北 京 邮 电 大 学

本科毕业设计（论文）中期进展情况检查表

学院	人工智能学院		专业	信息工程	
学生姓名	刘孙炎	学号	2020212733	班级	2020219106
指导教师姓名	李珂	所在单位	人工智能学院	职称	副教授
设计（论文）题目	（中文）文本可控 bash 命令生成算法				
	（英文）Algorithm for Text-Controlled Bash Command Generation				

前已完成任务

在毕业设计的前期阶段，我按照规划有序进行了工作，并在学习、调研和实践中取得了一些进展。具体来说：

1、研究方向确定与学习阶段

在初期，我花费了一段时间来确定研究方向和目标，明确了项目的关键问题，并准备了开题报告。在这个阶段，我也开始系统学习了用于文本生成的可控扩散模型相关知识，包括了解其发展现状、趋势以及主流技术方案。确定了三个主要目标：

(1) 确定选用哪种扩散模型作为本项目的基础进行改进

(2) 如何从生成自然语言到生成 bash 命令

(3) 改进模型的准确度和多样性

这为后续的研究奠定了基础。

2、文献调研与资料收集

随后，我进行了深入的文献调研和资料收集工作，以了解可控扩散模型的不同技术路线和应用研究现状。我利用文献检索工具，获取了关于该领域的最新研究成果，并将其作为参考资料。下面是我阅读的主要论文及其简介：

2020 年 6 月，Jonathan Ho 等人[1]提出了基于普通扩散模型的改进版 DDPM。DDPM 是扩散模型最重要的改进，它通过一个前向过程向图像中添加高斯噪声，再通过反向过程学习噪声分布来拟合图像。这篇论文开启了扩散模型的时代，扩散模型成为了图像生成领域最先进的模型。

2021 年 OpenAI[2]提出 Classifier Guidance，使得扩散模型能够按类生成。作者通过在模型中添加一个分类器通过梯度引导来实现扩散模型条件生成。

2022 年，Li Xiang 等人[3]提出 Diffusion-LM 将扩散模型应用到文本生成领域，并使用分类器引导进行可控文本生成。

同年，Jonathan Ho 等人[4]提出 Classifier-Free Guidance 使用有条件生成与无条件生成相结合的方式可以同时提高图像生成的准确度和多样性。这篇论文并不局限于哪一种模型，只需要训练一个有条件的模型和一个无条件的模型，最后在生成样本的时候将两个结果结合。

2023 年，Gong Shansan 等人[5]提出 Diffuseq，通过修改扩散模型，同时输入条件和目标而只对目标加噪的方式，实现了无分类器的可控文本生成。

3、主要难点与解决方法

3.1 bash 命令处理

第一个难点是怎样处理 bash 命令：由于扩散模型最初设计用来进行图像生成，而图像与文本之间存在本质的不同，图像连续而文本离散。所以我参考 Diffsion-LM 的方法对扩散模型进行改进，在前向过程中向模型第一层添加嵌入层，将离散的文本编码映射成连续的嵌入向量再输入扩散模型。

在反向过程中每次采样之后使用嵌入函数，将词向量映射到最近的词嵌入上。最后利用一个舍入函数，将连续的词嵌入向量映射回文本编码。这样就完成了扩散模型生成图片到生成文本的改造。

下一步是完成生成自然语言到生成 bash 命令。bash 命令相比自然语言，有更严格的格式要求，其形式相对固定。如果生成的命令有一个字符错误，可能这条 bash 命令就无法执行。因此，这对模型的准确性提出了更高的要求。

生成文本到生成 bash 命令需要定制一套特殊分词方法。对于条件语句和 bash 命令分别使用不同的算法去分词，然后再映射成词向量。分词方法主要有几种：

第一种方案是将 bash 命令部分模板化，比如对于命令“将/home 下的所有文件都删除:rm -rf /home/*”，将“-rf”替换为特殊标记 FLAG；将/home/*替换为 REGEX。这样做的优势有可以降低数据集词汇量，便于模型学习。

但是，这种方案需要一个命令识别器能准确识别各个 bash 参数并正确替换。并且，在最后生成命令时，需要依赖后处理过程再将标记还原。所以这种方案只减轻了学习难度，但是在其他地方带来了更多问题。

第二种方案是，将上述的参数不做改变，视为完整的一个词进行映射。这种方案的问题在于，出现生词的情况下模型无法正确生成包含生词的命令。

第三种方案是，将上述的参数分解成单个字符输入模型。这样做的好处在于能解决生词问题。但是，这种方案极大增加了学习难度。

我通过理论分析和实验验证，最终采取方案二为目前的分词方法。他结合了其余两种方法的优点，最后的生成效果最好。

3.2 提高准确度与多样性

第二个难点在于提高生成的准确性与多样性。在学习和调研的基础上，我进行了多种文本扩散模型的对比实验，包括 DiffuSeq[5]、SeqDiffuSeq[6]这些模型。并最终基于 DiffuSeq[5]的工作，借助它的模型，使用 Classifier-Free[4]的方法完成。Classifier-Free[4]这篇论文的工作已经在图像生成领域取得了不错的效果，目前流行的图像生成领域应用 Stable Diffusion 以及 DALL·E 2 都采用这种方案用于实现可控图像生成。

Classifier-Free Diffusion Guidance[4]是一种可控扩散模型，其核心是通过一个隐式分类器来替代显示分类器，而无需直接计算显式分类器及其梯度。与传统的基于分类器的方法不同，这种方案不依赖于任何预先训练的分类器，因此具有更好的可扩展性和通用性。可以在训练模型之后，再通过调整引导系数控制生成样本的准确性和多样性。

这种模型的优势在于其简洁而有效的设计，使其能够在文本生成任务中实现较高的灵活性和性能。通过扩散过程，模型可以根据输入的条件生成符合要求的文本，例如在 bash 命令生成中，可以实现对生成文本的控制，确保生成的文本满足特定的语法和语义要求。

在 Classifier-Free[4]与 DiffuSeq[5]的结合后，模型的各项指标都有所提高。但是还是不能满足要求。Classifier-Free[4]在训练时需要随机丢弃整个条件来训练无条件下的模型。我通过更改训练方法为随机丢弃单个条件的部分 token 来训练模型。在进行这样的更改后，我发现无论是有条件性能还是无条件性能都得到了非常大的提升。

在我的研究中，我对 Classifier-Free Diffusion Guidance 进行了深入的研究和实践，通过训练模型并进行多次改进、实验，验证了其在文本生成任务中的有效性和可行性。经过优化和调整，我成功地将该模型应用于 bash 命令生成任务，并取得了令人满意的生成效果。

4、性能分析与比较

我对用于文本生成的可控扩散模型在文本可控的 bash 命令生成中的性能进行了分析，并与其他类似模型进行了比较分析。这有助于我更好地了解所选模型的优势和不足。

模型	BLEU	ROUGE-L	Nl2CMD-Metric	BertScore	Dist-1	Acc
Improved Classifier-Free	**0.743**	0.883	**0.89**	0.930	**0.990**	**0.683**
Classifier-Free	0.684	**0.893**	0.83	**0.936**	0.985	0.599
Diffuseq	0.653	0.886	0.79	0.933	0.977	0.558

表格 1 优化后的 Classifier-Free 与 Classifier-Free 以及 DiffuSeq 性能对比

BLEU 指标（Bilingual Evaluation Understudy）。BLEU 旨在度量 DiffuSeq 模型生成文本与参考文本之间的相似性，通过考察 n-gram 重叠来评估生成文本的准确性和质量。

ROUGE-L 指标（Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence）。ROUGE-L 通过计算生成文本和参考文本之间最长公共子序列的比例来衡量它们的相似性，从而量化生成文本与参考文本的重叠程度。

Nl2CMD-Metric 指标是在 nl2cmd[7]论文中提出的衡量命令之间相似度的指标。通过两个命令重复和不重复命令的数量来衡量命令之间的相似度。

BertScore 是一种自然语言处理的评估方法，通常用于评估生成文本（例如机器翻译或文本生成）的质量。

Dist-1 通过衡量样本不重合的 n-gram 数量来评价生成样本的多样性。

Acc 通过生成样本的词汇出现在目标结果的比例来衡量生成样本的准确度。

不难发现，优化后的 Classifier-Free 模型在多数指标中都明显强于 Diffuseq 模型，少数指标与其他模型差距很小。具有更高的准确度和更好的多样性。

参考文献

[1] Ho J, Jain A, Abbeel P. Denoising Diffusion Probabilistic Models[A]. Advances in Neural Information Processing Systems[C]. Curran Associates, Inc., 2020, 33: 6840–6851.

[2] Dhariwal P, Nichol A. Diffusion Models Beat GANs on Image Synthesis[A]. Advances in Neural Information Processing Systems[C]. Curran Associates, Inc., 2021, 34: 8780–8794.

[3] Li X, Thickstun J, Gulrajani I, 等. Diffusion-LM Improves Controllable Text Generation[J]. Advances in Neural Information Processing Systems, 2022, 35: 4328–4343.

[4] Ho J, Salimans T. Classifier-Free Diffusion Guidance[A]. 2021.

[5] Gong S, Li M, Feng J, 等. DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models[A]. 2022.

[6] Yuan H, Yuan Z, Tan C, 等. SeqDiffuSeq: Text Diffusion with Encoder-Decoder Transformers[J]. 2023.

[7] Fu Q, Teng Z, Georgaklis M, 等. NL2CMD: An Updated Workflow for Natural Language to Bash Commands Translation[J]. 2023.

是否符合任务书要求进度　　是 □　　否 □

<table>
<tr><td rowspan="2">需完成的任务</td><td colspan="2">在接下来的工作中，我将重点进行以下方面的工作：
1. 完善论文初稿：加快论文的撰写速度，尽快完成初稿，包括对已完成部分的修改和优化，以及对未完成部分的补充和完善。

2. 深入实验和数据分析：进一步完善实验设计，进行更多实验并收集数据，对实验结果进行深入分析，总结结论并与相关文献进行对比和讨论。

3. 论文润色和修改：根据导师和同行的意见反馈，对论文进行进一步的润色和修改，提高论文的质量和学术价值。

4. 最终论文撰写和答辩准备：在完成以上工作后，对最终论文进行整理和排版，并做好答辩准备工作，以确保毕业设计的顺利完成。</td></tr>
<tr><td colspan="2">是否可以按期完成设计（论文） 是 □ 否 □</td></tr>
<tr><td rowspan="2">在问题和解决办法</td><td></td><td>经过长时间的学习、研究和实验，我发现当前方案存在一些问题：
1、 生成命令的可执行性问题：在无条件生成时，模型生成的命令往往无法正确执行，这限制了生成样本的准确性和多样性。
2、 处理生词能力不足：模型在处理含有重要生词的源语句时，其表现较差，难以生成正确的样本。
3、生成速度较慢：模型的采样速度较慢，导致生成一个样本需要的时间过长。
4、实验数据不充足：由于时间和资源限制，我目前只完成了初步的功能实现，没有进行足够充分的实验，因此论文中的实验数据不够充分。</td></tr>
<tr><td></td><td>针对以上问题，我拟采取以下解决方案：
1、模板化的无条件生成：针对生成命令的可执行性问题，考虑采用模板化的方法进行无条件生成，以提高生成命令的准确性和可执行性。
2、改进文本编码解码算法：为了解决处理生词能力不足的问题，计划改进文本编码解码算法，使模型能够更好地处理生词，提高生成样本的正确性。
3、优化推理时的采样算法：针对生成速度较慢的问题，打算改进推理时的采样算法，以提高模型的生成速度，减少生成时间。
4、进行更多实验：为了解决实验数据不充足的问题，计划进行更多的实验，尝试使用不同的数据集进行训练，并测试各种超参数的效果，以收集更充分的实验数据。</td></tr>
<tr><td>指导教师签字</td><td></td><td>日期</td><td>年 月 日</td></tr>
<tr><td>检查小组评分及意见</td><td colspan="3">评分： （总分： ）

组长签字： 年 月 日</td></tr>
</table>