

人工智能技术与实践 产品设计报告

- 产品名称：深度学习对对联
- 项目成员：
 - 吕传承-2017011441：产品调研
 - 李冠达-2017013615：产品设计和具体工程实现

一、产品立项

春节是中国历史悠久的传统佳节，有着丰富的传统习俗。对联，中国的传统文化之一，又称楹联或对子，是写在纸、布上或刻在竹子、木头、柱子上的对偶语句。对联，对仗工整，平仄协调，是一字一音的中华语言独特的艺术形式。对联是中国传统文化瑰宝。

然而随着社会的发展，越来越多的传统文化正在逐渐遗失，这便是我们的立项初衷——传承对联文化。

产品立项

因为自动对对联文化的目标用户并不是普通大学生群体，因而用户调研比较难进行，因而主要产品立项需要通过市场调研分析开展。目前随着国家经济发展，国家文化产也有了巨大的发展，《中国诗词大会》、《演说家》等节目作为文化输出窗口为我们展示了传统文化在现代依然能展现出来的溢彩流光。

在人工智能突飞猛进地发展的现代，利用人工智能帮助文化产业发展同样是非常值得期待的结合。在诗歌上我们有了九歌人工智能诗词生成器，同样的传统文化下我们希望能够用深度学习来帮助对联文化传承与发展。

产品用研

1. **目标用户**：正如上所讲，文化产业的目标用户包括学生、文化产业研究者等对文化有兴趣进行研究的群体
2. **产品定位**：深度学习的先进技术与对联的传统文化相结合的文创产品
3. **商业模式**：目前文化产品的商业模式较为稚嫩，主流的盈利模式包括广告流推送、文创活动合作等

需求分析

根据以上产品用研情况，产品的主要需求如下；

基本需求：1. 对于已有的上联（或下联），获取对应的下联（或上联）

2. 根据已有的对联中的关键字，获取合适的横批

扩展需求：1. 根据指定的关键字（对子）进行扩展获得对联

2. 根据指定的嵌字以及嵌入字的位置生成对应的嵌字对联

3. 对生成的对联进行装裱，生成对应图片保存

二、产品设计

产品形态

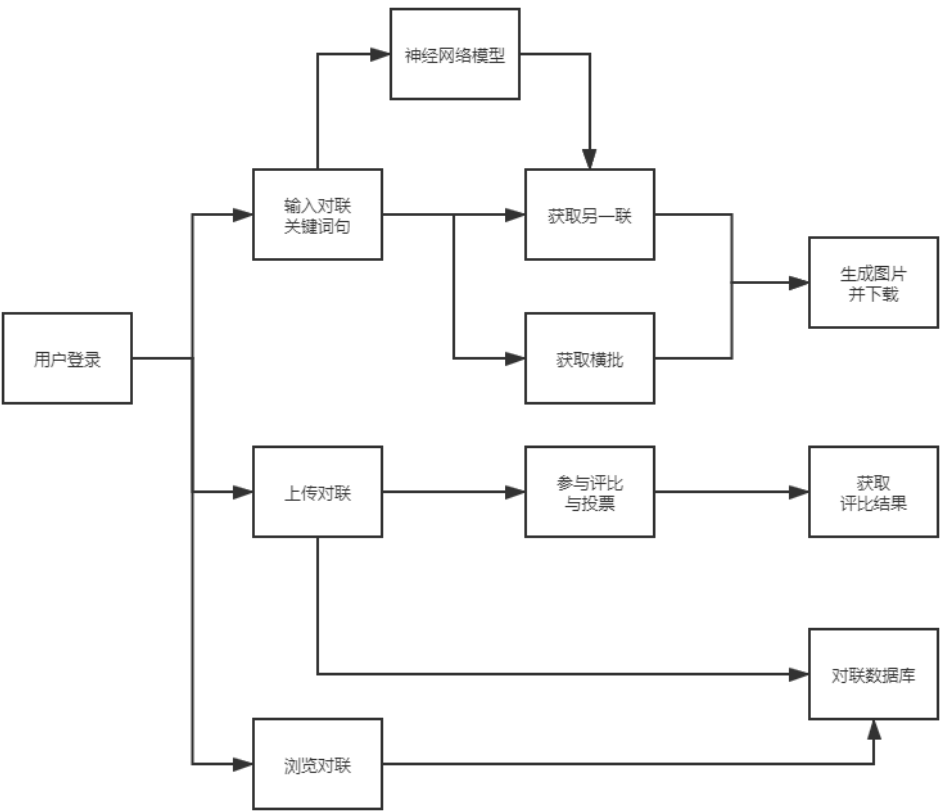
本产品设计形态为Web App，即可以登录网页即可获取的产品服务，其中页面功能主要有以下方面：

- 1. **自动对联服务**：给定已有的上联（或下联），获取对应的下联（或上联）并且获得横批
- 2. **嵌字对联服务**：给出已有的嵌字然后扩展获得对应的对联
- 3. **对联名句服务**：整合目前的经典对联供用户阅读学习
- 4. **对联竞赛服务**：对于给定的上联供用户做对子并与人工智能做出的对联共同竞赛，根据用户投票产生最佳对联

由以上产品形态设计可以看出，本产品基于网页应用整合了众多对联文化的相关活动，利用人工智能对联文化进行推广发扬，实现传统文化的传承与输出。

触发逻辑

本产品触发逻辑如下：



实现方案

根据触发逻辑可以看出本产品的简单的实现方案如下：

- 根据目前已有的语料库进行神经网络训练，获得模型
- 后端部署到服务器，提供训练好的模型与“对对联”过程
- 前端负责发送用户填写的信息和返回的结果

在产品迭代过程中，可以进行扩展的方向包括：扩充语料集，优化现有的神经网络模型，后端逻辑优化以及前端优化等方面。

特殊情况处理

目前针对以下特殊情况进行处理

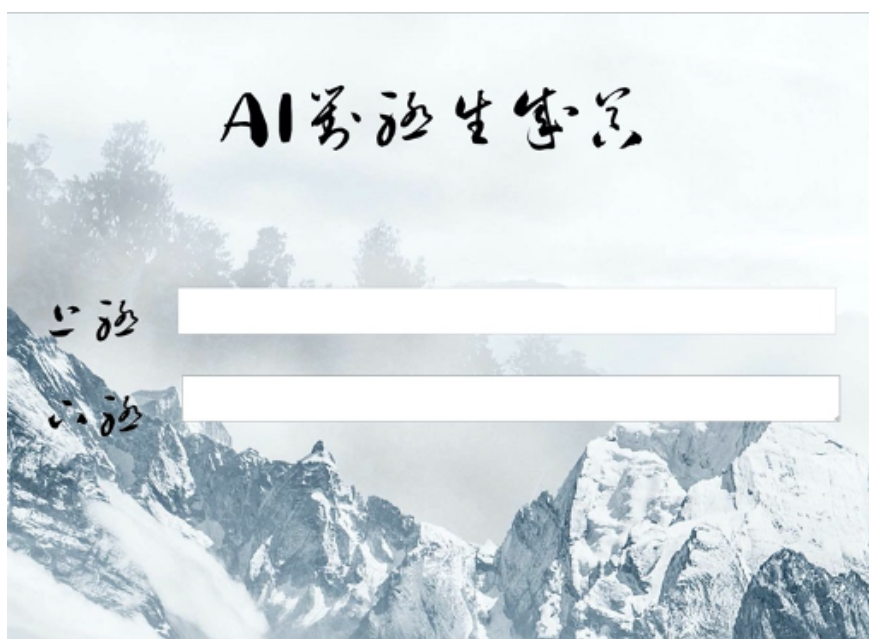
用户输入不合法字符：进行用户输入信息监测，返回不合法字符的错误信息给用户。

三、产品开发

前端实现

由于缺少服务器用于部署及其他种种原因，该项目采用了本地存储并运行 `html` 网页服务器的方式。由于曾经在软工课程中使用 `vue.js` 框架实现网站搭建，因此即使条件不允许使用高级框架，我也对 `html` 和 `css` 的手写工作较为熟悉。

经过网上查阅资料，我选取了一种较为简单的在本地运行网页服务器的方式——利用 `python` 语言的 `flask` 库实现简单的服务器部署，在其内部实现调用神经网络模型、获取输出的功能，并与写好的 `index.html` 实现交互。其具体实现代码见根文件夹下的 `webdemo.py` 文件，及文件夹 `templates` 下的 `index.html`；其中 `index.html` 所需的图片素材位于 `static` 文件夹下。想要运行并打开网站时，只需命令行输入 `python3 webdemo.py` 即可，打开效果如下：



目前此前端有一个现象是用户输入上联并按回车输入后，显示下联的同时上联不再显示。此问题应与数据绑定和渲染有关，但尚未耗费精力解决。此外，为了便于调试和轻量调用，我还实现了一个用命令行进行交互的版本——直接运行 `python3 qa.py` 即可，`q` 退出。运行效果如下：

```
liguanda@MacBook-Pro-16-for-LIGDana Couple_CNN % python3 qa.py
请输入上联：一曲笙歌春似海
AI给出的下联为：千杯灯色月如山
请输入上联：平安夜吃平安果
AI给出的下联为：幸福年开幸福花
请输入上联：苟利国家生死以
AI给出的下联为：不为天子死亡之
请输入上联：q
liguanda@MacBook-Pro-16-for-LIGDana Couple_CNN %
```

后端实现

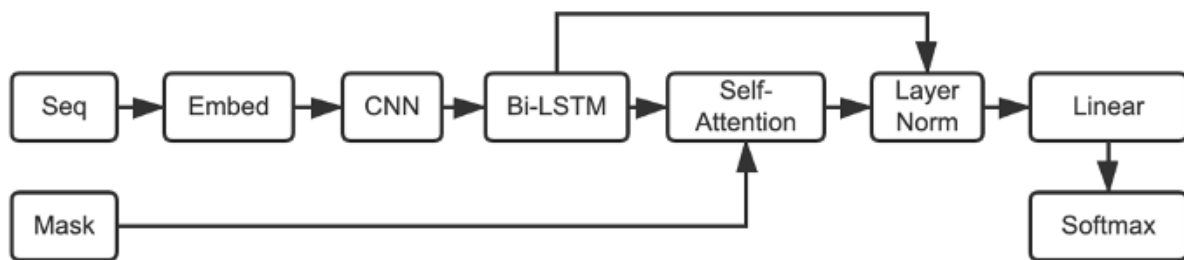
原理

对对联，尤其是现阶段的给出上联对答下联这一过程，实际上序列生成问题。由于对联中上下联字数相等，而且上下联对应的字词相关性很强（如含义、意象、指向性等），联内部前后词语的相关性也很强（如感情色彩，逻辑联系等），因此我们将其转化为序列标注问题——在训练中找出已对应过的字组或词组（这证明该字组或词组的相关性较强，可能是对上下联的优异选择），二者互相标记；在应用时将上联分词，对每个词找出其标注，再串联起一个逻辑正确的句子即可。

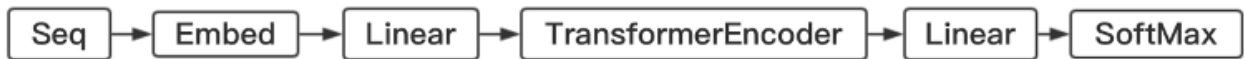
模型选取

本次项目我主要使用了两种较为经典的模型作为效果上的对比：

1. **CNN+ Bi-LSTM + Attention模型**^[1] (工程代码位于根目录下" /AI_CNN "文件夹内)



2. **Transformer模型** (工程代码位于根目录下" /AI_TRAN "文件夹内)



为了能够更好地控制训练差异导致的效果不均，我对两个模型的训练参数均设定为epoch = 20，batch_size = 64。其他技术特点如下：

1. 数据源^[2]为 in.txt 和 out.txt 形式给出，分别一行一条存储上联和下联。原库中共包含77.5万条对联数据，其中77万条作为训练集，4000余条作为验证集。
2. 预处理数据——数据先转化为 Tensor 并缓存至 tensor_dataset 文件夹中，再以 .pkl 文件模式直接喂给网络参与训练。实际训练中，在下载数据集并使其位于根目录 couplet 中，命令行运行 python3 pretrain.py 即可。
3. 可以使用 cuda 库判断当前硬件上可用的安装了 cuda 的GPU数量，并可以根据判断结果选择具体训练方式——CPU / 单个GPU / GPU并行 三种方式进行训练。
4. 由于资源限制，我使用了 Google Colab 作为训练平台。作为一份免费资源，训练时只能分配到一个装有 cuda 的GPU，但训练速度已经可以接受——两个模型均需2.5个小时左右。

最终结果

由于本项目以产品设计和构成为主导，预实现部分产品功能即可，因此我没有做过多的参数调节对比和分析。单纯就这两个模型来看，训练成本上 CNN + Bi-LSTM + Attention 模型要略低于 Transformer 模型——时间短30分钟左右；效果上来看，Transformer 模型所得的神经网络在应用中的效果更好，所得下联的上下文联系更紧密，逻辑更通畅，这可能与模型本身的侧重点有关。因此，最终使用的网络推荐为 Transformer 模型。

对联示例

- 上联：一曲笙歌春似海 下联：千杯锦色福如山
- 上联：放不开眼底乾坤，何必登斯楼把酒 下联：吞得尽胸中云梦，方许对古人言诗

四、实验心得

李冠达：

本课程项目中，我通过队内商讨出来的产品设计和构造思路，大致完成了该产品的部分功能——一个基本的前端网页和两个效果中等的自动根据上联对下联的神经网络。

1. 构造前端的过程中，手写“枯燥”的 `html` 和 `css` 格式控制让我对 `html` 和 `javascript` 理解更加深刻
2. 而构造后端过程中，模型的选取与训练、相关文献的查阅和硬件资源的搜寻也让我对NLP相关领域的知识和方法有了一定程度的了解，并知道了 `Google colab` 这个免费的训练平台。

总之，这次项目从产品调研、产品立项再到产品设计、产品实现，让我真实地体验到人工智能相关技术投入于产业界、产品化的过程，让我从校园内的纯技术知识走向了实践和应用，收获颇丰！

五、参考文献

[1] `Self-Attention` 层的设计参考了 `pytorch-pretrained-BERT`，地址<https://github.com/huggingface/pytorch-pretrained-BERT>。结构设计参考了 `NERs`，<https://github.com/WiseDoge/NERs>。

[2] 数据源地址：<https://github.com/wb14123/couplet-dataset/releases/download/1.0/couplet.tar.gz>。