“智慧政务”中的文本挖掘应用

摘要

随着大数据、云计算、人工智能等技术的发展，建立基于自然语言处理技术的智慧政务系统已经是社会治理创新发展的新趋势，对提升政府的管理水平和施政效率具有极大的推动作用。

针对问题一，对留言进行分类，要首先对附件 2 进行数据的预处理，筛去空值，用数字 id 代替文本标签，对留言详情进行结巴分词以及过滤停用词，计算长文本的平均长度，对超长的进行修剪[1]，而后用 keras 库中的文本训练器建立关于留言内容的一级标签分类模型，进行训练，最后用测试集得出准确率，分析结果。

针对问题二，某一时段内群众集中反映的某一问题可称为热点问题，首先为附件 3 定义合理的热度评价指标，并给出评价结果，如留言的点赞数和反对数，以及相似度匹配高的留言信息热度值也随之增长，按表 1 的格式给出排名前 5 的热点问题，保存为文件“热点问题表.xls”。按表 2 的格式给出相应热点问题对应的留言信息，保存为“热点问题留言明细表.xls”。

针对问题三，我们先用 TF-IDF 来提取留言和回复的特征词，然后再用多维余弦定理来计算其相似度，通过设定阈值来求完整性。我们先对数据集中的留言和回复进行数据预处理，去掉停用词和标点符号，然后用 jieba 分词对留言和回复进行分词处理，接着将分词好的留言和回复进行向量化，把每一条留言与其对应的回复的词语相加，然后一一比较留言和回复中是否有出现该词语，有则为 1，否则为 0，构成 0 和 1 组成的多维向量。每一组对应两条。然后进行 TF-IDF 计算，把向量带进公式中得相似度。

关键词： 数据预处理　　分类　　指标　　评价结果

目录

1 挖掘目标

1.1 挖掘背景

近年来，随着互联网的广泛应用以及微信、微博、市长信箱、阳光热线等网络问政平台的迅速发展，网络投稿逐步成为政府了解民意、汇聚民智、凝聚民气的重要渠道，各类社情民意相关的文本数据量不断攀升，给以往主要依靠人工来进行留言划分和热点整理的相关部门的工作带来了极大挑战。同时，随着大数据、云计算、人工智能等技术的发展，建立基于自然语言处理技术的智慧政务系统已经是社会治理创新发展的新趋势，对提升政府的管理水平和施政效率具有极大的推动作用。

针对问题一，在处理网络问政平台的群众留言时，工作人员首先按照一定的划分体系（参考附件 1 提供的内容分类三级标签体系）对留言进行分类，以便后续将群众留言分派至相应的职能部门处理。目前，大部分电子政务系统还是依靠人工根据经验处理，存在工作量大、效率低，且差错率高等问题。

针对问题二，某一时段内群众集中反映的某一问题可称为热点问题，如“XXX 小区多位业主多次反映入夏以来小区楼下烧烤店深夜经营导致噪音和油烟扰民”。及时发现热点问题，有助于相关部门进行有针对性地处理，提升服务效率。为此需要定义合理的热度评价指标，并给出评价结果，给出排名前 5 的热点问题。

针对问题三，可以分析相关部门对留言的答复意见，从答复的相关性、完整性、可解释性等角度对答复意见的质量给出一套评价方案，从而判断该相关部门的质量，是否积极为人民服务。

1.2 挖掘目标

根据附件 1 里的数据建立起来的关于留言内容的一级标签分类模型，将大大减少人工的投入，提高工作效率。

根据附件 2 里的数据提取出来的 top5 排行榜，可以针对性的对热点问题采取有效的行动。结合附件 3 建立起来的评价方案，为民众带来真正的便利。

2 分析与解题过程

2.1 问题一

2.1.1 数据统计与预处理

(1)首先查看一下我们的数据，这些数据都是来自于互联网公开来源的群众问政留言记录，

我们想要把不同留言信息分到不同的分类中去，且每条数据只能对应 7 个类中的一个类。首先，提取附件 2 里面的两个字段 “一级标签”、“留言详情”，然后统计数据总量为 9210 条留言信息；最长留言信息的长度为 3403；接下来要清洗掉有空值的数据。

（2）我们看到各个类别的数据量不一致，如表 2-1 所示，城乡建设、劳动和社会保障、教育文体的数据量就占了总数的 2/3，而交通运输的数据量最少只有 613 条，分布很不均匀。

表 2-1：各类别的数据量

一级标签	count
城乡建设	2009
劳动和社会保障	1969
教育文体	1589
商贸旅游	1215
环境保护	938
卫生计生	877
交通运输	613

（3）接下来我们要将“一级标签”类转换成 id，如表 2-2 所示，这样便于以后的分类模型的训练。

表 2-2：一级标签转换

一级标签	cat_id
城乡建设	0
劳动和社会保障	1
教育文体	2
商贸旅游	3
环境保护	4
卫生计生	5
交通运输	6

（4）我们将“一级标签”类转换成了 id（0 到 6），由于我们的评价内容都是中文，所以要对中文进行一些预处理工作[2] [3]，这包括删除文本中的标点符号，特殊符号，还要删除一些无意义的常用词（stopword），中文停用词包含了很多日常使用频率很高的常用词，如：吧、吗、呢、啥等一些感叹词等，这些高频常用词无法反应出文本的主要意思，所以要被过滤掉。因为这些词和符号对系统分析预测文本的内容没有任何帮助，反而会增加计算的复杂度和增加系统开销，所有在使用这些文本数据之前必须要将它们清理干净。我们过滤掉了“留言详情”中的标点符号和一些特殊符号，并生成了一个新的字段。接下来我们要在新字段的基础上进行分词，把每个评论内容分成由空格隔开的一个一个单独的词语。

2.1.2 LSTM 建模

（1）数据预处理完成以后，接下来我们要开始进行 LSTM 的建模工作[4]：我们要将停用词过滤以及分词后的数据进行向量化处理,我们要将每条留言信息转换成一个整数序列的向量。

注意我们只选择使用前 50000 个使用频率最高的词，在这个预训练词向量模型中，一共有 260 万词汇量，如果全部使用在分类问题上会很浪费计算资源，因为我们的训练样本很小，一共只有 9k，如果我们有 100k，200k 甚至更多的训练样本时，在分类问题上可以考虑减少使用的词汇量。

（2）每一串索引的长度并不相等，如图 2-1 所示，呈正态分布，所以为了方便模型的训练我们需要把索引的长度标准化[5]，上面我们选择了 403 这个可以涵盖 95%训练样本的长度，设置每条留言信息最大的词语数为 250 个，接下来我们进行 padding，超过的将会被截去，不足的将会被补 0。我们一般采用'pre'的方法，这会在文本索引的前面填充 0，因为根据一些研究资料中的实践，如果在文本索引后面填充 0 的话，会对模型造成一些不良影响。

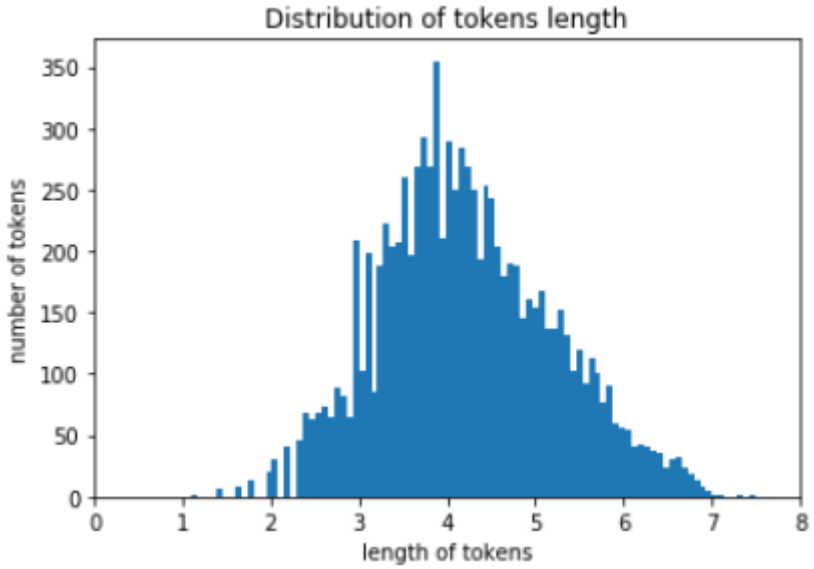

图 2-1：留言信息长度呈正态分布

（3）下面我们要拆分训练集和测试集好以后，我们要定义一个 LSTM 的序列模型：模型的第一次是嵌入层(Embedding)，它使用长度为 100 的向量来表示每一个词语。SpatialDropout1D 层在训练中每次更新时，将输入单元的按比率随机设置为 0，这有助于防止过拟合。LSTM 层包含 100 个记忆单元，输出层为包含 7 个分类的全连接层，由于是多分类，所以激活函数设置为'softmax'，而损失函数为分类交叉熵 categorical_crossentropy。

（4）定义好 LSTM 模型以后，我们要开始训练数据：设置 10 个训练周期，batch_size 为 128。下面我们画损失函数趋势图和准确率趋势图，如图 2-2、图 2-3 所示：

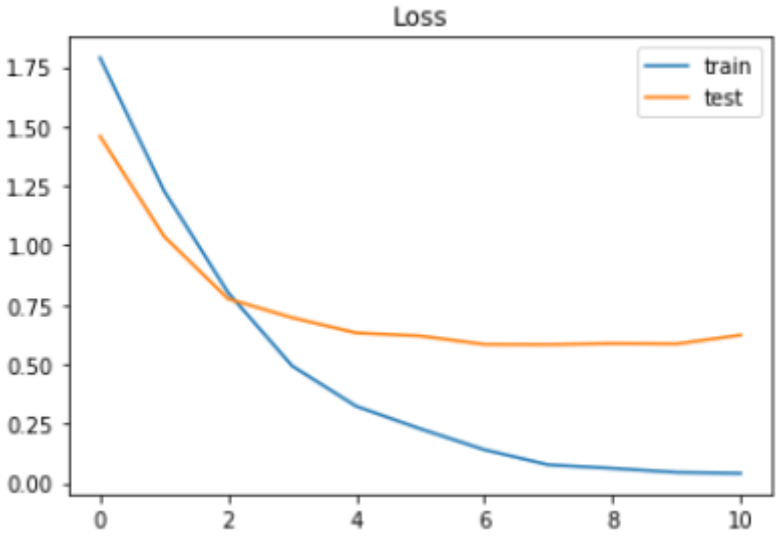

图 2-2：损失函数趋势图

从上图中我们可以看见，随着训练周期的增加，模型在训练集中损失越来越小，这是典型的过拟合现象，而在测试集中，损失随着训练周期的增加由一开始的从大逐步变小,再逐步变大。

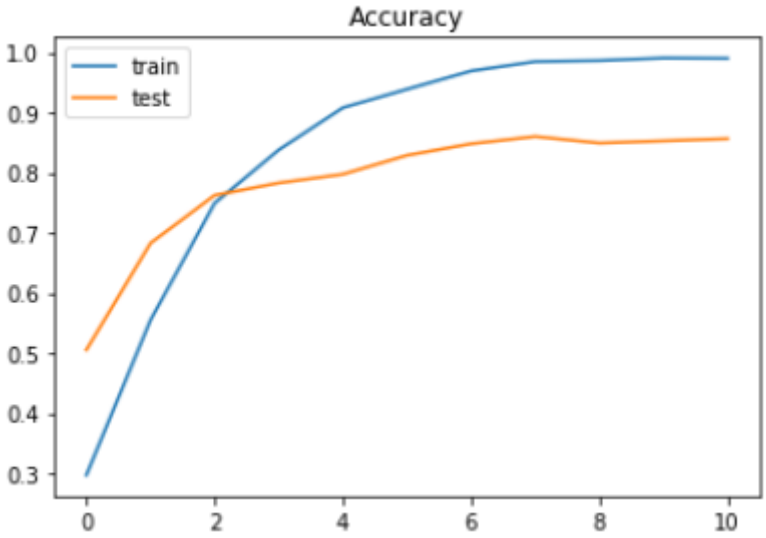

图 2-3：准确率趋势图

从上图中我们可以看见，随着训练周期的增加，模型在训练集中准确率越来越高，这是典型的过拟合现象，而在测试集中，准确率随着训练周期的增加由一开始的从小逐步变大，再逐步变小，有点波动。

2.1.3 模型评估

我们可以借助 sklearn 库中的 metrics 模块快速对该模型的结果进行监控指标，可以得到下图统计数据，如表 2-3 所示：

表 2-3：各个类别的评价

	Precision	Recall	F1-score	Support
城乡建设	0.81	0.82	0.82	200
环境保护	0.91	0.87	0.89	102
交通运输	0.77	0.59	0.67	73
教育文体	0.92	0.89	0.90	170
劳动和社会保障	0.87	0.91	0.89	172
商贸旅游	0.69	0.75	0.72	114
卫生计生	0.82	0.88	0.85	90

Accuracy			0.83	921
Macro avg	0.83	0.82	0.82	921
Weighted avg	0.84	0.88	0.83	921

2.1.4 自定义预测

首先我们定义一个预测函数，对输入的文本进行停用词分词等预处理，最后调用这个预测函数，判别它的类目。

```
predict('家里房子总是漏水。')
```

```
'城乡建设'
```

```
predict('今年五一回家路上堵了好多次车。')
```

```
'交通运输'
```

图 2-4：自定义预测

2.2 问题二

2.2.1 数据统计与预处理

（1）根据'附件 3.xlsx'信息表，统计共有 4326 条留言信息，其中最高赞数高达 2097 票，而反对数最多 53 票，这是合理的指标之一，此外还需统计涵盖相同信息点的留言信息，成为另外一个指标。

（2）设置热度值初始为 1，鉴于点赞数高出反对数如此之多，可把点赞数（或反对数）按一定的比例加进（或减去）热度值，通过排序得出点赞数多的排在前头，反对数多的排在队尾。

2.2.2 计算 TF 系数

（1）TF [6]：对于一条留言 dj ，里面的词语 ti 来说，ti 的词频可表示为

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

相当于:

$$TF_{\mathrm{w}} = \frac{\text{在某一类中词条}w\text{出现的次数}}{\text{该词条中所有的词条数目}} \tag{2}$$

（2）IDF [7]：某个词语对于整个语料库的重要性的度量。也就是所有数据集的词相对于一个词的度量。

$$\mathrm{idf}_{\mathrm{i}} = \lg \frac{|D|}{|j : t_i \in d_j| + 1} \tag{3}$$

$$IDF = \log \frac{\text{语料库的文档总数}}{\text{包含词条}w\text{的文档数} + 1} \tag{4}$$

（3）TF-IDF=TF * IDF

某一特定留言内的高词语频率，以及该词语在整个数据集合中的低留言频率，可以产生出高权重的 TF-IDF。因此，TF-IDF 倾向于过滤掉常见的词语，保留重要的词语。

2.2.3 生成热点问题表

（1）利用 sklearn 包中的 CountVectorizer 库进行文本相似度匹配，数据间每匹配到与自己相似的留言信息，则叠加一定数值代表热度增加，最后通过排序，可得到热度值排名前五的热点问题。

（2）要将上面得到的 top5 热点问题保存为新的表，首先 pandas 中的 DataFrame 建立一个空表，逐步加上该有的列属性，如：热度排名、问题 ID、热度指数、时间范围、问题描述。

（3）热度排名和问题 ID 分别与热度值排名前五的热点问题从 1~5 一一对应。

（4）时间范围是由 top5 的问题描述分别去原附件 3 里匹配相似度达到一定标准的留言信息的时间，获取全部相似问题的时间后，通过排序，分别取队头和队尾，转换为字符串覆盖原有的时间 [8]。

（5）最后用 pandas 导出这个新建的表格，命名“热点问题表.xlsx”（详见附录）。

2.2.4 生成热点问题留言明细表

（1）第一步同样是先建立一个空表，随后在附件 3 中添加一新属性“问题 ID”，初始为 0。

（2）根据上边 top5 设置的问题 ID（对应 1~5），用相似度匹配分别去附件 3 里筛选，达到标准的将其属性问题 ID 赋值为对应的数字，并将其一整行数据添加进新建的空表内。

（3）最后用 pandas 导出这个新建的表格，命名“热点问题留言明细表.xlsx”（详见附录）。

2.3 问题三

2.3.1 数据预处理

（1）先对数据集中的留言和回复进行读取存为列表，然后加载停用词表去掉停用词和标点符号。然后用 jieba 分词对留言和回复进行分词处理。

（2）TF-IDF 权重计算：把留言和回复进行词语数量统计，统计每一条中词语的个数，以及词性，然后组成一个二维的列表。对其进行 TF 和 IDF 计算，得到两个列表数据，相乘得 TF-IDF 数值。

2.3.2 数据向量化

样本转化为对应的空间中的两个向量。先将分词好的留言和回复进行向量话，把每一条留言与其对应的回复的词语相加，然后一一比较留言和回复中是否有出现该词语，有则为 1，否则为 0，构成 0，1 多维向量。每一组对应两条。

例如：

s1 = "乌干达外交部就此事件可能对中国大使馆造成的负面影响感到抱歉。",

s2 = "此事件对中国大使馆造成了一定的负面影响，乌干达外交部感到抱歉并公开道歉。"

X = (1,1,1,1,1,1,1,1,1,1,1,0,1,0,1,0,0)

Y = (0,1,1,0,1,1,1,0,1,1,0,1,1,1,1,1,1)

2.3.3 利用余弦定理计算文本相似度

（1）把向量化的留言和回复两两进行计算：

留言中出现的字为：Z1c1,Z1c2,Z1c3,Z1c4……Z1cn；它们在章节中的个数为：Z1n1,Z1n2,Z1n3……Z1nm;

回复中出现的字为：Z2c1,Z2c2,Z2c3,Z2c4……Z2cn；它们在章节中的个数为：Z2n1,Z2n2,Z2n3……Z2nm;

其中，Z1c1 和 Z2c1 表示两个文本中同一个字，Z1n1 和 Z2n1 是它们分别对应的个数，最后进行相似度计算 [9]：

$$SimilaryValue=\frac{(Z1n1\times Z2n1)+(Z1n2\times Z2n2)+(Z1n3\times Z2n3)+.....+(Z1nn\times Z2nn)}{\sqrt{Z1n1^2+Z1n2^2+Z1n3^2.....+Z1nn^2}+\sqrt{Z2n1^2+Z2n2^2+Z2n3^2.....+Z2nn^2}} \quad (5)$$

(2)设置阈值来剔除相似性过低的值,同时用提出的数量和原来的数量相除得出其完整性。

3 结果分析

3.1 问题一结果分析

(1) 索引长度标准化

因为每段留言信息的长度是不一样的，我们如果单纯取最长的一个评语，并把其他评填充成同样的长度，这样十分浪费计算资源，所以我们取一个折中的长度。

3.2 问题二结果分析

(1) 关于点赞数的占比，不同人可能会有不同的看法，因为个别问题的点赞数很高，考虑到动动手指头点个赞可能只是随意为之，而相比花费时间写留言信息的民众，后者就显得更加有诚意。如果不注重投稿人的心思，排行榜上明显都是点赞数最高的那几个，而其他不被人同情、却又一部分人反映的问题留言则会被挤到后面，写再多、投再多的留言信息也会被忽略，所以在加大留言数的权重后，得到的结果如图 3-1 所示，靠点赞数支撑 top5 的有三个，靠留言数量挤进 top5 有两个，功夫不负有心人，还算满意。

```
#显示top5
data.head()
```

	留言编号	留言用户	留言主题	留言时间	留言详情	反对数	点赞数	count
0	208636.0	A00077171	A市A5区汇金路五矿万境K9县存在一系列问题	2019/8/19 11:34:04	\n\t\t\t\t\t\n\t\t\t\t\t\t我是A市A5区汇金路五矿万境K9县24栋的一...	0.0	2097.0	630.1
1	223297.0	A00087522	反映A市金毛湾配套入学的问题	2019/4/11 21:02:44	\n\t\t\t\t\t\n\t\t\t\t\t\t书记先生：您好！我是梅溪湖金毛湾的一名业主，...	5.0	1762.0	555.6
2	284192.0	A00081652	咨询A市楚税社保审核问题	2019/11/17 13:44:57	\n\t\t\t\t\t\n\t\t\t\t\t\t尊敬的领导，我在楚税社保上交了4个人的农村合...	0.0	0.0	265.0
3	239648.0	A909211	A市A2区丽发新城小区附近搅拌站明目张胆污染环境	2020-01-06 22:41:31	丽发新城小区附近近日突然建起了搅拌厂！特别扰民，机器一天到晚的响，吵得人不得休息，还有灰尘颗...	0.0	0.0	261.6
4	220711.0	A00031682	请书记关注A市A4区58车贷案	2019/2/21 18:45:14	\n\t\t\t\t\t\n\t\t\t\t\t\t尊敬的胡书记：您好!A4区p2p公司58车贷...	0.0	821.0	257.3

图 3-1：初步统计出来的 top5

(2) 在相似度的匹配中，采取的是冒泡算法，但是测试一开始的小数据还可以，后面数据一增多就电脑就跑不动。既然小数据跑得动，那就利用“分而治之”的思想，将大数据分为多个小数据，最后再合并汇总，再次匹配相似度，叠加留言信息的热度值，最终达到预期效果。但是有一个需要注意的地方，因为一开始的思路是两两匹配相似度，你分的数据块越多，则准确率会微微下降，而如果分的数据块太大，运行时间会变长，所以不断的调参，选出一个最佳值。

(3) 此外，为了避免数据过多，设置留言信息在一个范围内进行相似度匹配后的热度值没有很大进展的话，则认为其不是“潜力股”，可以将其过滤，减小表格长度。

（4）最后，为了使热点问题留言明细表更加准确，我在前面数据预处理对留言主题进行了分词和自定义的停用词处理，要观察数据以及多次运行的结果，一点点积累起来的自定义停用词 txt，与网上下载的 stopword.txt 有很大不同，具有更加明显的针对性。

3.3 问题三结果分析

（1）我通过去除一些名词来可以对结果的精确度进行提高，比如像 n 这种（ '物业管理', '市', '区', '街道', '区', '公安分局', '乱象', '小区'）是不能区分哪个哪个小区或地方的去掉后其相似性就会提高，还有'x'这种，虽然相似性会高，但其实是空格和退格键，所以不能当作特征词。
（2）相似性低可能是因为数据量太少，但我们可以由此得知剔除那些词性特征词来提高相似度。

4 结论

收集自互联网公开来源的群众问政留言记录，及相关部门对部分群众留言的答复意见。利用自然语言处理和文本挖掘的方法进行分析研究，了解社会群众和相关部门的需求特点与趋势，对国家和社会的发展有重大意义，同时也是文本分析的一个课题、一个难题。

随着互联网的快速发展，传统的文本解读以及不能满足数据量庞大的网络问政平台信息，本文采用基于 LSTM 多分类模型，统计群众留言的标签分类，给工作人员减少了负担。

由分析结果可以看出，热点问题应该及时安排相关人员处理，避免给民众带来不必要的麻烦，争取做到为人民服务；最后的评价方案，不仅能做到公平公正，还能督促相关部门做好自己的本分工作，为社会做出实质的贡献。

基于 LSTM 分类模型的优点是，在训练的时候自带进度条，而且具有 callbacks 功能，避免了资源的过度浪费，而且可以生成各种监控指标，再通过损失函数和准确率趋势图，调节合适的超参，以达到更优的效果；缺点是基于深度学习的模型共性的特点，运行时间会随着 epochs、batch_size 的增加而变长。

5 参考文献

[1] 基于深度学习的自然语言处理（NLP Based on Deep Learning）, JinTa Weng,2019.
[2] 文本挖掘和可视化案例：基于文本内容的垃圾短信分类_人邮版 ,泰迪杯云课堂,2020.
[3] 案例:电商产品评论数据情感分析（Python）,杨惠,2020.
[4] 链接:https://blog.csdn.net/weixin_42608414/article/details/89856566,该文为博主原创文章,2019-05-06.
[5] github:https://github.com/aespresso/chinese_sentiment, aespresso, 2018-08-28.
[6] 链接: https://www.jianshu.com/p/edad714110fb ,该文为博主原创文章, 2019.06.23.
[7] 链接: https://blog.csdn.net/Turbo_Come/article/details/89455008 ,该文为博主原创文章,

2019-04-22.
[8] 链接: https://blog.csdn.net/spartanfuk/article/details/82051656 ,该文为博主原创文章, 2018-08-25.
[9] 链接: https://www.cnblogs.com/yytxdy/p/11836740.html ,该文为博主原创文章, 2019-11-11.

附录

1．建模所使用的软件: Jupyter Notebook 、tensorflow1.12.0 、keras2.2.3 、 sklearn 等库。
2．LSTM 模型（其余代码见附件）:

```
model = Sequential()
model.add(Embedding(MAX_NB_WORDS, EMBEDDING_DIM,
                    input_length=X.shape[1]))
model.add(SpatialDropout1D(0.2))
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(7, activation='softmax'))
model.compile(loss='categorical_crossentropy',
                    optimizer='adam', metrics=['accuracy'])
print(model.summary())
epochs = 12
batch_size = 128
history = model.fit(X_train, Y_train,epochs=epochs,
                        batch_size=batch_size,validation_split=0.1,
                        callbacks=[EarlyStopping(monitor='val_loss',
                        patience=3, min_delta=0.0001)])
accr = model.evaluate(X_test,Y_test)
print('Test set\n  Loss: {:0.3f}\n  Accuracy: {:0.3f}'.format(accr[0],accr[1]))
```