doi: 10.7690/bgzdh.2022.10.008

# 基于深度学习的无人机指令意图识别技术

符　凯[1]，朱雪耀[1,2]，吕全喜[1,2]，姜　超[1,2]
（1. 航空工业西安飞行自动控制研究所，西安 710065；2. 飞行控制航空科技重点实验室，西安 710065）

**摘要：**为实现空管员直接发布指令来操控无人机，结合深度学习在自然语言处理(natural language processing，NLP)中的应用，提出一种基于深度学习的无人机指令意图识别方法。使用改进 Skip-Gram 模型生成指令文本的词向量，输入到卷积神经网络进行指令文本分类，得到空管员发布指令的意图。通过实验验证，结果表明：该方法能够较准确地对指令意图进行识别，有助于后续指令理解技术的实现，为进一步实现空管员与无人机直接交互做准备。

**关键词：**无人机；空地对话；自然语言处理；意图识别；卷积神经网络
**中图分类号：**V279　　**文献标志码：**A

# UAV Command Intent Identification Technology Based on Deep Learning

Fu Kai[1], Zhu Xueyao[1,2], LYU Quanxi[1,2], Jiang Chao[1,2]
(1. *AVIC Xi'an Flight Automatic Control Research Institute*, *Xi'an* 710065, *China*;
2. *Aviation Key Laboratory of Science and Technology on Aircraft Control*, *Xi'an* 710065, *China*)

**Abstract:** In order to realize that air traffic controllers can directly issue instructions to control unmanned aerial vehicles (UAVs), combined with the application of deep learning in natural language processing (NLP), a method of unmanned aerial vehicle (UAV) command intention recognition based on deep learning is proposed. The improved Skip-Gram model is used to generate the word vector of the instruction text, which is input into the convolutional neural network to classify the instruction text, and the intention of the air traffic controller to issue the instruction is obtained. The experimental results show that the method can accurately identify the command intention, which is helpful for the realization of the subsequent command understanding technology and for the further direct interaction between air traffic controllers and UAVs.

**Keywords:** UAV; air-to-ground dialogue; natural language processing; intent identification; convolutional neural networks

## 0　引言

随着协同控制技术和通信网络技术的不断发展，军民航空领域迫切需要对无人机进行空域整合，未来空中联合作战体系也将向“有人/无人协同作战”“全无人机协同作战”的方向发展，将要求无人机与有人机在同一空域共存。目前，操控一架无人机需配备多个地面操作人员，既需要处理飞行制导，又需处理任务和传感器管理，而且远程操控时，无人机操作员很难保持与机载飞行员相同的态势感知和活动节奏[1]。未来的系统将需要一个操作员或有人机飞行员同时操控多架无人机，必然给无人机操作员或飞行员增加巨大的操作负荷，尤其在终端区域遇到的独特问题[2]。空地对话作为无人机与空管员的一种重要通信方式。为了同时不牺牲操作效率和安全性，各种高效率的新型无人机交互技术得到了前所未有的重视；与其他方法如遥控器、手势、思维跟踪等相比，语音控制被认为是控制的最佳技术(在室内应用)，因其同时具备灵活性、可移植性和可扩展性[3]。

近年来，深度学习在计算机视觉和语音识别方面取得了显著成就。自然语言处理(NLP)技术包括但不限于词嵌入将单词转换为向量，这通常是第 1 个数据处理层，到主题建模、命名实体识别(named entity recognition，NER)即“从文本数据中提取出具有真实含义的实体”、语言建模如 Transformer 的双向编码器表示(bidirectionalencoder representations from transformers，BERT)或问答系统。随着过去 10 年空中交通管制(air traffic control，ATC)的现代化和发展，航空领域现有研究主要集中使用 NLP 技术来帮助飞行员解决飞机维护问题和进行航空分析。文献[4]开发了一个 NLP 通道来预测航空安全事件中的人为因素，其包括预处理、TF-IDF、Word2Vec 和 Doc2Vec 进行特征提取，以及使用半监督标签传播(labelspread，LS)和监督

**收稿日期：**2022-06-01；**修回日期：**2022-07-28
**作者简介：**符　凯(1995—)，男，陕西人，从事无人机控制研究。E-mail: 18392127840@163.com。

SVM 进行数据建模。迄今为止，NLP 技术大部分研究都集中在安全事件分析以及航空飞行手册或维护中的其他一些应用上。

结合以上现状，为了让无人机在国家空域系统(national airspace system，NAS)中安全运行，在终端区域实现无人机的类人反应操作挑战。笔者提出基于深度学习的无人机指令意图识别技术，即将语音识别后的空管员指令文本送入到卷积神经网络进行短文本分类任务，根据指令内容的不同，得到空管员发布指令的真实意图。该深度学习模型在其任务领域表现良好，可直接作用于后期的无人机指令理解技术，并嵌入到无人机端，开发出通过语音直接控制无人机的能力，为将来融合飞行和有人/无人协同中的人机交互做准备。

## 1 空地对话框架

如图 1 所示，无人机空地对话框架由管制中心、空地对话传输链路及无人机系统构成。无人机在接收到空管员或有人机飞行员发布的语音指令后，经语音电台识别得到文本指令，意图识别系统将对其领域相关性和执行意图进行判断，然后进行语义理解得到无人机可执行的结构化指令，用于飞行控制系统控制无人机的执行机构。理解语义后，无人机需对结构化指令进行复诵(即对结构化前后的指令进行语义相似度计算)，并通过语音合成反馈给空管员，经空管员确认，若指令理解正确(语义相似度达到所设阈值)，则将结构化指令直接加入到无人机飞行计划当中；若指令理解错误，则需要空管员重新发布。

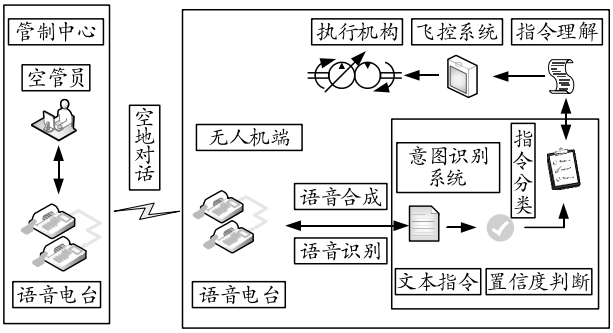


图 1 空地对话框架

笔者主要针对指令意图识别进行分析，包含 2 个子任务：置信度判断和基于短文本分类的指令意图识别。

置信度判断通过对识别后的文本指令与神经网络学习模型进行评估，判断空管员所发布指令是否属于无人机领域：如果不是(指令置信度 Confidence 小于所设阈值)，则返回提示信息“无法识别，请重新输入”；如果是，则基于短文本分类的指令意图识别，即采用卷积神经网络抽取指令的关键词作为特征训练分类器并分类，得到文本指令的意图。笔者共定义了 13 种指令意图。文本分类可分为单标记和多标记 2 种任务：单标记是指每个文本只属于一个类别，多标记是指一个文本可能属于一个或多个类别。笔者只分析单标记分类问题，但同一指令可能包含 2 种意图信息[5]，后期将对此问题作进一步分析。

## 2 原理及方法介绍

### 2.1 预训练词向量

目前，在无人机领域还未找到可用于训练的指令数据集，本文中所有指令集均由笔者人工标注完成。在没有大型监督训练集的情况下，使用从无监督神经网络语言模型中获得词向量是一种提高性能的好方法。笔者使用的是腾讯中文预训练词向量[6]，提供了 800 万中文词汇的 word embedding(200 维词向量)，每行是中文词及它的词向量表示，每一维用空格分隔，使用 Directional Skip-Gram(Skip-Gram 改进模型)训练而成。关于停用词、数字、标点等，为满足一些场景的需求，腾讯词向量并没有去掉这些，使用时需要自己构建词表并忽略其他无关词汇。该部分将用在指令理解任务中。

### 2.2 Directional Skip-Gram(DSG)模型

Skip-Gram(SG)根据中心词 $w_t$ 来预测上下文词 $w_{t+i}$，但没有考虑位置信息。基于此，Structured Skip-Gram(SSG)模型提出上下文不再由一个预测器生成，而是 2c 个预测器共同决定。具体地，对于任意一个词 $w_{t+i}$ 都会计算它出现在中心词 $w_t$ 每个上下文位置上的概率，然后全部相乘作为 $w_t$ 的预测概率。在 DSG 中，一个词的词嵌入不仅受到词共现的影响，还受到其上下文词方向的影响[7]。比如在无人机指令领域“上升”和“保持”都是“高度 $X$”的高频共现词，给定“高度 $X$”，识别要预测的词在左侧还是右侧对学习“上升”和“保持”的词嵌入非常重要。于是 DSG 提出如下 Softmax 函数：

$$g(w_{t+i},w_t)=\exp(\boldsymbol{\delta}_{w_{t+i}}^{\mathrm{T}}\boldsymbol{v}_{w_t})\Big/\sum_{w_{t+i}\in V}\exp(\boldsymbol{\delta}_{w_{t+i}}^{\mathrm{T}}\boldsymbol{v}_{w_t})。\quad(1)$$

式中：$V$ 为词汇表；$\boldsymbol{v}_{wt}$ 为 $w_t$ 的词嵌入；$\boldsymbol{\delta}$ 为向量，表示 $w_{t+i}$ 相对于 $w_t$ 的方向。

函数 $g$ 度量了上下文词 $w_{t+i}$ 在 $w_t$ 左侧或者右侧上下文时与 $w_t$ 的关联，共享了一个类似于负采样的

更新范式：

$$\begin{aligned}\boldsymbol{v}_{w_t}^{(\text{new})}&=\boldsymbol{v}_{w_t}^{(\text{old})}-\gamma(\sigma(\boldsymbol{v}_{w_t}^{\mathrm{T}}\delta_{w_{t+i}})-D)\delta_{w_{t+i}}\\ \delta_{w_{t+i}}^{(\text{new})}&=\delta_{w_{t+i}}^{(\text{old})}-\gamma(\sigma(\boldsymbol{v}_{w_t}^{\mathrm{T}}\delta_{w_{t+i}})-D)\boldsymbol{v}_{w_t}\end{aligned}。\tag{2}$$

式中：$\sigma$ 为 Sigmoid 函数；$\gamma$ 为衰减的学习率；$D$ 为给定 $w_t$ 条件下指定 $w_{t+i}$ 相对方向的目标标签，当 $i$ 小于 0 时，$D$=1，反之 $D$=0。

DSG 最终将前两者综合作为损失函数：

$$L_{\text{DSG}}=\frac{1}{|V|}\sum_{t=1}^{|V|}\sum_{0<|i|\leqslant c}\log(p(w_{t+i}\,|\,w_t)+g(w_{t+i},w_t))$$

$$p(w_{t+i}\,|\,w_t)=\frac{\exp(\sum_{r=-c}^{c}v'^{\mathrm{T}}_{r,w_{t+i}}v_{w_t})}{(\sum_{w_{t+i}\in V}\exp(\sum_{r=-c}^{c}v'^{\mathrm{T}}_{r,w_{t+i}}v_{w_t}))}。\tag{3}$$

式中：$|V|$为给定语料库词汇表大小；$c$ 为预测器数量，表示 $v'_{r,w_{t+i}}$ 是 $w_{t+i}$ 在 $w_t$ 每一个相对位置 $r$ 上的输出表征。

### 2.3　卷积神经网络(CNN)模型

CNN 最初为计算机视觉而发明，随后被证明对 NLP 有效并在语义解析上取得了优异的成绩[8]。文本分类的关键在于抽取文档或句子的关键词作为特征，基于这些特征去训练分类器并分类[9](笔者共定义 13 种指令意图)。因为 CNN 的卷积和池化过程就是一个抽取特征的过程，当准确抽取关键词的特征时，就能准确地提炼出文档或句子的中心思想[10]。Text-CNN 模型 RU 如图 2 所示。

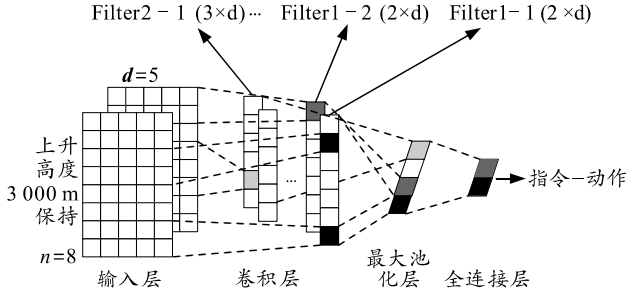


图 2　Text-CNN 模型

输入层(embedding layer)也叫词嵌入层，对输入的文本指令进行词嵌入，对于每个词生成一个词向量 $\boldsymbol{x}_i$ 并拼接成矩阵 $\boldsymbol{x}_{1,n}$，使其能够进行卷积操作，通常使用 word2vec、glove 等 word embedding 实现。$\boldsymbol{d}$=5 表示每个词转化为 5 维的向量，文本指令包含 8 个字，输入矩阵的维度即[8×5]。

$$\boldsymbol{x}_{1,n}=(x_1,x_2,\cdots,x_n)。\tag{4}$$

卷积层(convolutional layer)使用一个宽度为 $d$，高度为 $h$(通常取值 2,3,4,5)的卷积核 $w$ 与 $x_{i:i+h-1}$ 进行卷积操作，同一高度也可对应不同的卷积核(如图 2 中的 Filter1-1 和 Filter1-2)，得到更多不同特征，再使用激活函数激活得到相应的特征 $c_i$，则卷积操作可表示为：

$$c_i=f(w\otimes x_{i:i+h-1}+b)。\tag{5}$$

经过卷积操作之后，将 $c_i$ 拼接为一个 $n$-$h$+1 维的向量：

$$\boldsymbol{c}=[c_1,c_2,\cdots,c_{n-h+1}]。\tag{6}$$

最大池化层(max-pooling layer)：不同高度的卷积核使得卷积层后得到的向量维度不一致，使用 1-Max-pooling 将每个特征向量池化成一个值，即抽取每个特征向量的最大值表示其最重要的特征，最后将每个值拼接起来得到池化层最终的特征向量 $\boldsymbol{z}$：

$$\hat{c}=\max\{\boldsymbol{c}\},\ \boldsymbol{z}=[\hat{c}_1,\hat{c}_2,\cdots,\hat{c}_m]。\tag{7}$$

全连接层(fully connected layer)加上 Relu 作为激活函数，作用是对池化层结果进行连接，得到文本最终的特征向量 $\boldsymbol{C}$，该特征向量被输入到最后的 Softmax 层得到属于每个类别的概率，置信度 Confidence 则是取输出概率的最大值。

## 3　实验结果与分析

### 3.1　数据集

笔者使用的指令数据集来自 PEPEC900 句、民航空中交通无线电通话用语[11]及人工扩充 3 部分，数据需要经过筛选、修改和添加以适应无人机飞行任务，还有指令的标注与核对等工作均由人工手动完成。目前，无人机指令集共 2 000 条，并以 7:2:1 的比例分配给训练集、验证集和测试集，数据集总共定义 11 种指令意图(由于特定场景指令数据集欠缺，意图分类不能完全代表真实无人机场景)：指令-开关，指令-同意，指令-模式，指令-状态 1，指令-状态 2，指令-状态 3，指令-动作，指令-报告，指令-查询，指令-天气和指令-请求。

实验结果采用准确率(accuracy，Acc)和 F1 值对模型的性能进行评价[12]，评价指标定义如式(8)—(11)所示。

$$\text{Acc}=(TP+TN)/(TP+TN+FP+FN)；\tag{8}$$

$$P=TP/(TP+FP)；\tag{9}$$

$$R=TP/(TP+FN)；\tag{10}$$

$$F_1=2\bullet P\bullet R/(P+R)。\tag{11}$$

式中：$TP$ 为真阳性，预测为正，实际为正；$TN$ 为真阴性，预测为负，实际为负；$FP$ 为假阳性，预测为正，实际为负；$FN$ 为假阴性，预测为负，实际为

正；$N$ 为类别总数。

### 3.2 环境配置

算法实现及实验平台是 Anaconda3-2019.07 (Python3.7.3)+Pycharm-community-2020.3.3，深度学习框架采用 Facebook 推出的 Torch-1.6.0(CPU)，处理器 Intel(R) Core(TM) i5-9500 CPU @ 3.00GHz，实验环境均在离线状态下完成配置。

### 3.3 结果与分析

参数设置：训练次数 Epoch=55，批样本数量 Batch_size=50，学习率 Learning_rate=5e-4，卷积核数量 Filters=256，卷积核尺寸 Filter_size=(2,3,4)，训练集损失函数 Train_Loss，验证集损失函数 Val_Loss，训练集准确率 Train_Acc，验证集准确率 Val_Acc，测试集准确率 Test Acc，测试集文本指令数量 Num。

如图 3 所示，展示了在上述参数下模型的训练结果。如表 2 所示，该模型下呈现了在测试集上 11 种指令意图的 F1 值。对几种典型指令意图的识别结果进行对比，具体识别效果如表 3 所示。

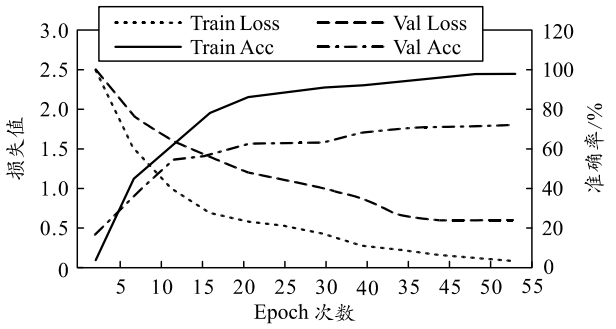


图 3 Text-CNN 训练

表 2 测试集指标评价

| 指令 | F1 | Num | 指令 | F1 | Num |
|---|---|---|---|---|---|
| 开关 | 1.000 | 4 | 报告 | 0.823 | 10 |
| 模式 | 0.857 | 3 | 请求 | 0.941 | 24 |
| 动作 | 0.772 | 65 | 天气 | 1.000 | 3 |
| 状态 1 | 0.484 | 17 | 状态 3 | 0.786 | 15 |
| 同意 | 0.667 | 9 | 查询 | 1.000 | 3 |
| 状态 2 | 0.294 | 29 | | | |
| 合计 | | | | | 182 |

表 3 不同指令意图结果对比

| 文本指令 | 意图结果 | Confidence | 用时/s |
|---|---|---|---|
| 左转航向 270 | 指令-动作 | 0.993 | 0.141 |
| 返回停机坪 | 指令-动作 | 0.963 | 0.253 |
| U001 西安提升到高度 3 000 m 维持 | 指令-动作 | 0.976 | 0.452 |
| 切换控制模式到半自动 | 指令-模式 | 0.936 | 0.429 |
| 当前位置北纬 42°，东经 165° | 指令-状态 3 | 0.857 | 0.757 |
| 回家吃饭 | 无法识别 | 0.614 | 0.046 |

从图 3 可以看出，随着训练次数的增加训练集和验证集的 Loss 值逐渐减小，Acc 值增大。由于数据集较小仅有 2 000 条，在最后 5 次 Epoch 中 Train_Loss 还未达到最小值，而 Val_Loss 基本不变，出现过拟合现象并且很难避免。此问题将在后期优化数据集得以解决。此时是模型最优参数，Train_Acc 和 Val_Acc 分别达到 98%和 73%。

表 2 为测试集 182 条文本指令、11 种指令意图分类的指标评价结果。由表可知，意图种类的数量 Num 分布并不均匀，但基本有明显的特征供模型提取，比如“指令-开关”如“开机、上升”等均由 2 个字组成，“指令-模式”如“切换到自动控制、切换人工模式”都包含模式切换的关键词，其 F1 值分别达到 1 和 0.875。相反，“指令-状态”如“当前位置北纬 42°，东经 165°，时间 0800，高度层 390”描述的是无人机当前位置、高度、时间和特殊状态等，包含信息与其他意图指令耦合较高，没有显著特征，与其他意图相比识别效果欠佳，以上充分体现出 CNN 对于特征提取的能力。另外，部分指令训练集的样本数量相应较少也是其 F1 值较低的原因。

表 3 中，笔者设置置信度(Confidence)值大于 0.8 才属于正确的指令意图。同一指令意图，随着文本的长度增加，其用时也相应增加；“指令-状态 3”，虽然指令文本较短但其特征与其他指令相比不够明显，能够识别但用时较长；当发布“提升高度 3 000 维持”此类非标准空管指令时，系统仍可以得出正确的意图结果；当发布如“回家吃饭”此类无关指令时，系统能够迅速识别并返回提示信息，且不再进行意图识别任务，提高了交互效率，意图识别用时基本保持在 1 s 以内。

综上所述，可知笔者提出的指令意图识别模型能够初步实现对空管员文本指令意图的理解：

1) 面对输入无关指令，该模型设置指令置信度(Confidence=0.8)，低于该阈值则返回错误信息提示，使得其可以面向更多普通用户；

2) 模型中加入歧义词的替换算法，即使发布非标准空管指令，该系统也能较准确地识别出管制意图，降低了用户的指令输入要求，提高了系统的灵活性；

3) 随着文本长度或复杂度的增加，任务用时也相应变长，识别过程需要进行 2 项任务但总用时均保持在 1 s 以内，响应效果良好；

（下转第 59 页）