

# P 大树洞生成器设计与实现

匡宇轩 (2100013089)<sup>1</sup>, 陈红韵 (2100013130)<sup>1</sup>, and 王天源  
(1700017703)<sup>2</sup>

<sup>1</sup> 北京大学信息科学技术学院

<sup>2</sup> 北京大学元培学院

2022 年 6 月

## 摘要

本项目为北京大学信息科学技术学院国家精品课程《人工智能引论》的课程项目。本文为该项目的总结报告。

本项目受到北京大学树洞的启发。树洞作为匿名交流论坛，是一个天然的自然语言处理语料库（在遵守相关规定的情况下）。

本项目基于因果语言建模 (CLM) 的原理开发，使用 Python 和 PyTorch 实现。

我们在树洞文本数据集上训练 LSTM 模型，并根据用户的输入，使用训练好的模型产生回复。我们将模型命名为 HoleAI。

**关键词：**自然语言处理；长短期记忆网络；因果语言建模

## 目录

<b>1 背景介绍</b>	<b>4</b>
<b>2 相关工作</b>	<b>4</b>
2.1 Word2Vec . . . . .	4
2.2 LSTM . . . . .	4
<b>3 项目细节</b>	<b>5</b>
3.1 数据集收集与预处理 . . . . .	5
3.2 LSTM 介绍 . . . . .	6
3.3 神经网络构建 . . . . .	7
3.4 神经网络训练 . . . . .	7
<b>4 实验结果与分析</b>	<b>9</b>
4.1 实验结果 . . . . .	9
4.2 结果分析 . . . . .	9
4.3 模型生成 . . . . .	10
4.4 模型部署与可视化 . . . . .	11
<b>5 项目总结</b>	<b>13</b>
5.1 意义 . . . . .	13
5.2 挑战 . . . . .	13
5.3 局限性 . . . . .	13
5.4 收获 . . . . .	13
<b>小组分工</b>	<b>14</b>
<b>致谢</b>	<b>14</b>
<b>参考文献</b>	<b>14</b>
<b>附录</b>	<b>15</b>

# 1 背景介绍

P 大树洞作为校内学生匿名交流平台，是一个完美的语料库（在遵守相关规定的情况下），十分适合 NLP 相关任务。

本项目通过收集大量树洞文本数据，用神经网络进行拟合，希望能创造出一个具有 P 大学生气质的 AI。

使用者可以作为“洞主”，模拟发树洞的过程，但是回复的字母君都是 AI 自动生成。

# 2 相关工作

## 2.1 Word2Vec

该工作提出了两种新颖的模型架构，用于从非常大的数据集中计算单词的连续向量表示。这些表示的质量是在单词相似度任务中测量的，并将结果与以前基于不同类型神经网络的最佳性能技术进行比较。

研究者观察到本方法以更低的计算成本显着提高了准确性，即从 16 亿个单词数据集中学习高质量的单词向量只需不到一天的时间。此外，该工作表明这些向量在测试集上提供了最先进的性能，用于测量句法和语义词的相似性。[1]

## 2.2 LSTM

通过循环反向传播学习在较长的时间间隔内存储信息需要很长时间，主要是因为不足的、衰减的梯度回传。

该工作简要回顾了 Hochreiter (1991) 对这个问题的分析，然后通过引入一种新的、有效的、基于梯度的方法来解决这个问题，称为长短期记忆 (LSTM)。

通过在不造成伤害的情况下截断梯度，LSTM 可以通过在特殊单元内通过恒定误差轮播强制恒定误差流来学习弥合超过 1000 个离散时间步长的最小时间滞后。

与实时循环学习、随时间的反向传播、循环级联相关、Elman 网络和神

经序列分块相比，LSTM 运行更加成功，并且学习速度更快。LSTM 还解决了以前的循环网络算法从未解决过的复杂、人为的长时间滞后任务。[2]

### 3 项目细节

本项目的基本框架为：

- (1). 数据集收集与预处理
- (2). 神经网络构建与训练
- (3). 前端可视化、人机交互

#### 3.1 数据集收集与预处理

鉴于 P 大树洞的相关管理规范，很遗憾我们无法使用爬虫技术获取数据集，因此我们采用了手动收集的方式收集数据。

经过收集，我们获取了一份包含 3000 余条树洞，约一百万字的树洞文本，其中包含了三部分：日常话题、经典话题和神洞（即回复量较多的树洞）。

数据收集完毕后，我们对数据进行了清洗。主要包括：去除洞号、回复数、收藏数等无用信息；去除出现频率过低的字符；去除大量重复的字符。数据集格式如图 1 所示。



图 1: 数据集

接着我们利用 `gensim.models.Word2Vec` 构建字符到索引的双射，从而将每个字符都表示为一个 One-hot vector，作为训练的初始输入。

## 3.2 LSTM 介绍

LSTM，中文为“长短期记忆网络”，是一种循环神经网络，可以学习和预测时间序列。

相较于普通的 RNN (如图 2)，LSTM 的优势在于其可以保持长时间的记忆，并且可以根据监督灵活的选择是否“忘记”，从而可以持久的保存语义信息。这在文本生成的工作中十分重要。

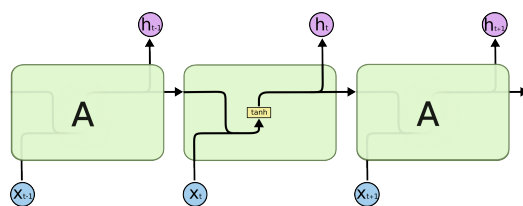


图 2: Vanilla RNN

LSTM 除了可以学习长序列外，还能够进行一次多步预测，对于时间序列预测有一定参考价值。

LSTM 记忆长期记忆的秘密在于门控，每个 LSTM 单元内有三道门，分别为输入门、遗忘门和输出门。

输入门：

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

遗忘门：

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

输出门：

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

其基本单元如图 3 所示。[3]

更进一步，我们将 LSTM 的隐藏层输出再次输入新的 LSTM 单元，便类似全连接层构造出了多层 LSTM (Stacked LSTM)，如图 4 所示。

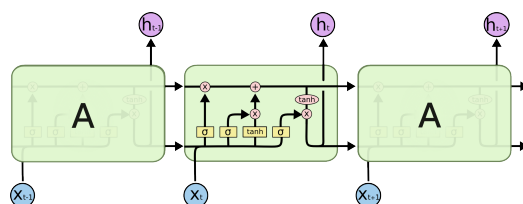


图 3: LSTM 单元

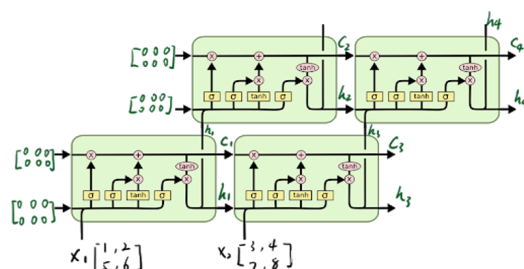


图 4: Stacked LSTM

### 3.3 神经网络构建

我们的输入是批量化的由截断的原始文本映射而来的 One-hot vectors, 其稀疏性不利于训练, 所以我们首先将其通过 Embedding 层, 转为 dense vectors。

接着我们依次将其通过 Stacked LSTM, Layer Normalization, Fully-Connected Layer, 形成与原向量维度一致的输出向量。

接着我们将其通过 Softmax 层形成概率分布, 将其与原始的 One-hot vectors 进行交叉熵损失函数计算, 再利用梯度进行 Backpropagation through time 进行参数更新。

完整的 Pipeline 如图 5 所示。

### 3.4 神经网络训练

我们使用树洞文本语料库对模型从头开始训练。

算力平台为北大人工智能集群系统, 使用 RTX 3080 进行训练。训练时间大致为 4 hours per 50 epochs, 这也与模型的超参数有关。

训练记录截图如图 6 所示。

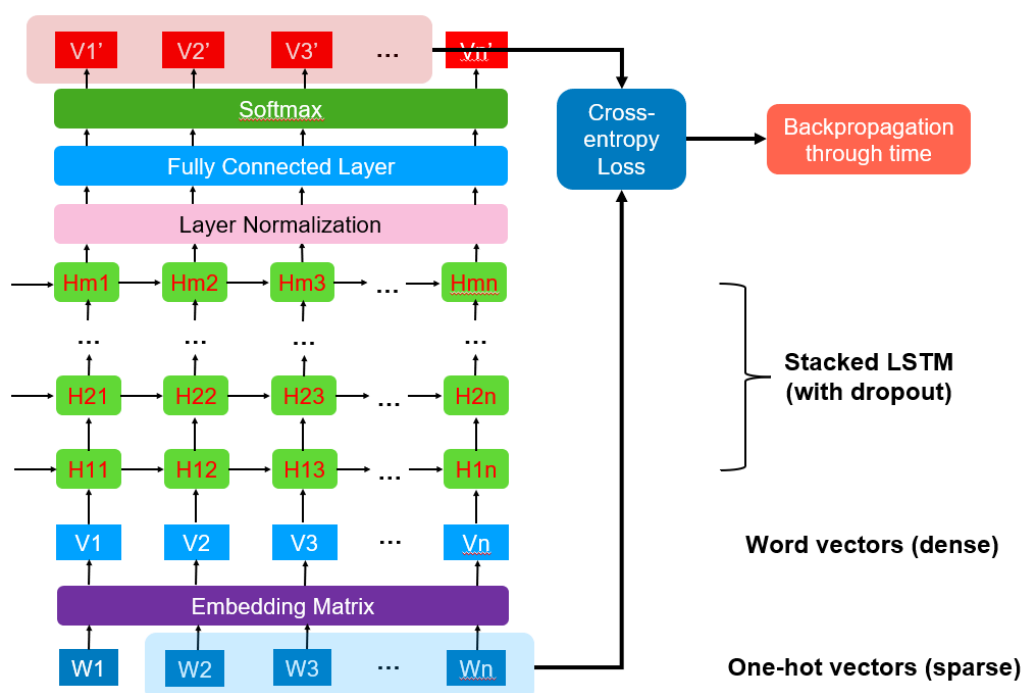


图 5: Whole Pipeline

训练任务		任务模板							
任务名称 搜索		搜索	高级搜索		创建任务		批量删除		
<input type="checkbox"/>	任务名称	算法名称	数据集名称	描述	分布式任务	状态	创建时间	运行时长	操作
<input type="checkbox"/>	50_256_3_50	0505	hole_merge		否	● 成功	2022-05-07 22:20:56	3h54m46s	<a href="#">详情</a> <a href="#">删除</a>
<input type="checkbox"/>	50_128_3_50	0505	hole_merge		否	● 成功	2022-05-07 22:15:25	4h9m13s	<a href="#">详情</a> <a href="#">删除</a>
<input type="checkbox"/>	30_512_4_50	0505	hole_merge		否	● 成功	2022-05-07 16:42:33	3h37m27s	<a href="#">详情</a> <a href="#">删除</a>
<input type="checkbox"/>	30_256_4_50	0505	hole_merge		否	● 成功	2022-05-07 16:13:17	3h30m43s	<a href="#">详情</a> <a href="#">删除</a>

图 6: 训练记录



## 4 实验结果与分析

### 4.1 实验结果

我们使用 tensorboard 绘制了训练过程中 `train loss`, `validation loss` 与 `epoch` 的曲线图，曲线图总览如图 7 所示。

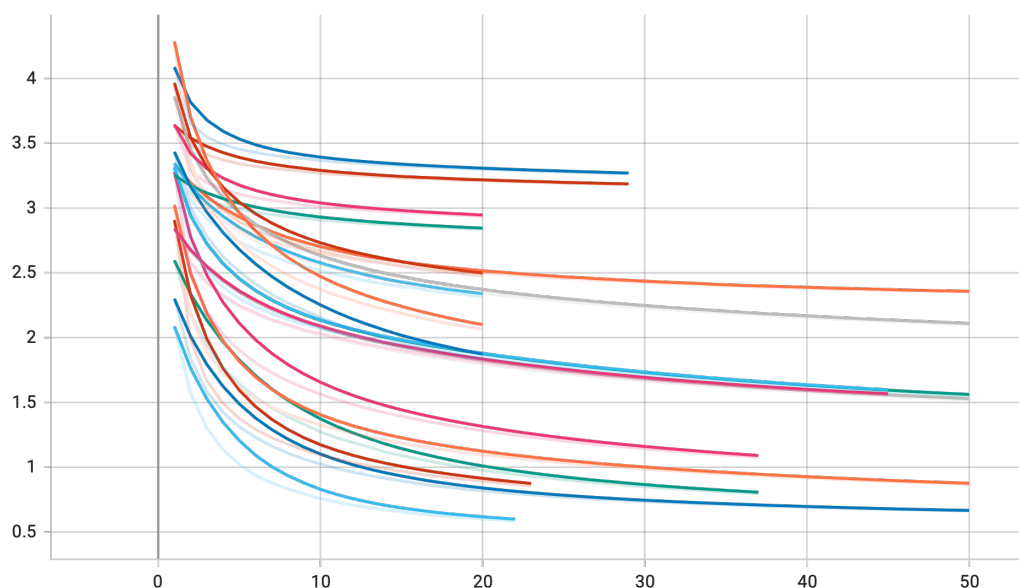


图 7: 训练结果总览

更多实验结果，如训练日志等，详见附录。

### 4.2 结果分析

我们取出其中较有代表性的三对曲线进行分析，如图 8 所示。

我们从曲线中可以得出如下结论：

(1). 训练效果较好，损失函数值单调下降，且收敛到了一个比较低的水平。

(2). 随机分割数据集可以起到一定正则化的作用，这使得训练最后 `validation loss` 低于 `train loss`。

我们在反复训练和调参中也总结出如下结论：

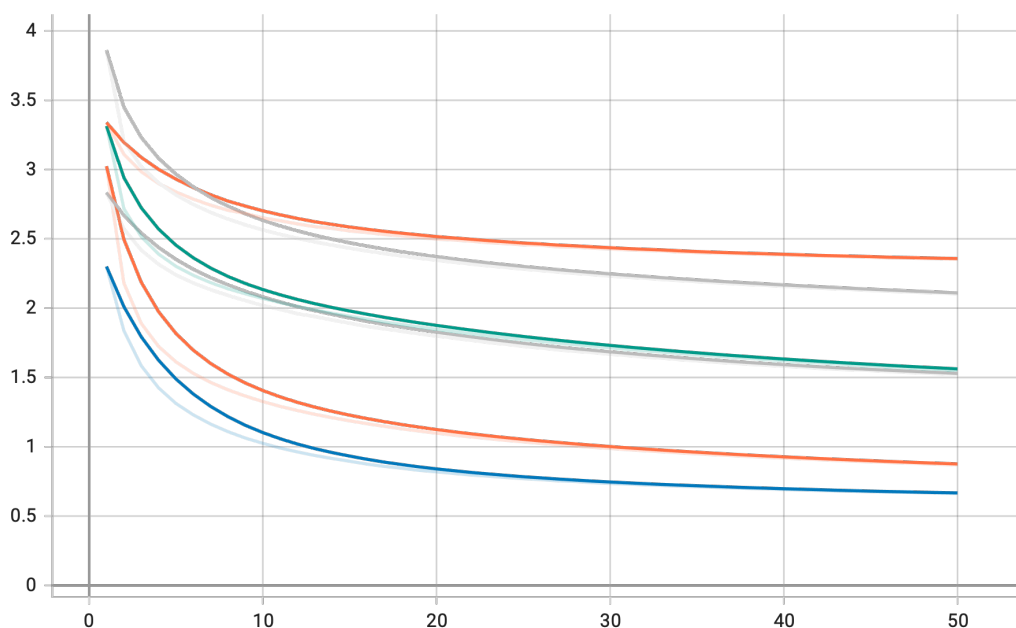


图 8: 分析曲线

- (1). Dropout rate 不宜过大, LSTM 堆叠层数不宜过多。否则会出现梯度消失现象, 不利于参数更新。
- (2). 随机 split 数据集, 可以起到一定正则化的作用, 但也因此丢失了部分语义信息。
- (3). 隐层大小至关重要。隐空间越大, 模型的表现力越强, 对数据的拟合能力越强。

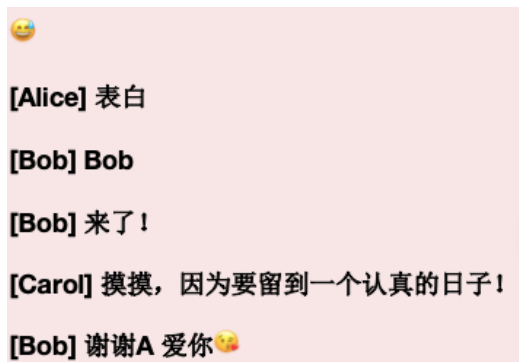
### 4.3 模型生成

我们使用训练好的模型对输入产生响应, 生成回复。生成样例如图 9、图 10 所示。

可以看出, 模型产生的回复捕捉到了文本格式, 例如 [Alice], [Bob] 等人名格式与 Re 洞主: 等回复格式。此外, 模型生成的回复在语言风格上也与真实的树洞文本有很高的相似性。即, “有内味了”。

然而, 模型的生成依旧存在一定问题。如:

- (1). 自说自话: 即前后回复语义相关性较弱, 或完全无关。
- (2). 指鹿为马: 即模型的回复指向性有误。



😊

[Alice] 表白

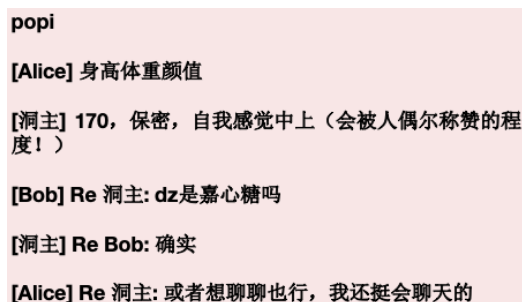
[Bob] Bob

[Bob] 来了!

[Carol] 摸摸，因为要留到一个认真的日子!

[Bob] 谢谢A 爱你🥰

图 9: 生成样例 1



popi

[Alice] 身高体重颜值

[洞主] 170，保密，自我感觉中上（会被人偶尔称赞的程度！）

[Bob] Re 洞主: dz是嘉心糖吗

[洞主] Re Bob: 确实

[Alice] Re 洞主: 或者想聊聊也行，我还挺会聊天的

图 10: 生成样例 2

(3). 虚空索敌：即模型会对未出现的事物进行评论。

(4). 过拟合现象依旧存在：在较长文本的生成中，模型往往会“照抄”原文，产生大段已有文字。这说明模型的泛化能力依然有待提升。

## 4.4 模型部署与可视化

模型训练完成后，我们希望能用更加方便与美观的形式呈现模型的实际效果，也希望以此模拟树洞，故我们依托 GitHub Pages 和 streamlit 框架开发了本项目的前端界面。

用户点击 <https://kryptonite.work/pkuhole/> 链接即可进入本项目的网站，界面如图 11 所示。

在本网站的 Let's Try! 栏目下，是 P 大树洞网页版的图标，用户直接点击或点击此处即可进入人机交互界面进行游玩。用户可以自行调整模型大小、情绪、长度，进行树洞生成。

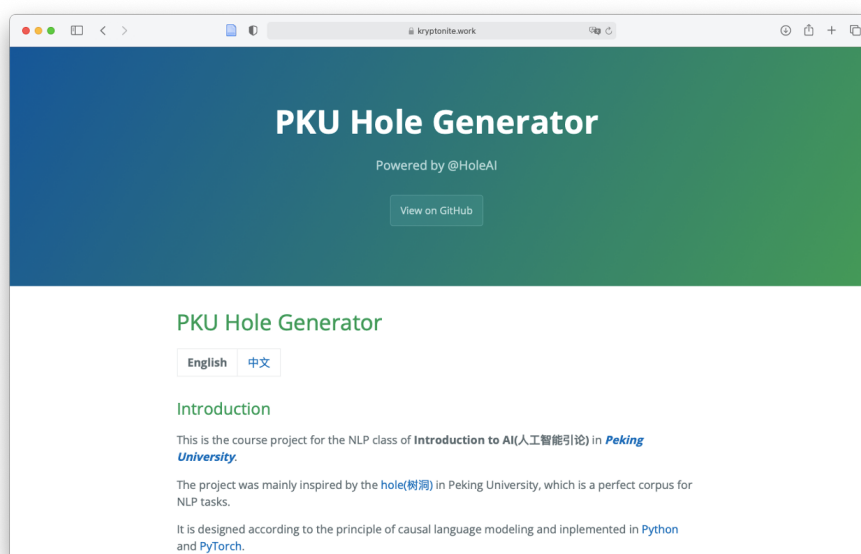


图 11: 网站界面

人机交互界面如图 12 所示。

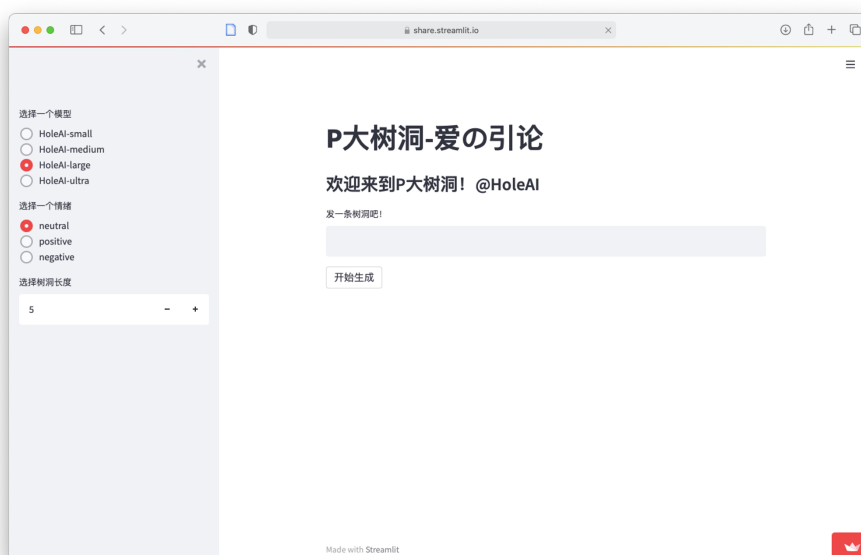


图 12: 人机交互界面

## 5 项目总结

### 5.1 意义

这个任务十分新颖有趣，而且具有一定社会科学方面的研究价值。

通过本项目，我们刻画了一幅北大学子的群像，我们可以通过模型体验千千万万个“我”的日常生活。

### 5.2 挑战

数据集的缺乏（根本没有）和数据集噪声过多，给我们带来了很大的挑战。我们需要花费大量时间进行数据收集与清洗。

模型大小与数据集大小的匹配则是另一个挑战。我们需要在欠拟合 (underfitting) 和过拟合 (overfitting) 之间寻找一种平衡。

### 5.3 局限性

虽然我们的项目制作较为完备，模型效果较好，但依然存在一定局限性：

(1). 数据集较小：由于时间和任务所限，我们无法收集常规情况下巨量的数据，这导致模型出现了不同程度的过拟合现象，如果可以，后续我们也会对数据集的大小和数据分布做出调整。

(2). 模型较为简单：我们采用的模型是经典的 LSTM 因果语言模型，实现难度较低，并没有 fancy 的多头注意力机制等等。但是在本项目较小的数据集下，LSTM 也可以达到较好的效果，故本问题其实也不构成问题。在对数据集大小进行调整之后，我们也可能会尝试更新的技术，甚至采用预训练模块等等。

### 5.4 收获

通过本项目，我们加深了对 Word2Vec、循环神经网络、LSTM 的认识，通过实操深入剖析了机器学习的全过程：数据处理、模型搭建、训练评估、

模型调参、部署实装，更进一步拓宽了我们对于前沿任务的了解，打开了新世界的大门。

此外，在本项目的完成过程中，我们也在不断学习新的技术，如训练过程可视化、模型部署、网页搭建等等。在这一过程中，看 API 文档成为了家常便饭，我们也学习了更多人工智能以外的知识。

## 小组分工

匡宇轩：组长。负责规划项目、模型训练、模型部署、前端制作。

陈红韵：组员。负责数据收集、数据清洗、海报制作。

王天源：组员。负责数据收集、汇报制作、报告写作。

## 致谢

感谢《人工智能引论》大班课老师的知识讲授，NLP 小班课邓志鸿教授的知识讲授与指导，NLP 小班课助教学长学姐的指导与帮助。

也感谢本课程独特的课程设置，使我们学会了许多包括但不限于人工智能领域的知识与技能，提高了我们对于互联网的利用能力以及对于个人电脑的熟练操作能力。在这一过程中，我们受益良多。

## 参考文献

- [1] Efficient Estimation of Word Representations in Vector Space, Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, 2013, <https://arxiv.org/abs/1301.3781/>
- [2] Long short-term memory, S Hochreiter, J Schmidhuber, 1997, <https://pubmed.ncbi.nlm.nih.gov/9377276/>
- [3] Understanding LSTM Networks, colah's blog, 2015, <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

## 附录

1. 本项目的 GitHub 地址: <https://github.com/HirojiFukuyama/pkuhole/>
2. 本项目的 GitHub Pages: <https://kryptonite.work/pkuhole/>
3. 本项目的人机交互网站: <https://share.streamlit.io/hirojifukuyama/pkuhole/app.py/>
4. 训练日志、汇报 PPT: 详见 GitHub 仓库