FAKE TEXT GENERATION

赖昱行
Yuhang Lai
北京理工大学
Beijing Institute of Technology

日期：2022 年 5 月 21 日

摘　　要

本文为网络与信息安全课程大作业的技术报告，程序使用 Python 语言编写，选题方向为人工智能安全方向。程序通过部署 DeepFake 接口，使用人工智能实现虚假文本的生成，并在此基础上编写爬虫爬取知乎热榜，进行了技术实践。报告主要介绍了如何设计和实现 FakeTextGenerator，如何设置和调用 DeepFake 相应模型的接口，如何编写爬虫爬取知乎热榜信息。最后，报告展示并分析了实验结果。本次实验相关内容已上传至 Github[1].

1　实验目的

此次实验旨在部署 DeepFake，编写实现虚假文本生成器，使用生成后的虚假文本进行网络舆论攻击，并以期获得一定的关注度和影响力。在实验结束后，应对实验结果进行总结和评估，分析攻击效果。

2　FakeTextGenerator

2.1　结构设计

本次作业设计了 FakeTextGenerator 类，用于部署模型 config，API 身份验证，文件 IO 等，提供了生成虚假文本的简单接口。在使用时，可以以文件形式输入提示文本，也可以在控制窗口中输入，文本输出同理。FakeTextGenerator 引用了多个 AI 模型，提供中文和英语两种语言的虚假文本生成功能。同时，FakeTextGenerator 能够读取爬虫爬取的网络热点信息并针对性生成虚假文本。

2.2　代码实现

main.py 函数中实现了 FakeTextGenerator 类。类中存放了输入输出的文件路径，AI 模型的 config 设置以及 APItoken，并将 AI 模型使用的 prompt 作为私有成员存放在类中。

文件路径的设置可以简单地调用以下函数

[1] halfrot/fake-text-generation

```
FakeTextGenerator().setInfilePath(filepath)
FakeTextGenerator().setOutfilePath(filepath)
```

同时也可以直接传入字符串形式的 prompt

```
FakeTextGenerator().readSentence(prompt)
```

query_ch 和 query_en 用于将 prompt 传入 AI 模型，若出现错误则返回 None，否则在控制台打印并返回生成文本。其中 query_ch 表示中文文本生成，query_en 表示英语文本生成。在函数实现时，需要对 FakeTextGenerator 中文件输入输出的路径进行判断，并检查 prompt 是否存在。对于不同的 AI 模型，有不同的 config 设置和调用接口，需要将 prompt 填入相应的 json 文件。核心代码如下

```
data = self.defaultConfig_ch.copy()
data["prompt"] = self.__prompt
response = executeEngine(wudao.ability_cc, wudao.engine_cc, self.token_ch, data)
```

query_ch 和 query_en 最终返回的结果是由 prompt 和生成文字拼接在一起的字符串。

与爬虫对接的部分为 generatorOnZhihu 函数，用于处理爬虫爬取的知乎热榜 top50 的信息。在处理时，取用热榜的标题信息作为 prompt，并调用 AI 翻译模型翻译成英语，作为第二个 prompt，将中英文的生成文本分别写入到两个不同的文件中。翻译部分所使用的为 translate，作为 FakeTextGenerator 的私有成员函数，与生成函数部分编码类似。

3 模型调用

3.1 悟道

悟道作为此次选用的中文文本生成的 AI 模型，选用了 Transformer-XL[2] 作为模型基础结构。Transformers 作为近年来流行的通用 AI 模型，具有长期学习依赖的潜力，但在语言建模设置中受到固定长度上下文的限制。而 Transformer-XL 作为一种新神经网络架构可以很好地解决长文本依赖问题。悟道基于 Transformer-XL 训练并开放 29 亿的语言模型，在长文本生成方面具有优势。FakeTextGenerator 中调用的正是悟道的 question-answer 模型。

```
response = executeEngine(wudao.ability_cc, wudao.engine_cc, self.token_ch, data)
```

3.2 GPT-J

在英语文本生成方面，选用了 EletherAI 发布的 GPT-J-6B 版本。GPT-3[1] 可以做到这一领域的最强的效果，但由于 OpenAI 的 GPT-3 接口暂不开放给中国区域，本次作业选用了 GPT-J。GPT-J 是一个基于 GPT-3 的自然语言处理 AI 模型，该模型在一个 800GB 的开源文本数据集上进行训练，并且能够与类似规模的 GPT-3 模型相媲美。为了方便 API 的调用，直接选择了 huggingface 上提供的接口。

```
response = requests.post(huggingface.gptj, headers=self.token_en, json=data)
```

3.3 opus-mt-zh-en

为了实现 FakeTextGenerator 中翻译文本的需求，选择了 Helsinki NLP lab 开放的 opus-mt-zh-en 模型。此模型可以将中文直接翻译成英语，且同样在 huggingface 上提供了调用接口。

```
response = requests.post(huggingface.zh2en, headers=self.token_en, json=data)
```

4 爬虫实现

本次爬取的网站为知乎热榜[2]，由于网站使用了动态加载，故需使用 selenium 下的 webdriver 进行浏览器模拟操作。本次实验选用浏览器为 Chrome 以及配套 chromedriver。等待网页加载完毕后，找到热榜对应的 HotList-itemBody 即可爬取信息。

5 实验

5.1 实验方法

本次实验的目的是使用上述编写的 FakeTextGenerator 配合爬虫爬取的知乎热榜信息，进行虚假文本的生成，然后在知乎上回答对应热榜问题，通过记录回答最终的浏览量和点赞量来评价虚假文本的影响力。

由于知乎必须进行实名注册登录，并在使用账户一段时间之后才能在同一天内回答较多问题，这里使用了作者的私人账号进行实验。

5.2 实验结果

实验爬取了 2022 年 5 月 21 日 14:30 时刻的知乎热榜 top50 信息，排除 2 条政治敏感新闻，共采集到 48 条可能的 prompt 语句，耗时 1419 秒。由于账号限制，选择了 top15 的热榜问题进行回答。方法为热榜标题作为 prompt，将中文生成文本与英语生成文本拼接作为回答内容，人工提交到知乎对应问题的回答区。经过 7 小时，在 2022 年 5 月 21 日 21:40 收集各回答的详细数据，浏览量共计 3368，赞同数共计 8，具体结果见表1。

5.3 实验分析

可以看到，这 15 条回答经过 7 小时后取得了不错的浏览量，同时赞同数较少。由于缺乏足够的账号，乃至于不同曝光度的账号来进行对比实验，只能感性地认识一下这次实验的效果。而根据作者以往回答的情况比较，在知乎热榜的基础上生成的虚假文本确实获得了令人满意的浏览量，具有进行网络舆论攻击的潜力。

[2] http://www.zhihu.com/billboard

Fake Text Generation Results				
编号	问题标签	浏览量	赞同	问题浏览量
1	军事	6	0	2,909,008
2	体育	118	0	1,294,569
3	影视	78	1	1,067,143
4	疫情	509	0	1,717,414
5	职场	7	0	920,600
6	经济	255	1	711,605
7	疫情	1567	1	1,359,292
8	节气	103	0	179,399
9	就业	83	0	503,120
10	医院	85	1	2,672,236
11	军事	58	0	1,329,897
12	军事	146	1	287,462
13	政治	149	1	622,313
14	娱乐	34	1	1,528,494
15	军事	302	2	162,514

表 1: Results of `FakeTextGenerator` running on Zhihu hot

6 结语

本次实验所使用的 AI 模型以及 prompt 的选用方法皆存在可改进之处，对网络平台的攻击也可以使用更自动化的方式，从而进行大规模的舆论攻击。而目前类似 GPT-3 这样的超级 AI 模型仍没有对中国区域开放使用，作者认为应该提高对虚假文本生成基础上的网络舆论攻击提高警惕，防止潜在组织恶意攻击，操纵网络舆论，扰乱社会秩序。我们需要相应的安全技术用来保护社会免受人工智能攻击的侵袭。

参考文献

[1] Tom Brown et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[2] Zihang Dai et al. “Transformer-XL: Attentive Language Models beyond a Fixed-Length Context”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 2978–2988.