



第十四届全国大学生软件创新大赛

文档编号: SWC2021-T20210267-搞快点



我的动态

iMoments

技术研究报告

Version: [V2.0.0]



搞快点

2021-4-10

All Rights Reserved

目录

1	问题聚焦	1
1.1	问题描述	1
1.2	问题抽象	1
1.3	问题定位	2
1.4	问题评估	2
1.5	问题分解	2
2	相关工作	3
3	技术方案	4
3.1	技术方向	4
3.2	技术选择	4
3.3	结果期望	4
4	技术实践	5
4.1	使用的开发框架及依赖的库	5
4.2	技术实践过程	5
5	结果验证	7

文档修订历史

序号	修订原因	版本号	作者	修订日期	备注
1	创建封面	V1.0.0	队员 A	2021-1-8	
2	填写技术方案	V1.1.0	队员 B	2021-1-8	
3	填写问题聚焦	V1.2.0	队员 A	2021-1-9	
4	修改技术方案	V1.2.1	队员 B	2021-1-9	
5	修改技术方案	V1.2.2	队员 B	2021-1-10	
6	修改问题聚焦	V1.2.3	队员 B	2021-1-10	
7	填写相关工作	V1.3.0	队员 B	2021-1-10	
8	封面修改	V1.4.0	队员 D	2021-1-10	
9	最终修改	V1.4.1	队员 B	2021-1-10	
10	复赛补充	V2.0.0	队员 B	2021-4-9	

1 问题聚焦

1.1 问题描述

对应项目要实现的主要功能，有以下五大问题需求：

① 图片分析：

在给出的图片中进行通用物体和场景识别，得到图片的关键要素及相关性得分。

② 文本分析：

根据给出的文字，分析语义主题和情感倾向，并得到相关性得分和情感倾向得分。

③ 配文生成：

根据得到的图片关键要素、文本语义、情感倾向、用户风格偏好，生成有意境有美感的配文。

④ 相关图片和表情的检索和推荐：

根据图片关键要素、文本语义、情感倾向，在云端网络和数据库中检索相关图片和表情包。

⑤ 图片处理：

对模糊照片进行修复处理，并套用简单的滤镜和色调调整。

端云结合可能面临的问题：

① 客户端和服务端的适配：

客户端预计实现手机（安卓）app、微信小程序、网页端、PC 端中的两个以上，主要为安卓端。前端设计要美观、好用。

服务端同时接受不同客户端的请求，要稳定可靠。

② 人工智能、大数据在端云部署和工作分配问题。

1.2 问题抽象

① 图像要素信息提取

② 文字语义要素和情感提取

③ 关键词生成配文

④ 快速检索相关图片

⑤ 图像修复和艺术处理

1.3 问题定位

问题业务领域: 端云结合的针对图像与文字分析处理所提供的人工智能服务。

问题技术领域: 人工智能对图像识别和处理、人工智能对自然语言的分析、自然语言生成、机器学习、大数据检索

1.4 问题评估

人工智能对图像和文字的分析处理是当下的一大热点, 各大云服务商都提供了有效的 API 接口供使用。在此基础上, 还需要结合动态、朋友圈的特点, 训练自己的模型, 实现更准确、更人性化的服务。

项目技术性要求较高, 热度较高, 普适性好。

1.5 问题分解

问题一 图片与文字的分析:

- ① 提取图片主要元素
- ② 提取自然语言表达要点
- ③ 关联主题, 分析情绪 (依赖①②)
- ④ 根据获取的碎片化信息检索素材 (依赖①②③)

问题二 配文生成

根据分析结果生成配文 (依赖问题一)

问题三 图片处理

- ⑤ 基本的图片编辑
- ⑥ 图片修复和艺术处理 (依赖⑤)

问题四 端云分布式设计

- ⑦ 端云工作分配
- ⑧ 前端设计 (依赖⑦)
- ⑨ 后端服务设计 (依赖⑦)

2 相关工作

- ① “九歌”: 清华大学自然语言处理与社会人文计算实验室研发的自动古代诗歌生成系统。该系统基于工作记忆方法对诗歌进行系统分析, 支持用户输入多个关键词并单独保存在模型中, 并根据对已有数据集的相关性检验, 筛选出相关性高的语句来生成诗歌。
- ② dcalsky/AIPoet (GitHub): 基于人工智能的 Android 写诗应用, 无须联网随时随地写藏头诗、意境诗。使用长短时循环神经网络, 针对五万多首唐诗进行训练, 并且将训练模型移植到 Android 客户端中
- ③ ChineseLyrics 中文歌词数据库 (GitHub 开源): 通过网络收集整理的中文歌词数据库, 包含:绝大多数华语歌手在 2019 年之前的歌曲、4019 位歌手, 其中作品数 20 首以上的 1086 人, 100 首以上 233 人、102197 首歌曲, 平均每人 25.4 首歌。
- ④ ChineseBQB 开放数据源: 表情包目录(共收录 4423 张表情包), 且有详细分类。

3 技术方案

3.1 技术方向

深度学习:

- ① 识别图片中主要对象和特征
- ② 识别文字中的主题和情感倾向
- ③ 根据特征、主题关键词和情感倾向创作配文
- ④ 对模糊或低像素图片进行修复和超分辨率

云端大数据:

- ① 查找网络上的相关图片
- ② 查找能表达用户心情的表情包

3.2 技术选择

图像通用物体和场景识别:	百度 AI 开放平台
文字主题和情感倾向提取:	百度 AI 开放平台
诗歌生成:	Tensorflow + WorkingMemory
散文歌词生成:	Pytorch + GPT-2
修复图片和超分辨率:	OPPO AI Unit
相关图片查找:	百度 AI 开放平台、百度图像搜索 API
表情包查找:	百度 AI 开放平台、百度图像搜索 API

3.3 结果期望

在用户可接受的时间范围内, 根据用户提交的主题关键词以及图片或文字, 生成和提交内容相关的配文, 向用户推荐类似的图片或表情包, 并对低分辨率和模糊的图片进行处理和修复。

散文要求语言自然通畅, 有一定美感, 与用户意图没有明显偏差。

诗歌和歌词要朗朗上口、有一定的意境。

补充的图片要求与主题相关, 补充的表情包要求能准确表达用户心情。

修复处理的图片要求比原图清晰度提升, 细节辨识度更高。

4 技术实践

4.1 使用的开发框架及依赖的库

移动端: Flutter、Flutter Dio、Shared Preference、Flutter Toast 等

移动端 AI 框架: TensorFlow Lite

云端: Golang + GIN

云端 AI 框架: TensorFlow、Pytorch

4.2 技术实践过程

表 1 移动端实践过程

时间	内容过程	人员安排
2.10 – 2.20	完成登录页面编写, 实现与后端通信	队员 C
2.21 – 2.28	完成 Gallery 及页面内图片浏览功能	队员 C
3.1 – 3.9	完成图片和文字提交页面, 实现与后端通信	队员 C
3.1 – 3.9	完成个人页面编写, 实现个人信息修改功能	队员 D
3.10 – 3.20	实现收藏查看功能	队员 D
3.10 – 3.20	整合客户端人工智能推断模型	队员 B、C
3.21 – 3.31	重新组织页面, 美化主题, 修复若干 Bugs	队员 C

表 2 云端实践过程

时间	内容过程	人员安排
2.10 – 2.20	完成 GIN 和 MySQL 基本框架搭建, 实现登录功能, 并申请公网 IP	队员 A
2.21 – 2.28	编写网页, 申请域名, 并向前端提供 Gallery 图片	队员 A
3.1 – 3.9	完成图片和文字提交记录保存, 并将图片发送至百度云进行图像识别, 将结果返回客户端	队员 A
3.1 – 3.9	实现用户信息修改、用户管理封禁等功能	队员 A
3.10 – 3.20	实现收藏内容保存功能	队员 A
3.21 – 3.31	整合部署云端人工智能推断模型	队员 A、B
4.1 – 4.5	美化介绍网页, 修复若干 Bugs	队员 A

表 3 人工智能推断实践过程

时间	内容过程	人员安排
2.10 – 2.25	训练写诗模型并优化	队员 B
2.26 – 3.5	使用 TensorFlow Lite 将模型转换并移植到 Flutter	队员 B
3.6 – 3.10	与客户端整合人工智能模型	队员 B、C
3.10 – 3.20	训练写歌词模型并优化	队员 B
3.21 – 3.25	训练写散文模型并优化	队员 B
3.25 – 4.5	完成模型云端部署和配置	队员 A、B

备注:

由于项目使用的是 flutter 框架, oppo ai unit 也没有 flutter 支持, 所以很遗憾之前设想的照片降噪修复等功能无法实现。

相关图片的查找暂时只做了已有图片的浏览, 后续会加上对图片搜索 API 的调用。

5 结果验证

普通古诗创作示例:

示例 1:

输入文字: 江南古镇

输入图片:



输出:



示例 2:

输入文字: 古韵小城

输入图片:



输出:



输出的古诗语句较为流畅,能够切合图片的主要内容信息,也能有一定的押韵。虽然无法达到写诗大家的水平,但已经能比一般人类写的诗更优美。

藏头诗创作示例:

示例 1:

输入: 天气真好

输出:

天气真好:
天道与我心, 不知何处
无。
气吞万里山, 竦倪一时
昇。
真人与形影, 此地无为
媒。
好鸟不敢飞, 老翁何所
穷。

示例 2:

输入: 春光明媚

输出:

春光明媚:
春风澹无情, 一瞬不能
任。
光景与君别, 相欢在泮
滨。
明朝有余酒, 白发如清
沦。
媚怕贡言间, 不如山与
人。

输出的藏头诗语句通畅, 押韵也比较恰当。但与主题切合度不高, 分析原因是因为要将模型放在客户端运行, 为追求较低的内存和性能占用, 对模型进行了较大的简化, 导致效果不佳。

同时我们也发现, 生成的藏头诗偶尔会出现几句诗是重复的, 并且效果较差, 但是这种情况出现概率较低, 50 次尝试内约有 4 次出现, 概率在 10% 以下。

歌词创作示例:

输入: 我们的爱情

输出 1:

我们的爱情, 不是在演
戏, 我不要你的美丽, 也
不要你的温柔, 我不要你
的承诺, 也不想你对爱已
远走, 就算是今生无缘,
我也不会再强求, 不想让
你再伤心, 让你承受孤
独, 只要你能快乐, 我会
用我一生去爱你
你。

输出 2:

我们的爱情已经到了尽
头, 曾经的誓言, 只剩下
我一个人守候, 那些所谓
的幸福, 现在变成了孤
独, 我的世界只剩下了痛
苦, 我的爱情再也不会
有未来, 如今你的世界
里, 我已不能够回头, 你
给的爱已经太多, 你给的。

输出的歌词较为通畅，有一定的美感，内容能贴合主题词。但是最后一句话往往不全，这是由于人工智能模型的输出已达到最大输出的长度，为 90 个字符。我们目前并没有找到较好的方式截断最后一句话，但是由于用户选择后可以编辑，不会对使用造成障碍。

有时模型会输出重复的文字，如输出 2 的最后几个词已经和前文有所重复。原因可能是要求输出的文字长度过长。经过测试，有文字重复的概率约 40%，大面积的文字重复出现概率约为 20%。同样地，用户可以对分享的内容进行修改，可能需要用户进行手动的删除操作，影响了用户体验。

散文创作示例：

输入：阳光明媚，春暖花开

输出 1：

阳光明媚，春暖花开。我们走进那条熟悉的小路，听着它们的歌谣，想着那些过去的往事，一幕幕的呈现在眼前。一切的一切，仿佛都在慢慢的远去。春天走近，秋天走近，冬天走远，心中有着难以言表的情愫，一点点的。

输出 2：

阳光明媚，春暖花开的时候，我总会在春天的窗台，看着满山遍野的油菜花在蓝天下盛开，那醉人的花香，那醉人的花香，让人痴迷，让人流连。我的故乡在绿油油的麦田里，在绿荫下举目眺望，那美丽的油菜花在微风里轻。

由于该模型结构与歌词创作模型类似，只是训练的数据集不同，因此也存在歌词创作模型类似的特性。同样地，输出的散文较为通畅，有一定的美感，内容能贴合主题词，最后一句话也会不完整。

在输出 2 中也出现了重复的话语，经过测试，文字重复的概率约 30%，大面积文字重复的概率约 20%。