自然语言处理第二次作业

陈远腾 2020k8009929041

一、环境搭建

1.新建conda环境

```
conda create -n NLP python=3.7 anaconda
conda activate NLP
```

2.安装tensorflow_gpu

(安装cuda+cudnn见网上教程)

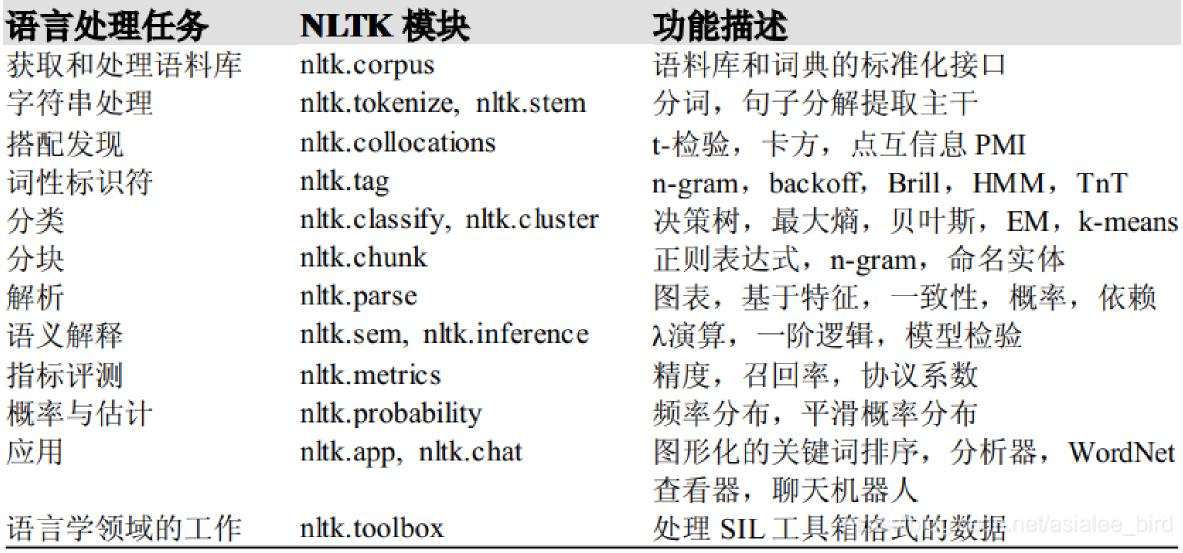
```
conda install tensorflow
```

3.所需其它依赖库

- re
- argparse
- csv
- numpy
- nltk

4.nltk_data安装

nltk是一套基于python的自然语言处理工具集，主要功能如下：



语言处理任务	NLTK 模块	功能描述
获取和处理语料库	nltk.corpus	语料库和词典的标准化接口
字符串处理	nltk.tokenize，nltk.stem	分词，句子分解提取主干
搭配发现	nltk.collocations	t-检验，卡方，点互信息 PMI
词性标识符	nltk.tag	n-gram，backoff，Brill，HMM，TnT
分类	nltk.classify，nltk.cluster	决策树，最大熵，贝叶斯，EM，k-means
分块	nltk.chunk	正则表达式，n-gram，命名实体
解析	nltk.parse	图表，基于特征，一致性，概率，依赖
语义解释	nltk.sem，nltk.inference	λ演算，一阶逻辑，模型检验
指标评测	nltk.metrics	精度，召回率，协议系数
概率与估计	nltk.probability	频率分布，平滑概率分布
应用	nltk.app，nltk.chat	图形化的关键词排序，分析器，WordNet 查看器，聊天机器人
语言学领域的工作	nltk.toolbox	处理 SIL 工具箱格式的数据

nltk库的安装比较麻烦，所以这里单独说明：

首先需要 pip install nltk, 然后需要下载nltk具体的packages：

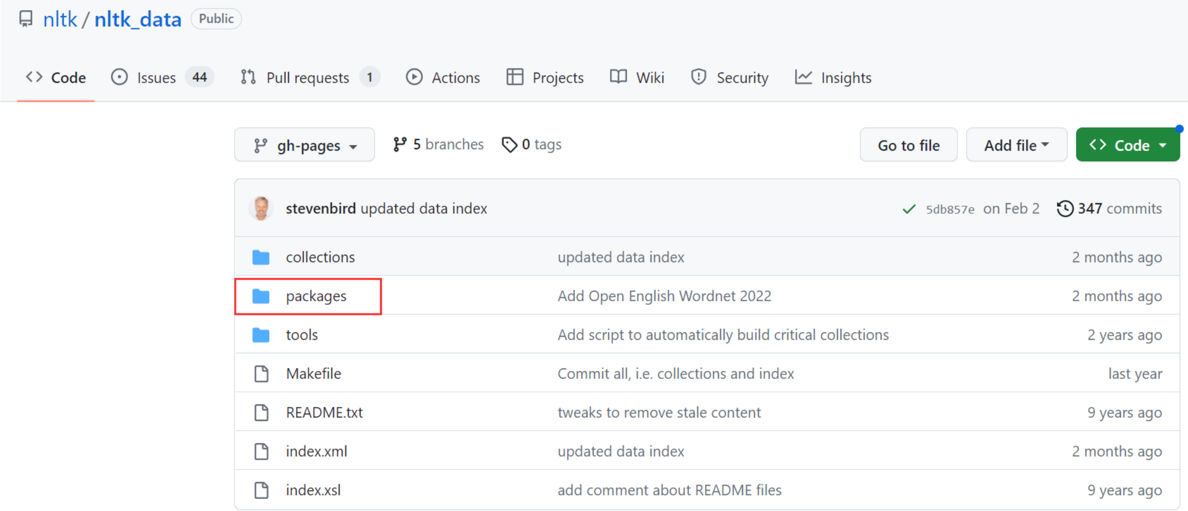
官方给出的方法是先进入python环境，然后运行：

```
import nltk
nltk.download()
```

但实测这样很难下载成功。替代方案是去github上的nltk主页

NLTK_github主页

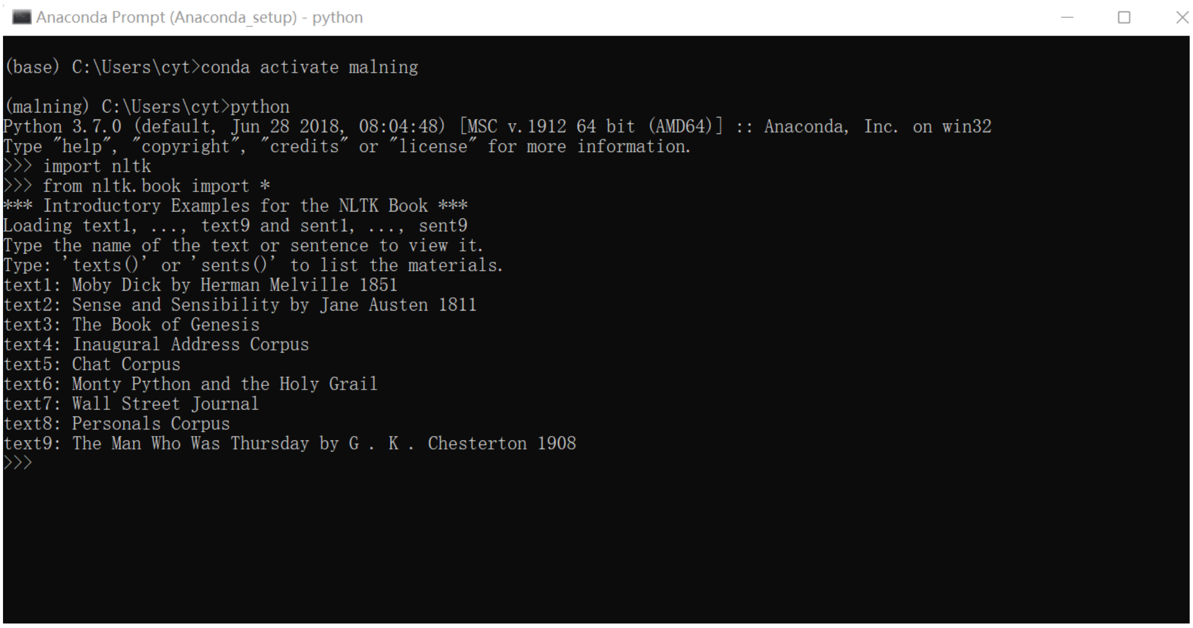
下载其中的packages文件夹，下载后将其解压在D盘的根目录下，并将文件夹改名为nltk_data



stevenbird updated data index ✓ 5db857e on Feb 2 347 commits

collections	updated data index	2 months ago
packages	Add Open English Wordnet 2022	2 months ago
tools	Add script to automatically build critical collections	2 years ago
Makefile	Commit all, i.e. collections and index	last year
README.txt	tweaks to remove stale content	9 years ago
index.xml	updated data index	2 months ago
index.xsl	add comment about README files	9 years ago

安装成功的检测方法：



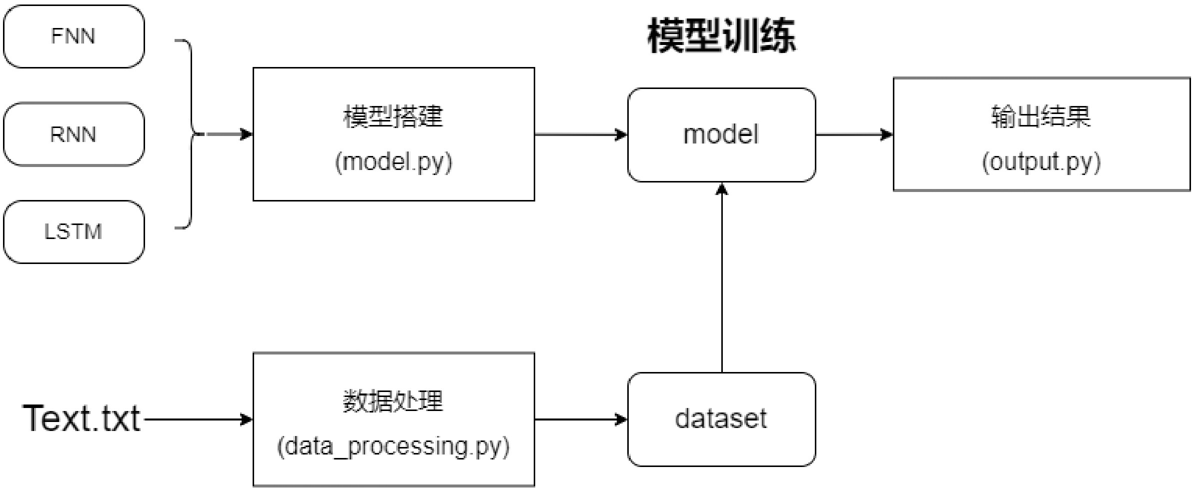
```
(base) C:\Users\cyt>conda activate malning

(malning) C:\Users\cyt>python
Python 3.7.0 (default, Jun 28 2018, 08:04:48) [MSC v.1912 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import nltk
>>> from nltk.book import *
*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
>>>
```

正常显示出text1-9则说明nltk_data安装成功。

(如果安装还是不成功，可以参考csdn，nltk_data安装的教程很多)

二、实验过程

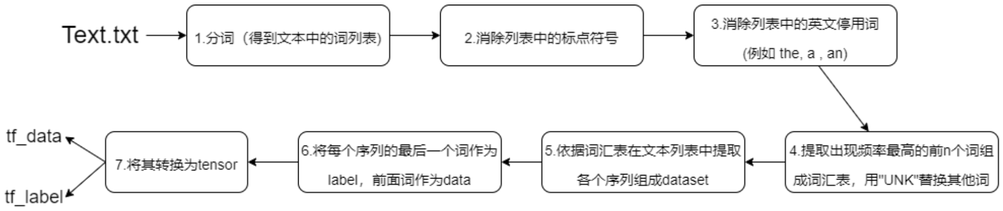
实验过程主要分为数据处理、模型训练和结果分析及输出三部分：



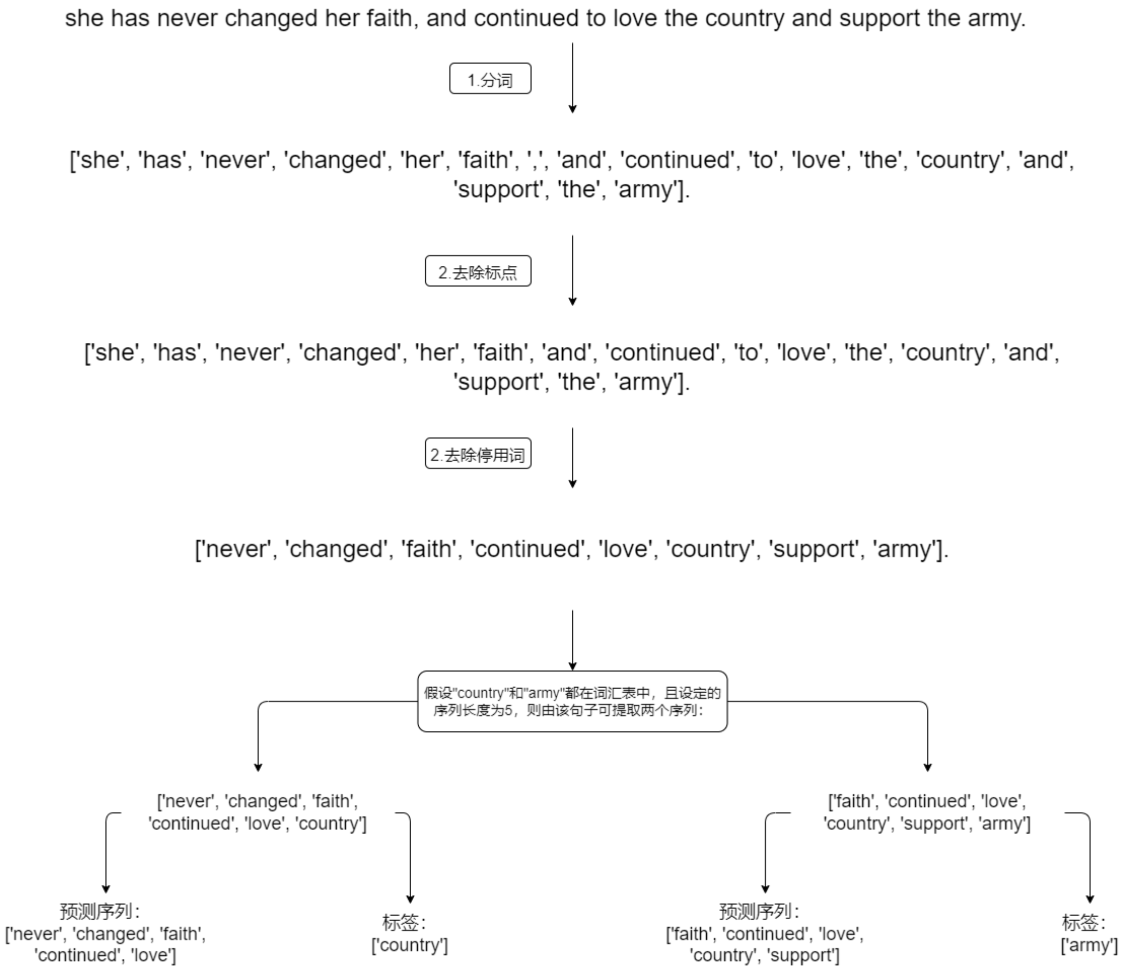
下面依次介绍这三部分内容：

1.数据处理

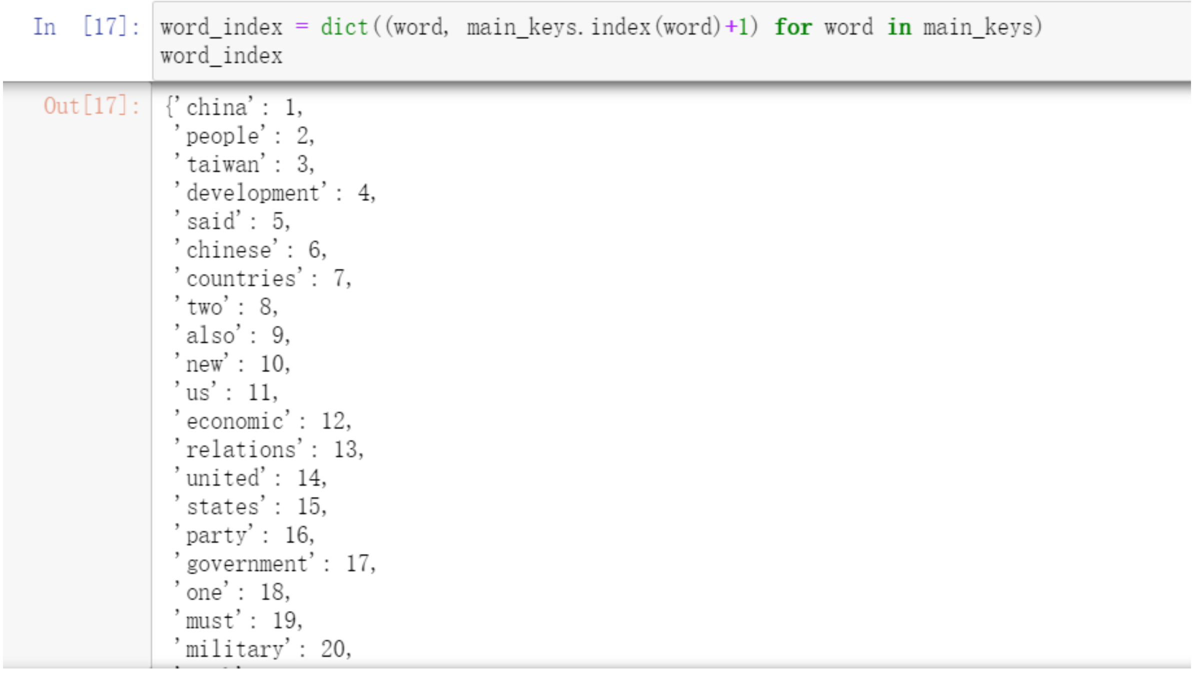
数据处理模块的输入为原始的英文文本，输出为序列化的dataset：



以处理以下文本为例：


she has never changed her faith, and continued to love the country and support the army.
1.分词
['she', 'has', 'never', 'changed', 'her', 'faith', ',', 'and', 'continued', 'to', 'love', 'the', 'country', 'and', 'support', 'the', 'army'].
2.去除标点
['she', 'has', 'never', 'changed', 'her', 'faith', 'and', 'continued', 'to', 'love', 'the', 'country', 'and', 'support', 'the', 'army'].
2.去除停用词
['never', 'changed', 'faith', 'continued', 'love', 'country', 'support', 'army'].
假设"country"和"army"都在词汇表中，且设定的序列长度为5，则由该句子可提取两个序列：
['never', 'changed', 'faith', 'continued', 'love', 'country']
['faith', 'continued', 'love', 'country', 'support', 'army']
预测序列：
['never', 'changed', 'faith', 'continued', 'love']
标签：
['country']
预测序列：
['faith', 'continued', 'love', 'country', 'support']
标签：
['army']

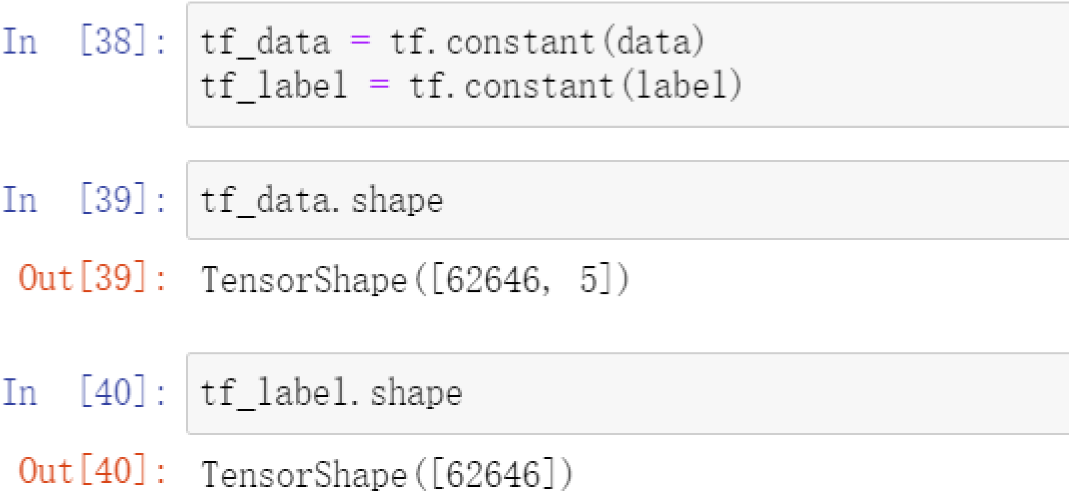
实际处理中，我们首先需要将词汇表中的每一个词映射为一个值，编码后才能进行训练，如下：



```
In  [17]: word_index = dict((word, main_keys.index(word)+1) for word in main_keys)
          word_index

Out[17]: {'china': 1,
          'people': 2,
          'taiwan': 3,
          'development': 4,
          'said': 5,
          'chinese': 6,
          'countries': 7,
          'two': 8,
          'also': 9,
          'new': 10,
          'us': 11,
          'economic': 12,
          'relations': 13,
          'united': 14,
          'states': 15,
          'party': 16,
          'government': 17,
          'one': 18,
          'must': 19,
          'military': 20,
```

依据word_index，将序列中的每个词转换为对应数字，词汇表外的词映射为0。

最终经过以上过程处理，得到6w+条训练序列：



```
In  [38]: tf_data = tf.constant(data)
          tf_label = tf.constant(label)

In  [39]: tf_data.shape

Out[39]: TensorShape([62646, 5])

In  [40]: tf_label.shape

Out[40]: TensorShape([62646])
```

这些序列与标签后续将送入模型进行训练。

2.模型搭建

本次实验中，基于tensorflow2.0搭建了FNN, RNN, LSTM三个model，这三个网络的第一层均为embedding层，输出层均为（Num_keys + 1)个单元的Dense层，区别仅在于中间层为：

- FNN：Flatten() + Dense(Num_kernels)
- RNN：SimpleRNN(Num_kernels)
- LSTM：LSTM(Num_kernels)

（1）layers.embedding层

由于这三个模型中的第一层均为embedding层，且这是实现词向量嵌入的关键，故我们单独进行介绍：

```
#layers定义
tf.keras.layers.Embedding(
    input_dim, output_dim, embeddings_initializer='uniform',
    embeddings_regularizer=None, activity_regularizer=None,
    embeddings_constraint=None, mask_zero=False, input_length=None, **kwargs
)
```

参数说明：(介绍中省略了我们使用默认值的参数)

- input_dim：词汇表的大小，即最大整数索引+1。
- output_dim：稠密嵌入的维数。
- imput_length：输入序列的长度。

输入：(batch_size, input_length)的二维张量。

输出：(batch_size, input_length, output_dim)的三维张量。

同时最重要的是，embedding层的参数即为最终欲求的词向量，

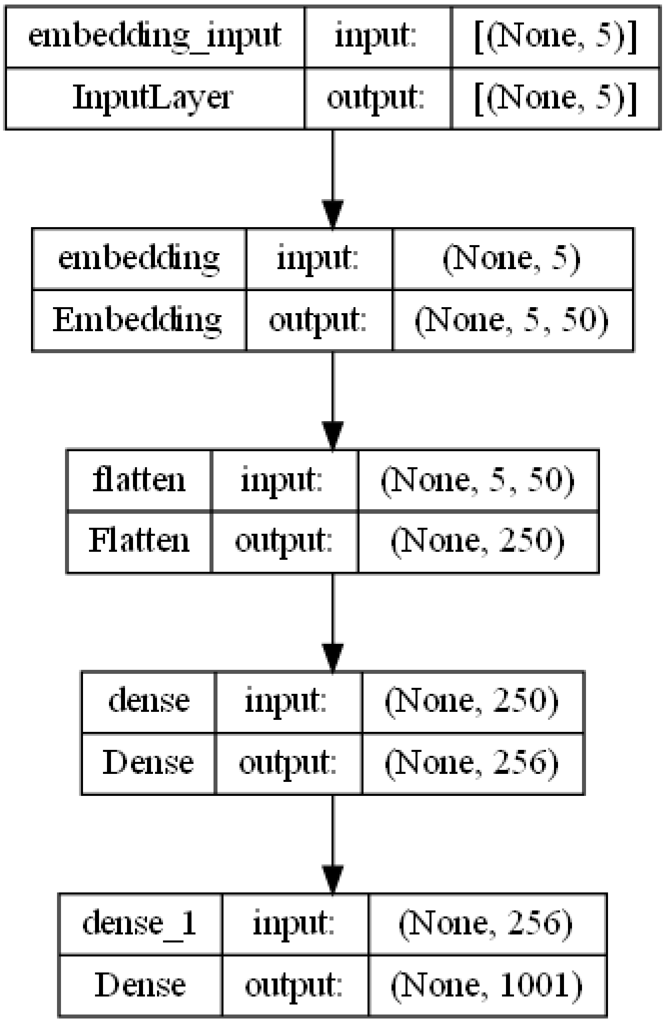
其形状为(input_dim, output_dim)，也即(词汇表长度 x 词向量嵌入维数)。

故在训练model后，我们只要提取embedding层的weights即可得到最终的词向量。

(2) FNN model

```
# model1: FNN model
def Build_FNN_model(Num_keys, Vector_len, Input_len, Num_kernels):
    """

    :param Num_keys: 词汇表长度
    :param Vector_len: 单个词向量的长度
    :param Input_len: 单个输入向量的长度
    :param Num_kernels: Hidden Layer Dense层的核数
    :return: model
    """

    model = keras.Sequential()
    # Enbedding层将Num_keys个词汇嵌入到Num_keys个长度为Vector_Len的向量中
    model.add(layers.Embedding(Num_keys+1, Vector_len, input_length=Input_len))
    # FNN模型需要先将Enbedding的输入结果铺平（将向量展开）
    model.add(layers.Flatten())
    """
    Enbedding_output:  (Batch_size , Input_len , Vector_len))
                                           |
                                           |  layers.Flatten()
                                           V
    Flatten_output:    (Batch_size , (Input_len x Vector_len))
    """
    #添加隐藏层
    model.add(layers.Dense(Num_kernels))
    #输出层，多分类激活函数使用softmax
    model.add(layers.Dense(Num_keys+1, activation='softmax'))
    return model
```

利用pydot绘图如下：


embedding_input | input: | [(None, 5)]
InputLayer | output: | [(None, 5)]
embedding | input: | (None, 5)
Embedding | output: | (None, 5, 50)
flatten | input: | (None, 5, 50)
Flatten | output: | (None, 250)
dense | input: | (None, 250)
Dense | output: | (None, 256)
dense_1 | input: | (None, 256)
Dense | output: | (None, 1001)

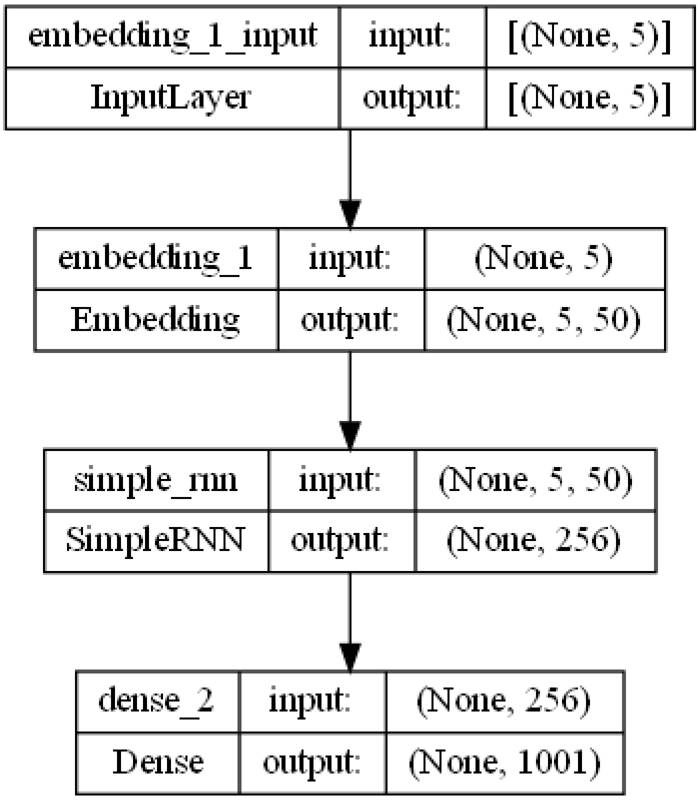
值得注意的有两点：

- 一是这里的词汇表长度为(Num_keys + 1)的原因是“UNK"也在词汇表中。
- 在FNN模型中，由于后续没有处理序列的RNN单元，故需要一个Flatten层将输入的向量序列铺平，才能够输入到后续的Dense层中。

(2) RNN model

```
# model2: RNN model
def Build_RNN_model(Num_keys, Vector_len, Input_len, Num_kernels):
    """

    :param Num_keys: 词汇表长度
    :param Vector_len: 单个词向量的长度
    :param Input_len: 单个输入向量的长度
    :param Num_kernels: RNN层的核数
    :return: model
    """

    model = keras.Sequential()
    # Embedding层将Num_keys个词汇嵌入到Num_keys个长度为Vector_Len的向量中
    model.add(layers.Embedding(Num_keys+1, Vector_len, input_length=Input_len))
    #添加RNN层
    model.add(layers.SimpleRNN(Num_kernels))
    #输出层，使用"softmax"作为激活函数
    model.add(layers.Dense(Num_keys+1, activation='softmax'))

    return model
```

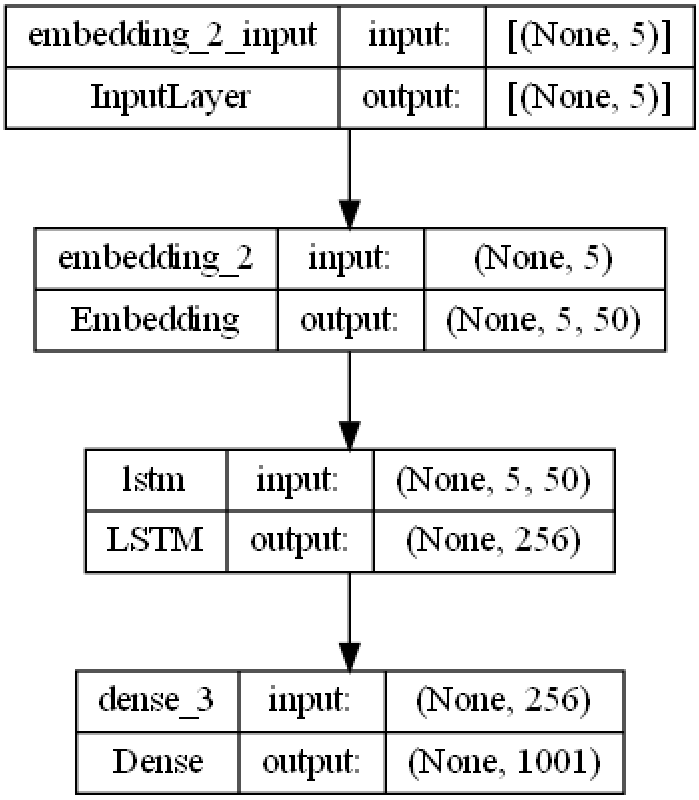
利用pydot绘图如下：



(3) LSTM model

```
# model3: LSTM model
def Build_LSTM_model(Num_keys, Vector_len, Input_len,  Num_kernels):
    """

    :param Num_keys: 词汇表长度
    :param Vector_len: 单个词向量的长度
    :param Input_len: 单个输入向量的长度
    :param Num_kernels: LSTM层的核数
    :return: model
    """

    model = keras.Sequential()
    #Enbedding层将Num_keys个词汇嵌入到Num_keys个长度为Vector_Len的向量中
    model.add(layers.Embedding(Num_keys+1, Vector_len, input_length =
Input_len))
    #LSTM层
    model.add(layers.LSTM(Num_kernels))
    #输出层，使用"softmax"作为激活函数
    model.add(layers.Dense(Num_keys+1, activation='softmax'))

    return model
```

利用pydot绘图如下：



(4) 模型编译

```
# compile model
# 由于三个模型使用的优化器、损失函数及评估方法相同，故定义同一个编译函数

def Compile_model(model):
    """
    优化器：adam
    损失函数：交叉混合熵
    (由于label是直接编码而非独热码，故使用"sparse")
    评估：acc

    :param model: 搭建的模型
    :return: 编译后的模型
    """

    model.compile(optimizer='adam',
             loss='sparse_categorical_crossentropy',
             metrics=['acc'])

    return model
```

- 优化器：Adam优化器
- 损失函数：'sparse_categorical_crossentropy'，因为我们这里的标签采取的是直接编码而非独热码，所以需要使用"sparse"。
- 评价指标：accuracy

(5) 模型训练

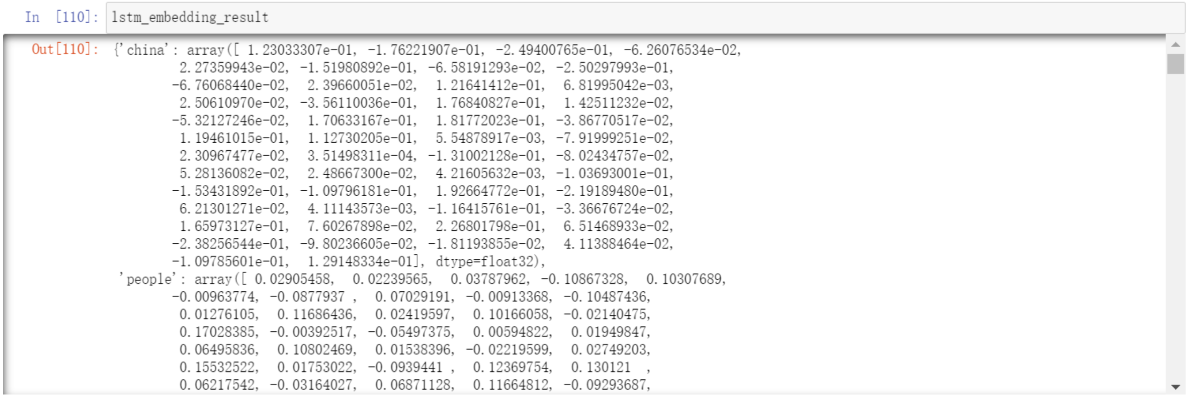
```
# train model
def Train_model(model, tf_data, tf_label, epochs, batch_size, validation_split):
    """

    :param model: 编译过的模型
    :param tf_data: 训练数据(向量序列)
    :param tf_label: 标签集
    :param epochs: 训练轮数
    :param batch_size: batch_size
    :param validation_split: 训练集/测试集划分 (0.2意味20%训练集)
    :return: 训练过的模型
    """

    model.fit(tf_data, tf_label, epochs=epochs, batch_size=batch_size,
validation_split=validation_split)
    return model
```

3.结果输出

前面我们提到过，训练过后的model的embedding层的weights即为词汇表的词向量。

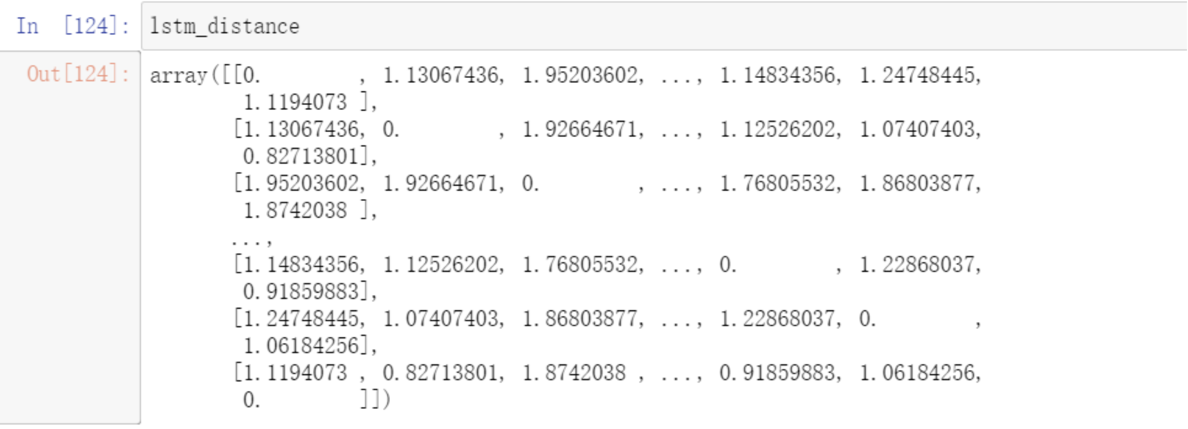
(1) 词向量提取+计算距离矩阵

- step1：将embedding层的weights提取出来并与词汇表结合制作字典。



```
In  [110]: lstm_embedding_result

Out[110]: {'china': array([ 1.23033307e-01, -1.76221907e-01, -2.49400765e-01, -6.26076534e-02,
                  2.27359943e-02, -1.51980892e-01, -6.58191293e-02, -2.50297993e-01,
                 -6.76068440e-02,  2.39660051e-02,  1.21641412e-01,  6.81995042e-03,
                  2.50610970e-02, -3.56110036e-01,  1.76840827e-01,  1.42511232e-02,
                 -5.32127246e-02,  1.70633167e-01,  1.81772023e-01, -3.86770517e-02,
                  1.19461015e-01,  1.12730205e-01,  5.54878917e-03, -7.91999251e-02,
                  2.30967477e-02,  3.51498311e-04, -1.31002128e-01, -8.02434757e-02,
                  5.28136082e-02,  2.48667300e-02,  4.21605632e-03, -1.03693001e-01,
                 -1.53431892e-01, -1.09796181e-01,  1.92664772e-01, -2.19189480e-01,
                  6.21301271e-02,  4.11143573e-03, -1.16415761e-01, -3.36676724e-02,
                  1.65973127e-01,  7.60267898e-02,  2.26801798e-01,  6.51468933e-02,
                 -2.38256544e-01, -9.80236605e-02, -1.81193855e-02,  4.11388464e-02,
                 -1.09785601e-01,  1.29148334e-01], dtype=float32),
           'people': array([ 0.02905458,  0.02239565,  0.03787962, -0.10867328,  0.10307689,
                 -0.00963774, -0.0877937 ,  0.07029191, -0.00913368, -0.10487436,
                  0.01276105,  0.11686436,  0.02419597,  0.10166058, -0.02140475,
                  0.17028385, -0.00392517, -0.05497375,  0.00594822,  0.01949847,
                  0.06495836,  0.10802469,  0.01538396, -0.02219599,  0.02749203,
                  0.15532522,  0.01753022, -0.0939441 ,  0.12369754,  0.130121  ,
                  0.06217542, -0.03164027,  0.06871128,  0.11664812, -0.09293687,
```

- step2：制作距离矩阵，取两个词的词向量之间的欧氏距离作为两个词之间的距离



```
In  [124]: lstm_distance

Out[124]: array([[0.        , 1.13067436, 1.95203602, ..., 1.14834356, 1.24748445,
                  1.1194073 ],
                 [1.13067436, 0.        , 1.92664671, ..., 1.12526202, 1.07407403,
                  0.82713801],
                 [1.95203602, 1.92664671, 0.        , ..., 1.76805532, 1.86803877,
                  1.8742038 ],
                 ...,
                 [1.14834356, 1.12526202, 1.76805532, ..., 0.        , 1.22868037,
                  0.91859883],
                 [1.24748445, 1.07407403, 1.86803877, ..., 1.22868037, 0.        ,
                  1.06184256],
                 [1.1194073 , 0.82713801, 1.8742038 , ..., 0.91859883, 1.06184256,
                  0.        ]])
```

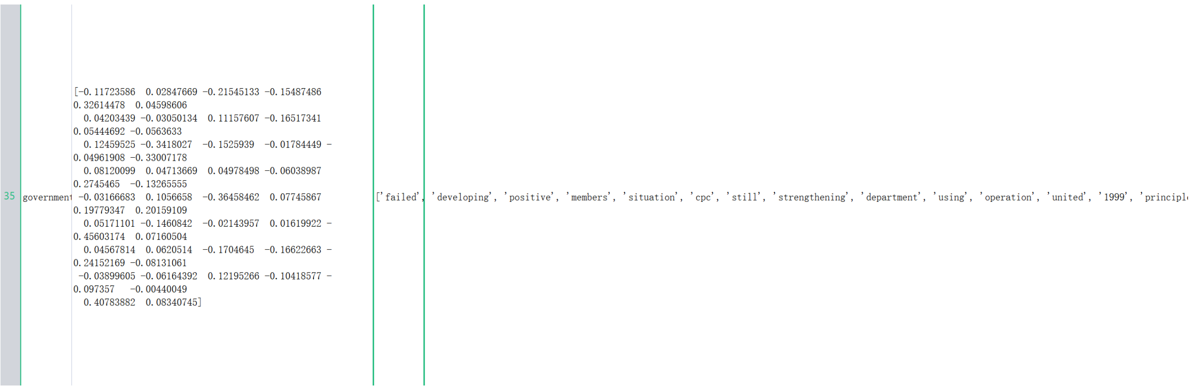
(2) 输出结果

对词汇表中的每个词，根据距离矩阵，提取与其最相似的Num_likely个词，并将结果输出为csv文件。

我们以单词“government"为例，看一看FNN，RNN，LSTM的输出结果：

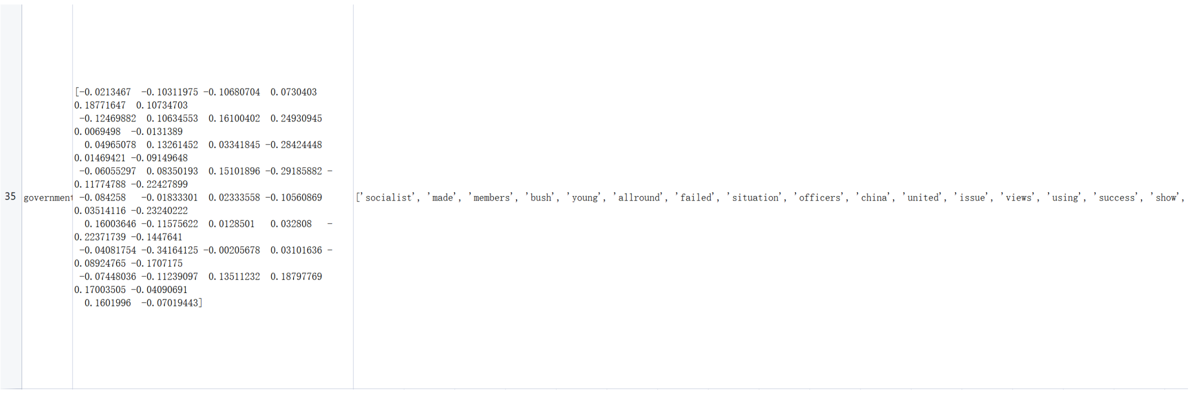
单词：“government” (表格中第三列最左边的词是与其联系最紧密的词)。

[1] FNN



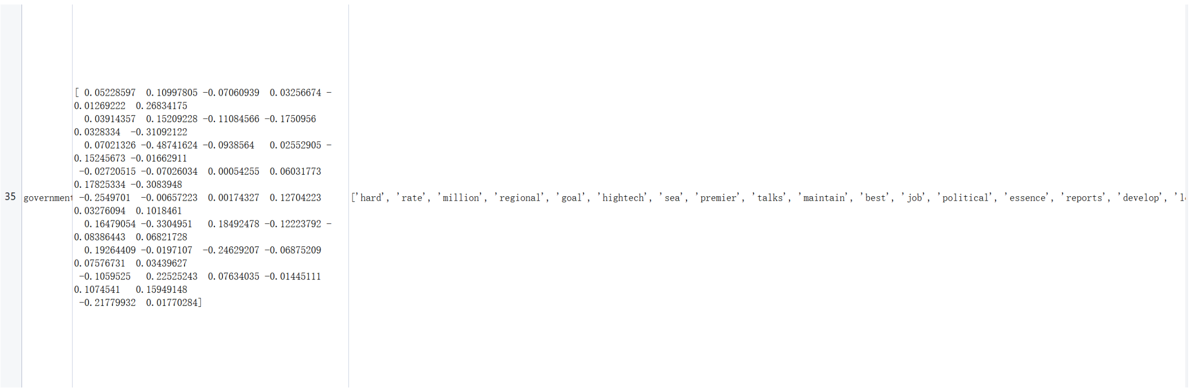
35	government	[-0. 11723586 0. 02847669 -0. 21545133 -0. 15487486 0. 32614478 0. 04598606 0. 04203439 -0. 03050134 0. 11157607 -0. 16517341 0. 05444692 -0. 0563633 0. 12459525 -0. 3418027 -0. 1525939 -0. 01784449 - 0. 04961908 -0. 33007178 0. 08120099 0. 04713669 0. 04978498 -0. 06038987 0. 2745465 -0. 13265555 -0. 03166683 0. 1056658 -0. 36458462 0. 07745867 0. 19779347 0. 20159109 0. 05171101 -0. 1460842 -0. 02143957 0. 01619922 - 0. 45603174 0. 07160504 0. 04567814 0. 0620514 -0. 1704645 -0. 16622663 - 0. 24152169 -0. 08131061 -0. 03899605 -0. 06164392 0. 12195266 -0. 10418577 - 0. 097357 -0. 00440049 0. 40783882 0. 08340745]	['failed', 'developing', 'positive', 'members', 'situation', 'cpc', 'still', 'strengthening', 'department', 'using', 'operation', 'united', '1999', 'principl

[2] RNN



35	government	[-0. 0213467 -0. 10311975 -0. 10680704 0. 0730403 0. 18771647 0. 10734703 -0. 12469882 0. 10634553 0. 16100402 0. 24930945 0. 0069498 -0. 0131389 0. 04965078 0. 13261452 0. 03341845 -0. 28424448 0. 01469421 -0. 09149648 -0. 06055297 0. 08350193 0. 15101896 -0. 29185882 - 0. 11774788 -0. 22427899 -0. 084258 -0. 01833301 0. 02333558 -0. 10560869 0. 03514116 -0. 23240222 0. 16003646 -0. 11575622 0. 0128501 0. 032808 - 0. 22371739 -0. 1447641 -0. 04081754 -0. 34164125 -0. 00205678 0. 03101636 - 0. 08924765 -0. 1707175 -0. 07448036 -0. 11239097 0. 13511232 0. 18797769 0. 17003505 -0. 04090691 0. 1601996 -0. 07019443]	['socialist', 'made', 'members', 'bush', 'young', 'allround', 'failed', 'situation', 'officers', 'china', 'united', 'issue', 'views', 'using', 'success', 'show',

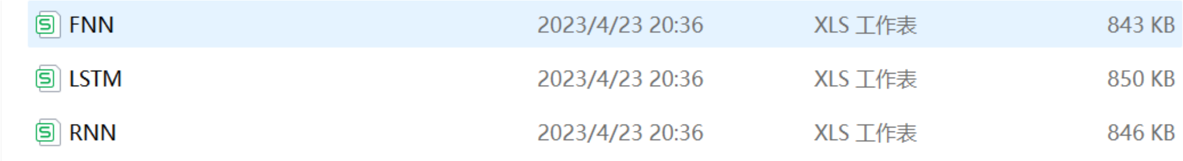
[3] LSTM



35	government	[0. 05228597 0. 10997805 -0. 07060939 0. 03256674 - 0. 01269222 0. 26834175 0. 03914357 0. 15209228 -0. 11084566 -0. 1750956 0. 0328334 -0. 31092122 0. 07021326 -0. 48741624 -0. 0938564 0. 02552905 - 0. 15245673 -0. 01662911 -0. 02720515 -0. 07026034 0. 00054255 0. 06031773 0. 17825334 -0. 3083948 -0. 2549701 -0. 00657223 0. 00174327 0. 12704223 0. 03276094 0. 1018461 0. 16479054 -0. 3304951 0. 18492478 -0. 12223792 - 0. 08386443 0. 06821728 0. 19264409 -0. 0197107 -0. 24629207 -0. 06875209 0. 07576731 0. 03439627 -0. 1059525 0. 22525243 0. 07634035 -0. 01445111 0. 1074541 0. 15949148 -0. 21779932 0. 01770284]	['hard', 'rate', 'million', 'regional', 'goal', 'hightech', 'sea', 'premier', 'talks', 'maintain', 'best', 'job', 'political', 'essence', 'reports', 'develop', 'l

三、实验结果

以上我们已经将FNN、RNN、LSTM三个模型训练过后得到的词向量结果及每个词最想似词结果输出到csv文件：



FNN	2023/4/23 20:36	XLS 工作表	843 KB
LSTM	2023/4/23 20:36	XLS 工作表	850 KB
RNN	2023/4/23 20:36	XLS 工作表	846 KB

这里我们随机取词汇表中10个词，对比三个模型得到的最相似20词结果，如下：

word	FNN	RNN	LSTM
government	['failed', 'developing', 'positive', 'members', 'situation', 'cpc', 'still', 'strengthening', 'department', 'using', 'operation', 'united', '1999', 'principle', 'order', 'essence', 'able', 'premier', 'outside', 'million']	['socialist', 'made', 'members', 'bush', 'young', 'allround', 'failed', 'situation', 'officers', 'china', 'united', 'issue', 'views', 'using', 'success', 'show', 'province', 'far', 'development', 'present']	['hard', 'rate', 'million', 'regional', 'goal', 'hightech', 'sea', 'premier', 'talks', 'maintain', 'best', 'job', 'political', 'essence', 'reports', 'develop', 'lot', 'anniversary', 'members', 'made']
control	['policy', 'failed', 'beginning', 'high', 'defense', 'assistance', 'research', 'promote', 'general', 'form', 'productive', 'progress', 'best', 'zhang', 'put', 'come', 'fine', 'dialogue', 'committee', 'process']	['understanding', 'failed', '3', 'words', 'zemin', 'stressed', 'direction', 'democracy', 'committees', 'present', 'made', 'growth', 'internal', 'department', 'across', 'concerned', 'exchange', 'hard', 'situation', 'socialist']	['natural', 'maintained', 'another', 'possible', 'accordance', 'korea', 'establishment', 'exchange', 'reason', 'protecting', 'historic', 'improvement', 'status', 'put', 'believe', 'ties', 'step', 'difficulties', 'prospects', 'sinojapanese']
efforts	['stability', 'korean', 'failed', 'issue', 'natural', '40', 'party', 'province', 'hongzhi', 'economic', 'largescale', 'shuibian', 'even', 'committee', 'seen', 'maintaining', 'opening', 'realize', 'sea', 'seek']	['issue', 'present', 'style', 'stressed', 'views', 'china', 'powell', 'peoples', 'april', 'high', 'cpc', 'zemin', 'wto', 'united', 'discussion', 'domestic', 'failed', 'form', 'issued', 'internal']	['organizations', 'ministry', 'arms', 'going', 'globalization', 'localities', 'close', 'allround', 'seen', 'defense', 'judicial', 'vice', 'develop', 'stability', 'officers', 'addition', 'direct', 'onechina', 'present', 'order']
war	['failed', 'principle', 'better', 'defense', 'parties', 'question', 'economic', 'course', 'due', 'united', 'election', 'structural', 'situation', 'financial', 'administrative', 'communist', 'bush', '40', 'addition', 'talks']	['failed', 'united', 'political', 'development', 'present', 'internal', 'exchanges', 'principle', 'cpc', 'far', 'members', 'style', 'issue', 'country', 'promote', 'addition', 'struggle', 'socialist', 'bush', 'whole']	['high', 'science', 'peace', 'promote', 'rule', 'production', 'find', 'official', 'constitution', 'role', 'industry', 'research', 'percent', 'national', 'largescale', 'territorial', 'news', '3', 'using', 'improvement']

word	FNN	RNN	LSTM
market	['financial', 'could', 'committee', 'failed', 'zemin', 'university', 'UNK', 'hoped', 'develop', 'peoples', 'among', 'really', 'natural', 'economy', 'communist', 'remarks', 'measures', 'progress', 'democratic', 'visiting']	['understanding', 'committee', 'peasants', 'leading', 'rights', 'history', 'present', 'another', 'UNK', 'next', 'without', 'could', 'socialist', 'form', 'department', 'brought', 'small', 'failed', 'cause', 'united']	['understanding', 'university', 'build', 'brought', 'zemin', 'building', 'regions', 'problems', 'hightech', 'hoped', 'addition', 'stressed', 'corruption', 'successfully', 'macao', 'front', 'short', 'step', 'conditions', 'hopes']
laws	['province', 'come', 'full', 'hard', 'failed', 'possible', 'committee', 'improvement', 'process', 'united', 'regard', 'second', 'projects', 'nuclear', 'environment', 'political', 'provincial', 'back', 'lies', 'socialist']	['failed', 'asia', 'get', 'future', 'regard', 'nuclear', 'judicial', 'strong', 'made', 'leading', 'working', 'promoting', 'course', 'comprehensively', 'two', 'strait', 'part', 'back', 'socialist', 'united']	['regional', 'develop', 'prospects', 'sea', 'clinton', 'competition', 'funds', 'adopt', 'process', '1999', 'maintained', 'organizations', 'really', 'pressure', 'direct', 'arms', 'plan', 'second', 'reports', 'localities']
dialogue	['necessary', 'today', 'plans', 'efforts', 'economic', 'defense', 'science', 'using', 'toward', 'developing', 'failed', 'however', 'better', 'afternoon', 'strengthening', 'seen', 'guiding', 'county', 'changes', 'enterprise']	['nearly', 'development', 'began', 'issue', 'united', 'largescale', 'invitation', 'first', 'developing', 'failed', 'basis', 'korean', 'success', 'common', 'south', 'present', 'find', 'course', 'minister', 'political']	['words', 'hand', 'august', 'largescale', 'share', 'thanks', 'korean', 'toward', 'research', 'industry', 'guidance', 'opening', 'using', 'development', 'view', 'say', 'study', 'today', 'wu', 'correct']
complete	['study', 'third', 'concerned', 'benefit', 'failed', 'make', 'quality', 'rural', '40', 'natural', 'views', 'constitution', 'policies', 'although', 'good', 'talks', 'still', 'back', 'therefore', 'reason']	['prc', 'failed', 'constitution', 'operation', 'reunification', 'members', 'bush', 'concerned', 'certain', 'lu', '10th', 'antichina', 'adopt', 'contacts', 'closely', 'zemin', 'socalled', 'third', 'ethnic', 'create']	['half', 'speech', 'build', 'person', 'provided', 'units', 'democracy', 'entry', 'scope', 'quality', 'always', '100', 'large', 'financial', 'really', '2', 'study', 'office', 'around', 'national']

word	FNN	RNN	LSTM
continued	['operation', 'failed', 'big', '40', 'domestic', 'course', 'addition', 'cadres', 'wei', 'firmly', 'vigorously', 'socialist', 'stressed', 'peoples', 'cpc', 'issue', 'responsible', 'handling', 'evil', 'basis']	['deputy', 'present', 'high', 'failed', 'power', 'members', 'kind', 'plan', 'efforts', 'province', 'wto', 'amount', 'united', 'april', 'views', 'theory', 'existing', 'changed', 'therefore', 'service']	['half', 'speech', 'build', 'person', 'provided', 'units', 'democracy', 'entry', 'scope', 'quality', 'always', '100', 'large', 'financial', 'really', '2', 'study', 'office', 'around', 'national']
visiting	['circles', 'forms', 'added', 'basis', 'sound', 'opportunities', 'advance', 'united', 'financial', 'cult', 'dialogue', 'responsibility', 'failed', 'current', 'korean', 'center', 'agreed', 'success', 'face', 'production']	['circles', 'last', 'development', 'ninth', 'hu', 'internal', 'political', 'responsibility', 'says', 'failed', 'difficult', 'shows', 'links', 'progress', 'members', 'resources', 'financial', 'president', 'management', 'zemin']	['october', 'able', 'ago', 'strengthening', 'economy', 'using', 'ninth', 'development', 'electronic', 'zemin', 'objective', 'office', 'circles', 'within', 'actual', 'past', 'internal', 'success', 'structural', 'used']