一个 token 预测任务，因此训练出的 Qwen3-8B-base 模型参数具有极强的自回归特性，即高层对低层的依赖性非常高，自回归特性意味着：高层输出必须基于低层特征的精确传递。

虽然 Qwen3-Embedding 模型基于 base 模型又做了弱监督预训练和监督微调的对比学习训练，使 last token Embedding 作为输入序列的表征。但我们认为，此时 last token Embedding 的表现，本质上是两种特性的动态权衡：（1）自回归目标的生成导向的上下文压缩能力；（2）对比学习对表征空间的显式优化。这就解释了表 1 领域微调的结果不佳的原因——因为对比学习微调最后一层破坏了这种平衡。

3.2. LoRA 微调

基于上述分析，我们认为，进一步的领域微调须兼顾模型参数的两种特性，才能维持这种平衡，并提高模型在领域的表现。LoRA 微调作为一种轻量化参数微调的方法，相当于让模型的所有层都参与了反向传播参数更新，同时又极大地减少了可训练参数数量[11]，使微调成本大幅降低。

我们引入 LoRA 微调的方法，借助 swift 框架，将领域数据修改为合适的格式之后，进行 Qwen3-Embedding-8B 模型的微调。低秩矩阵的秩设为 8，采用跨 GPU 批次的 InfoNCE 损失，温度系数（temperature）设置为 0.1。考虑到 8B 模型的显存占用，全局 batch 为 16，梯度累计设为 4，学习率为 6e-6，使用 DeepSpeed Zero3 进行内存优化，在 4 个 GPU 上并行

训练，训练中的句子 Embedding 为 last token Embedding。

Checkpoint	JD2JD	JD2CV	CV2CV	AVG
base	60.98	**62.77**	52.14	58.63
1	**64.12**	59.05	**53.95**	**59.04**
2	63.14	57.23	51.51	57.29
3	64.02	54.88	47.77	55.56
4	63.38	52.31	47.03	54.24

表 2. LoRA 微调的实验结果

我们记录了一些训练中的 Checkpoint（代号 1、2、3、4），评估结果如表 2 所示。结果表明，虽然 JD2JD 任务有轻微提升，但其他两个任务、以及整体性能，都呈现一定的下降趋势，尤其是 JD2CV 任务。这表明，模型在微调训练中并没有很好地适应和学习领域数据的特点，因此训练并未起到实际效果，仅仅是一种“泛化”，而这种泛化揭示了我们的任务特点：相较于通用模型的能力而言，JD2CV 是一种“较难”的任务，其次是 CV2CV，而 JD2JD 相对较为简单。这与实际场景也一一对应，人岗匹配倾向于是一种“跨域匹配”任务，需要捕捉不同类型文本（岗位描述/简历）中的技能词的上下位匹配关系，这并非简单的关键词匹配，甚至需要一定的知识关联推理能力；人人匹配则倾向于“同质匹配”但相对更复杂，因为即使是同一种技能，不同的人的描述方式也会不同；而岗岗匹配几乎是“同质匹配”，其术语的一致性和规范性高，因此模型的泛化结果也可以。

表 2 的实验结果揭示了：当前的微调训练方式并没有达到预期的结果。回顾 Qwen3 Embedding 的对比学习训练过程，“高质量”的合成对数据，基于适合查询的文档候选、传入 Qwen3-32B LLM 中，结合多种维度的提示语提高了数据集的质量。这就意味着，它的训练数据相对是规范的，而且一定程度上借助了 LLM 的能力在数据层面“对齐”了文本。而我们的领域微调数据与之相比，文本内容和格式差异大、规范性相对低，且没有做这种“对齐”，导致我们的领域微调破坏了模型原来的“模式”，因此无法正确地将模型的能力“迁移”到领域内，从而造成了 LoRA 微调性能不合预期。

3.3. 合成数据集

通过以上分析，我们明确了：基座自身的训练和领域训练的不一致，以及数据质量差异，是影响领域微调的重要原因。因此，我们参考 Qwen3 Embedding 的合成数据过程，对我们的领域数据质量做一定的增强。具体来说，我们将原领域数据集的每个查询提供给 GLM-4-Flash LLM，通过多维度的提示（如行业背景、核心技能要求、技能迁移性等）指导模型输出更高质量的、偏结构化的增强查询。最后，我们使用新的查询与原始文档作为文本对，同 3.2 节的所有配置进行领域微调（在下文中，我们将 Checkpoint 4 的模型称为 CRE 1.1，将 Qwen3-Embedding-8B 称为 CRE 1.1-base）。

Checkpoint	JD2JD	JD2CV	CV2CV	AVG
base	60.98	62.77	52.14	58.63
1	64.84	60.82	56.87	60.84
2	65.19	61.29	57.24	61.24
3	65.29	61.99	56.99	61.42
4	**65.33**	**63.01**	**58.20**	**62.18**

表 3. 使用合成数据集 LoRA 微调的实验结果

我们同样记录了 4 个 Checkpoint，如表 3 所示。可以发现，此次领域微调使得模型在三个任务上的性能都得到了提升，而与 3.2 节实验的唯一的区别仅仅是训练数据。我们认为，合成的数据一定程度上缩小了跨域文本的差异，使得领域微调前后的训练一致。在领域微调训练中，模型并非完全在泛化，而是保持此前的能力，进行了一定程度的领域知识/模式学习，因此取得了预期的领域性能提高。此外，我们证明了：**训练时使用增强查询、而测试时使用原始查询，模型性能依然会提高**。我们认为，采用增强查询进行的对比学习训练，会在语义空间中拉近增强查询与相关文档的距离，而原始查询作为增强查询的基础，自然地就也得到了优化。这对实际的检索场景很有意义，意味着文档库无需频繁更新，具有很强的应用价值。

3.4. 指令测试

作为基于预训练的 LLM 进一步训练的 Embedding 模型，Qwen3-Embedding 的开发者建议用户根据具体场景、任务和语言来定制指令，这一点与只支持固定指令前缀、没有指令泛化能力的 Encoder-Only 模型有很大区别。因此我们精心设计了指令，并做了使用指令前后的对比实验，评估结果如表 4 所示，可见，领域微调训练保持了 Qwen3-Embedding-8B 模型的指令遵循能力，且在查询端使用指令达到了 SOTA 表现。此外，注意到 CRE 1.1-base 在使用了指令后，JD2CV 指标明显下降，CV2CV 微微下降，JD2JD 有所提升。因为 CRE 1.1-base 从未见过领域数据，所以该测试就是纯粹的泛化能力测试，泛化结果表明仅 JD2JD 性能可提高，这进一步支持了我们在 3.2 节中的分析，即人岗匹配是可能需要知识关联推理的、更难的“跨域”任务，人人匹配是较为复杂的“同质匹配”，而岗岗匹配是较为简单的、偏向于关键词匹配的“同质匹配”。

Model	JD2JD	JD2CV	CV2CV	AVG
CRE 1.1-base w/o instruct	60.98	62.77	52.14	58.63
CRE 1.1-base w/ instruct	65.35	59.42	52.11	58.96
CRE 1.1 w/o instruct	65.33	63.01	58.20	62.18
CRE 1.1 w/ instruct	**66.14**	**64.73**	**62.44**	**64.44**

表 4. CRE 1.1 是否使用指令的对比

3.5. 领域泛化测试

为验证模型的领域泛化性，我们还进行了跨领域的对比实验，结果如表 5 所示。实验结果表明，CRE 1.1 在金融和算法等不同领域同样具有良好的效果。

Model	Algorithm Domain	Finance Domain
BGE-large-zh-v1.5	34.05	58.18
CRE 0.4.1	42.88	63.70
Conan-embedding-v1	43.37	54.69
CRE 0.5.0	45.44	64.14
CRE 1.1-base w/ instruction	58.96	66.25
CRE 1.1 w/ instruction	**64.44**	**69.29**

表 5. 领域泛化测试结果

3.6. 消融实验

根据以上的实验和分析，我们可以明确：基于合成的训练数据进行领域微调是有效果的。为了进一步确定是合成数据本身有效，还是合成数据更适配于 Qwen3 Embedding-8B 这种 LLM-based Embedding 模型，我们做了消融实验。分别使用原始数据和合成数据，以完全一致的训练参数配置对 Encoder-Only 结构的模型（以 BGE-large-zh-v1.5 为例）进行领域微调，我们取第 3 个 epoch 后的 checkpoint 进行对比，实验结果如表 6 所示，使用合成数据训练的 BGE 甚至不如原始数据。可见，并非合成数据本身有效，而是因为 Qwen3 Embedding-8B 的模型结构更能够利用这种形式的数据。

Model	JD2JD	JD2CV	CV2CV	AVG
Origin-data	**44.34**	**28.27**	**41.74**	**38.12**
Synthetic Data	43.83	28.17	41.70	37.90

表 6. CRE 1.1 是否使用指令的对比

4. 讨论与相关工作

4.1. 合成数据的作用

在本节中，我们对消融实验中发现的问题展开进一步的分析。初步可以得到以下三个观点：

(1) 采用合成数据微调并非对所有模型都有效。

(2) 采用合成数据对 Qwen3 Embedding-8B 模型进行领域微调，有利于保持微调训练的一致性，而 BGE 自身的微调期间却未见过这样的数据，因此反而会使训练效果变差。

(3) 合成数据可能与 Qwen3 Embedding-8B 的模型结构更适配，即 Qwen3 Embedding-8B 更擅长利用这种样式的数据，我们认为，模型性能的提升主要源于其中的 GLU 模块从合成数据中受益。具体而言：

GLU 结构比 FFN 更适合于人岗匹配场景。因为 GLU 结构通过独立的权重分支实现 Token 内部维度的、更精细化的特征重要性控制，从而增强对局部关键特征的凸显与对冗余特征的抑制能力。合成数据提高了数据的规范性，使查询文本结构化，其带来的影响有：

(1) **帮助模型发现关键特征**，使模型更聚焦于学习如何精准控制这些已识别特征的表示。

(2) **有益于模型学习不同特征的重要性差异**。当使用如“职位”、“行业背景“等显式的标签时，相当于给文本添加了“特征容器”或“特征标记”，它清晰地划分了不同特征的类别和具体的特征值。

(3) **促进了同类特征之间的“分离”与“对比”**。促使模型在对比学习中“把握”文本对之间真正的相似与区别。

(4) **帮助模型发现有益的特征组合**。人岗匹配场景中，有很多情况依赖于技能词的组合，比如“项目经验”中的”强电项目经验“，与职位名称中”硬件工程师“的组合，这才是精准匹配的关键，而当数据被结构化为一个个的”标签“与”值“的形式时，注意力机制中的每个注意力头更容易捕捉到完整的标签与值，从而更有利于接下来的组合与交互处理，帮助模型发现有益的特征组合。

4.2. LLM-based Embedding 模型的推理能力

LLM-based Embedding 模型的初次实践来自 OpenAI[6]的 cpt-text 系列模型，该模型仍保

持 LLM 的单向注意力掩码，证明了基于一个预训练 LLM 模型进行大批次对比学习和大规模训练，就能够产生高性能的文本嵌入。虽未明言，但随后的 SGPT[5]的研究中明确了对比学习的训练过程，是“仅通过一次前向传播”得到的。

而 **LLM 的推理能力来自于多次前向传播中新旧 token 的动态交互与自终止机制**。LLM 通过迭代的自回归生成步骤，在因果注意力约束下动态融合输入与历史输出，形成渐进式语义合成；同时基于隐状态的信息熵变化自主决策生成终止，其核心能力源于预训练目标（Next-Token Prediction）对语言结构建模与任务边界识别的隐式学习。

因此，当前的 LLM-based Embedding 模型的对比学习任务与单次前向传播机制，注定了它没有、也不可能有真正的推理能力。

4.3. LLM-based Embedding 模型的价值

虽然当前的 LLM-based Embedding 模型没有推理能力，但并不妨碍它在推理任务中可以具有更好的表现，它的价值体现在：

(1) **更长的上下文长度支持**。RoPE 使得模型支持的上下文长度可无限扩展，只需训练数据足够长即可保证很长的位置依然具有意义，这使得从前受限于上下文长度而不得不截取文本的做法彻底成为了过去式；

(2) **更广的背景知识空间**。更大的参数量意味着它理论上能够具有更大的容量来存储知识；

(3) **指令遵循**。Qwen3 Embedding 系列证明了其具有一定的指令遵循与指令泛化能力；

(4) **合成推理轨迹数据训练的有效性**。当前的 LLM-based Embedding 模型，如 Qwen3 Embedding 系列、Conan-embedding-v2、Seed 1.5 等，在推理任务上也展现出了一定的能力，这证明了在 LLM-based Embedding 模型上进行 LLM 合成的推理轨迹数据的训练，能够提升模型在推理任务上的表现，这也是“生成模型指导对比学习训练”的一种方法。

(5) **可能是通往端到端推理 Embedding 的重要且必要的过渡**。

5. 结论

本文探究了 LLM-based Embedding 模型在招聘语义匹配任务中的领域适配机制。相较于传统 BERT 类 Embedding 模型，LLM-based Embedding 模型展现出了显著的语义表征优势，尤其适应岗位描述与简历间可能存在的异构文本对齐需求。本研究发现或证明了：

(1) **适配训练范式的有效性**：采用 LoRA 轻量微调结合领域合成数据，可提升 LLM-based Embedding 模型在 JD2JD、JD2CV、CV2CV 匹配任务上的性能；

(2) **技术演进的新趋势**：LLM-based Embedding 通过长上下文融合与指令控制，天然支持招聘场景的多粒度语义解析（如技能上下位关系捕捉），避免了传统模型的结构性瓶颈；

(3) **工业部署价值**：在训练阶段使用增强查询构造、测试阶段直接应用原始查询的设定下，模型性能仍保持稳定，验证了方法的鲁棒性与实用性。

本研究证实：以 LoRA+合成数据为核心的领域适配方案，为 LLM-based Embedding 模型在复杂招聘语义匹配场景中的工程落地提供了可靠路径。未来的研究可聚焦于：

(1) GLU 结构在领域特征学习中的激活机制优化；

(2) 双向注意力机制在解码器架构中的理论与应用验证；

(3) 编码器模型投影层迁移策略的解码器适配性研究；

(4) 生成式检索/端到端推理 Embedding 框架的进一步探究。

参考文献

[1] Reimers, N., Gurevych, I., 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. http://arxiv.org/abs/1908.10084.

[2] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. http://arxiv.org/abs/1810.04805.

[3] Liu, Y., Ott, M., Goyal, N., et al, 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. https://doi.org/10.48550/arXiv.1907.11692.

[4] Nie, Z., Feng, Z., Li, M., Zhang, C., et al, 2024. When Text Embedding Meets Large Language Model: A Comprehensive Survey. https://doi.org/10.48550/arXiv.2412.09165.

[5] Muennighoff, N., 2022. SGPT: GPT Sentence Embeddings for Semantic Search. https://doi.org/10.48550/arXiv.2202.08904.

[6] Neelakantan, A., Xu, T., Puri, R., Radford, et al, 2022. Text and Code Embeddings by Contrastive Pre-Training. https://doi.org/10.48550/arXiv.2201.10005.

[7] Brown, T.B., Mann, B., Ryder, N., et al, 2020. Language Models are Few-Shot Learners. https://doi.org/10.48550/arXiv.2005.14165.

[8] BehnamGhader, P., Adlakha, V., Mosbach, et al, 2024. LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders. https://doi.org/10.48550/arXiv.2404.05961.

[9] Zhang, Y., Li, M., Long, D., et al, 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. Technical Report.

[10] Yang, A., Li, A., Yang, B., et al, 2025. Qwen3 Technical Report. https://doi.org/10.48550/arXiv.2505.09388.

[11] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., 2021. LoRA: Low-Rank Adaptation of Large Language Models. https://doi.org/10.48550/arXiv.2106.09685.

[12] Gao, T., Yao, X., Chen, D., 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. https://doi.org/10.48550/arXiv.2104.08821.