文章编号：2095-2163(2019)01-0259-03 中图分类号：TP317.1 文献标志码：A

基于人工智能的政务办公系统预置批示与分工方法

黎嘉明
（广东省环境信息中心，广州 510220）

摘 要：人工智能技术特别是机器学习分支已在许多领域广泛应用，本文对多个与文本处理相关的机器学习技术进行研究，并利用这些技术及政务办公系统积累的样本数据建立模型，在政务办公系统中的“中枢节点”实现批示和分工的预置。
关键词：人工智能；机器学习；政务办公系统；批示；分工；预置

Preset instructions and assignments in government office automation system based on artificial intelligence

LI Jiaming
（Guangdong Environmental Information Center, Guangzhou 510220, China）

【Abstract】 Artificial Intelligence technology, especially machine learning branch, has been widely used in many fields.The paper researches some machine learning technologies about text processing with 10 years accumulated sample data in a government office to build models. The “central node”in the system implements the preset of instructions and assignments.
【Key words】 Artificial Intelligence; machine learning; government office automation system; instructions; assignments; preset

0 引 言

机器学习是人工智能目前应用最广泛、最成功的一个分支。在解决某一类问题时，机器学习让计算机可以从观测数据（样本）中寻找规律，并利用学习到的规律（模型）对未知或无法观测的数据进行预测。近年来，在深度学习算法以及专用处理器软硬双引擎的推动下，机器学习已广泛应用于数据挖掘、计算机视觉、生物特征识别、语音识别、语音合成、机器翻译、无人机等领域，并接近甚至超过人类的水平。从 2017 年开始，本文作者将机器学习的技术应用到政务办公系统中，希望人工智能的引入能大大提升其中一些关键环节的工作效率。经过对业务数据的发掘和模型的测试，基于哈工大 ltp、谷歌 tensorflow、keras、LSTM 等技术，开发出人工智能系统，可实现批示文本和部门分工的预置。

1 政务办公系统的痛点——“中枢节点”

政务办公系统是政府信息化中发展比较成熟的系统，其推行使政府部门实现了“无纸化”办公，大大提高了工作效率，也使政府内部的办公流程实现了标准化和规范化。政务系统的工作流中有各种各样的节点，如收文、办文、存档等，每个节点是一个处理环节，有固定的处理人员和工作任务。其中有一个特殊的节点，所有公文都需要先汇集到此，然后经过其分发，称为“中枢节点”。处理人需要根据来文的内容，做出初步的批示，以及将任务分工给相应的领导或者部门。这项工作需要处理人对部门分工非常熟悉、对业务工作有广泛的认识，而且言辞要严谨得当，办理速度要快捷。这个节点也成为政务办公流程中的“痛点”，一方面所有流程都经过该节点，而且时效要求高，一旦滞后会影响后续很多的环节；另一方面其工作质量要求高，能高质量完成这个处理工作的处理人员又很少。

2 为“中枢节点”提供预置数据

在烦杂的文件处理过程中，若“中枢节点”的 2 项工作成果（批示和分工）能够在一开始就预置好，处理人员就可以此为蓝本做些修改或者直接使用其作为结果，这无疑可以大大提高处理的效率，加快流转的速度。利用机器学习技术，基于历史数据作为样本，可以训练出一个模型，实现这些预置数据的推演生成。在“中枢节点”这个场景中，样本就是十年来积累的约 4 万个来文的信息（包括时间、标题、来文部门、文号、正文等）以及处理人员做出的结果（包括批示和分工）。可以把来文信息设为 x，而把做出的结果设为 y，利用现有样本数据训练出 $x -> y$ 规律的模型，有新的来文时，就是新的 x，可以通过

作者简介：黎嘉明（1975-），男，学士，教授级高级工程师，副主任，主要研究方向：电子政务、计算机科学、环境管理。
收稿日期：2018-11-07

模型推演出新的 y,作为预置的批示和分工。

作为模型输入的来文信息,都是文本这种非结构化的数据,而对于文本(字符串)一类数据在机器学习领域比较成熟有效的方法是使用循环神经网络(Recurrent Neural Network),简称 RNN,以及其派生的 LSTM 等模型。但作为模型输出结果数据的批示和分工是 2 种不同类型,其中批示是文本,同样可以用 RNN、LSTM 模型来处理。目前,机器学习领域可使用 seq2seq(sequence to sequence,即序列到序列)模型实现字符序列(文本)到另一个字符串序列的转换,即用 LSTM 对输入序列进行编码,得到一个向量,然后用 LSTM 对向量解码,生成输出的序列。另外一种输出结果是分工,这里的分工是指比如有 30 个部门,应该由哪些部门对这个文件做处理,这其实是一个多选结果(multi-hot)的结构化的数据。目前,可以用文本分类(text classifier)模型来实现字符串序列(文本)到多选结果的转换。

3 预置系统的开发过程

系统采用 python3.6 语言开发,运行在 ubuntu 16.04 操作系统上,使用 tensorflow1.6 作为机器学习框架,硬件加速设备为一个 nvidia 970 GPU,使用这些平台和工具主要是因为这些平台和工具都是业界比较成熟和主流的技术,有非常多的学习资料和经验可供借鉴,可减少搭建开发平台的耗时,将时间集中用于业务数据的分析和模型的实现。

预置系统需要通过 4 万多条的样本数据训练 2 个模型,通过测试并达到较高的准确度后,再与政务办公系统连接,自动为“中枢节点”推演出预置的批示和分工,提供给处理人员参考。系统还会将处理人员实际的操作结果(有可能修改了预置的内容)用于持续训练模型,使后面的推演更接近新的实际的要求。其中,推演的步骤比较简单,只要加载之前训练好的模型,将新的 x(来文信息)输入,即可推演出新的 y(批示和分工),以此作为预置交给政务办公系统。而持续训练的步骤与初始训练差不多,差别只在于样本的多少。因此,本文主要介绍模型的初始训练过程。如图 1 所示,训练主要由 3 部分组成,第一部分是对样本数据的预处理,然后针对 2 种结果数据(批示、分工)分别建立模型做训练。

预处理是训练之前必不可少的步骤,其目的是将从政务办公系统中获取的“业务”数据转化成一般的机器学习模型可处理的“矢量”数据。例如,来文信息文本为“关于召开全省‘两学一做’学习教育工作会议的通知”,需转化成数字序列“72 96 326 97 1219 4550 1032 299 107 448 740 142 82 80 83”。“矢量”转换需要对原始文本分离编码,根据汉语的特点,要以“词”而不是以“字”为单位对文本进行分离,这就是“分词”。本系统使用了哈工大社会计算与信息检索研究中心研制的语言技术平台(ltp)作为分词工具。而且根据政务的特点,将单位部门以及单位领导的全名和简称都作为 ltp 工具的预置词典,可让分词更加准确。将分词后的词去重,一行一个词归纳到一个文本文件中,行号(数字)就成为这个词的代号,这就归纳成了一个囊括样本数据中所有词的“词典表”,通过这个词典表的词与行号的映射,就可将文本的词序列转变为数字序列。

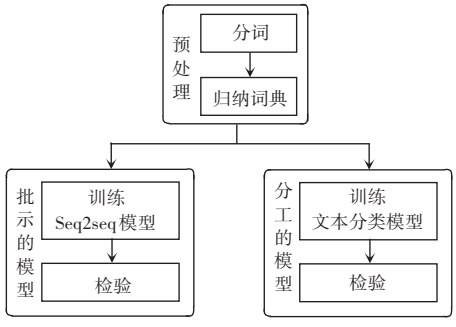

图 1 模型初始训练过程图

Fig. 1 Model initial training process chart

生成批示的模型是一个 seq2seq 模型,其输入输出都是文本序列,主要思路是通过深度神经网络模型(常用的是多个 LSTM——长短记忆网络)将一个作为输入的序列映射为一个作为输出的序列,这一过程由编码输入与解码输出 2 个环节组成。本系统测试了多个 seq2seq 模型,尝试了在不同参数下使用 4 万多条样本数据对这些模型进行训练,对比这些模型的准确率、损失、耗时等指标,选定了在编码和解码各自使用 2 层 LSTM,并使用注意力机制(Attention Model),这样构造的 seq2seq 模型在效率和准确率方面都有令人满意的结果。模型的训练监视曲线如图 2 所示。

生成分工的模型使用的是一个文本分类(textclassifier)模型,文本分类是根据文本内容本身将文本归为不同的类别。在本系统中的模型,需要的是多种分类的映射,即一个文本可以在若干个分类中对应多个,类似多选题,而每一个分类,在本系统的场景中,就代表一个分工。此外,在笔者工作的部门,“中枢节点”有 3 种类型的分工:领导办理、部门主办、部门传阅。因此,还需要为 3 种类型的分工各自训练文本分类模型。经过对比测试,使用了词

向量和双向 LSTM 层来实现这些文本分类模型，可以得到比较好的准确率和运行效率。模型训练的监视界面如图 3 所示。

由图中数据可以看到验证的准确率已经达到 0.98 以上。预置系统中的 2 个经过训练的模型准确度较高，与政务办公系统连接后，则每 15 min 查询“中枢节点”中未处理的来文。根据来文信息利用模型推演出批示和分工信息，并预置到办公系统中。待处理人员办理时，将依据预置的信息去办文，一般只需要少量的改动或者不改就可以形成处理结果，工作效率大大提高。

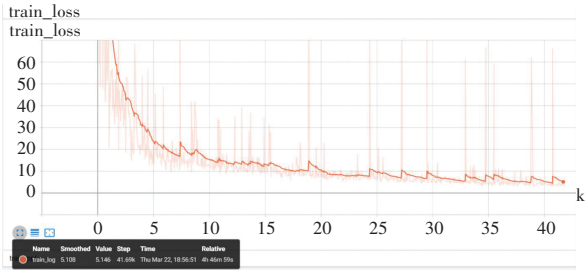

图 2　模型的训练监视图

Fig. 2　Monitoring of model training

```
15279/15279 [==============================] - 150s 10ms/step - loss: 0.1169 - acc: 0.9565 - val_loss: 0.0558 - val_acc: 0.9854
Epoch 2/10
15279/15279 [==============================] - 148s 10ms/step - loss: 0.0604 - acc: 0.9845 - val_loss: 0.0545 - val_acc: 0.9854
Epoch 3/10
15279/15279 [==============================] - 147s 10ms/step - loss: 0.0577 - acc: 0.9845 - val_loss: 0.0512 - val_acc: 0.9854
Epoch 4/10
15279/15279 [==============================] - 147s 10ms/step - loss: 0.0513 - acc: 0.9844 - val_loss: 0.0484 - val_acc: 0.9854
Epoch 5/10
15279/15279 [==============================] - 148s 10ms/step - loss: 0.0451 - acc: 0.9850 - val_loss: 0.0470 - val_acc: 0.9855
Epoch 6/10
15279/15279 [==============================] - 147s 10ms/step - loss: 0.0394 - acc: 0.9864 - val_loss: 0.0492 - val_acc: 0.9849
Epoch 7/10
15279/15279 [==============================] - 148s 10ms/step - loss: 0.0338 - acc: 0.9881 - val_loss: 0.0553 - val_acc: 0.9838
Epoch 8/10
15279/15279 [==============================] - 147s 10ms/step - loss: 0.0295 - acc: 0.9895 - val_loss: 0.0584 - val_acc: 0.9829
Epoch 9/10
15279/15279 [==============================] - 147s 10ms/step - loss: 0.0262 - acc: 0.9902 - val_loss: 0.0591 - val_acc: 0.9821
Epoch 10/10
15279/15279 [==============================] - 147s 10ms/step - loss: 0.0239 - acc: 0.9909 - val_loss: 0.0629 - val_acc: 0.9832
3820/3820 [==============================] - 8s 2ms/step

Test score: 0.063, accuracy: 0.983
```

图 3　模型训练的监视界面

Fig. 3　UI of text classifier model training

4　结束语

机器学习技术应用在政务办公系统“中枢节点”上能取得这么好的应用效果，可以说是“好钢用在刀刃”上。“好钢”指的是哈工大 ltp、tensorflow、LSTM 这一系列先进成熟的技术，“刀刃”就是“痛点”所在的中枢节点。一方面，这个节点很需要这种高效的辅助手段，因为这个节点的处理量很大，时限要求又高，处理滞后就会造成系统堵车；另一方面，机器学习技术也恰好适用于“中枢节点”的特点，这是因为“中枢节点”要求的处理结果是“明确的”、“严谨的”，短期内不易变化的，可以说是一种“规范化”的处理，这样的场景比较适合机器学习发挥其经验学习、历史模仿的优势。相反，倘若一个场景需要更多的创造性、自由发挥的思维，机器学习就难以给出满意的答案了。

参考文献

[1] CHE Wanxiang, LI Zhenghua, LIU Ting. LTP: A Chinese language technology platform[C]//Proceedings of the Coling 2010: Demonstrations. Beijing: dblp, 2010(8): 13-16.

[2] LUONG T, BREVDO E, ZHAO RuÍ. Neural machine translation (seq2seq) tutorial[EB/OL]. [2017]. https://github.com/tensorflow/nmt.

[3] LAI Siwei, XU Liheng, LIU Kang, et al. Recurrent convolutional neural networks for text classification[C]//Proceedings of AAAI 2015, Austin Texas, USA: AAAI, 2015: 2267-2273

(上接第 258 页)

[3] 李兴华，王月清. Java Web 开发实战经典基础篇[M]. 北京 清华大学出版社，2010.

[4] 张海藩，倪宁. 软件工程[M]. 3 版. 北京：清华大学出版社，2010.