Exp2 报告

Author:

- **骆霄龙 PB18151853**
- **张越群 PB17051070**

程序概要：

对数据集实现了两种不同模型下的关系抽取，分别为基于TF-IDF与逻辑回归模型和BiLSTM模型实现。并进行了模型融合 与预测结果的可视化与分析。

功能实现:

Model1:TF-IDF

- 主要利用sklearn的`TfidfVectorizer`和`LogisticRegression(solver='liblinear')`实现。通过将训练数据用TfidfVectorizer编码后对`LogisticRegression(solver='liblinear')`生成的模型进行训练。参数则使用默认参数。 随后对编码后的测试数据预测即可。

Model2:BiLSTM

- 模型参数: 其中Embedding层的embedding_size = 100。

```
Model: "sequential_11"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_11 (Embedding)     (None, 83, 100)           771500
_________________________________________________________________
bidirectional_11 (Bidirectio (None, 200)               160800
_________________________________________________________________
dense_11 (Dense)             (None, 10)                2010
_________________________________________________________________
activation_11 (Activation)   (None, 10)                0
=================================================================
Total params: 934,310
Trainable params: 934,310
Non-trainable params: 0
```

- 主要使用tensorflow的keras 实现
  - 通过VocabularyProcessor()构建词典，sorted_vocab中保存了每个词对应的数值编码。利用word2vec词袋化模型对出现的词进行编码，将分词后的文本转换为数值向量。

```
# word2vec 词袋化
vocab_processor = tf.contrib.learn.preprocessing.VocabularyProcessor(max_sequence_length,min_frequency=0)

text_processed_test = np.array(list(vocab_processor.fit_transform(texts_test)))
```

  - 通过keras来compile模型，保存模型训练记录:

```
ckpt = krs.callbacks.ModelCheckpoint('./temp/ckpt', monitor='val_categorical_accuracy', verbose=1,
                                     save_best_only=True, save_weights_only=False, period=1)
earlystop = krs.callbacks.EarlyStopping(monitor='val_loss', min_delta=0.0001,
                                        patience=10, verbose=1)
# patience: n次acc未增长则终止训练
```

- 模型构建:

```
model = krs.Sequential()
model.add(krs.layers.Embedding(len(dict.items()),
            embedding_size, input_length = max_sequence_length,
             mask_zero= True  ,dropout = 0.2  ))
model.add(krs.layers.Bidirectional(krs.layers.LSTM(100,dropout = 0.5)))
model.add(krs.layers.Dense(10))
model.add(krs.layers.Activation("softmax"))
#model.summary()
model.compile(loss = "categorical_crossentropy", optimizer= "adam", metrics=["categorical_accuracy"])
models.append(model)
```

- 进行10-Folds训练后，将模型预测结果投票:

```
def softmax_to_onehot(result):
    result = np.argmax(result, axis=1)
    one_hot = np.zeros((len(result),10))
    one_hot[np.arange(len(result)),result] =1
    return one_hot
```

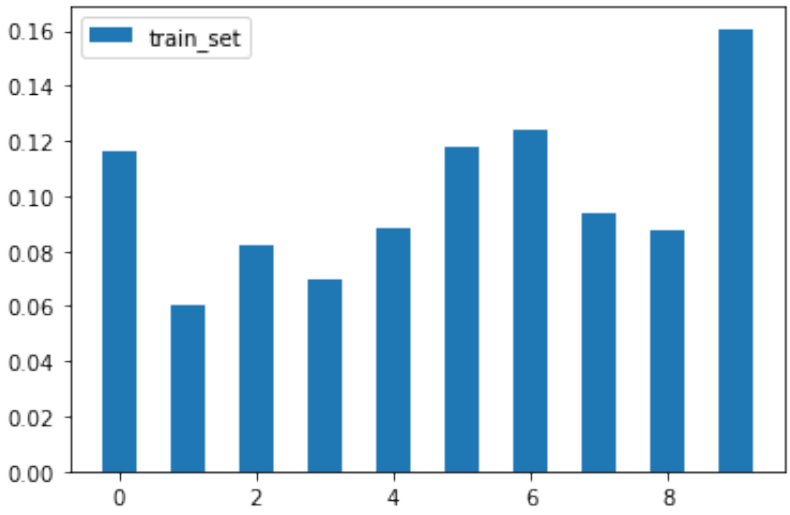
```
ed in 3ms, finished 19:58:25 2021-01-04
```

```
y = softmax_to_onehot(models[0].predict(text_processed_test))
for i in range(1,10):
    y = y+ softmax_to_onehot(models[i].predict(text_processed_test))
```

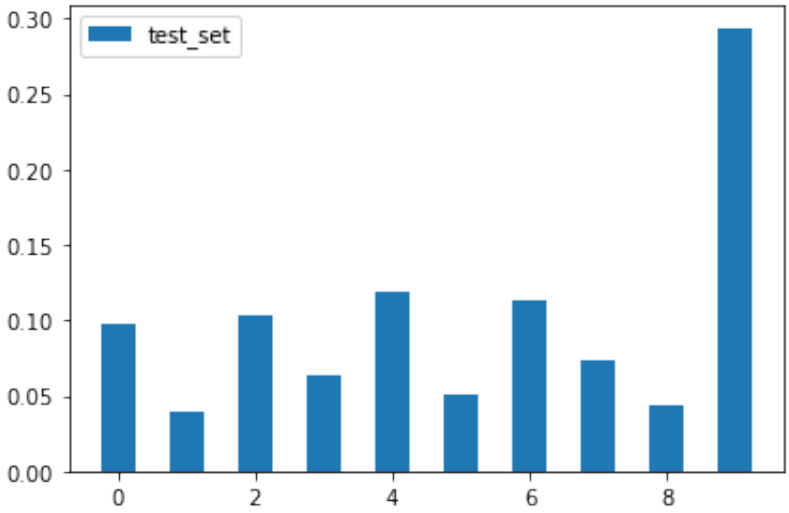
优化与分析

- **模型预测情况:**

- 训练集的labels:

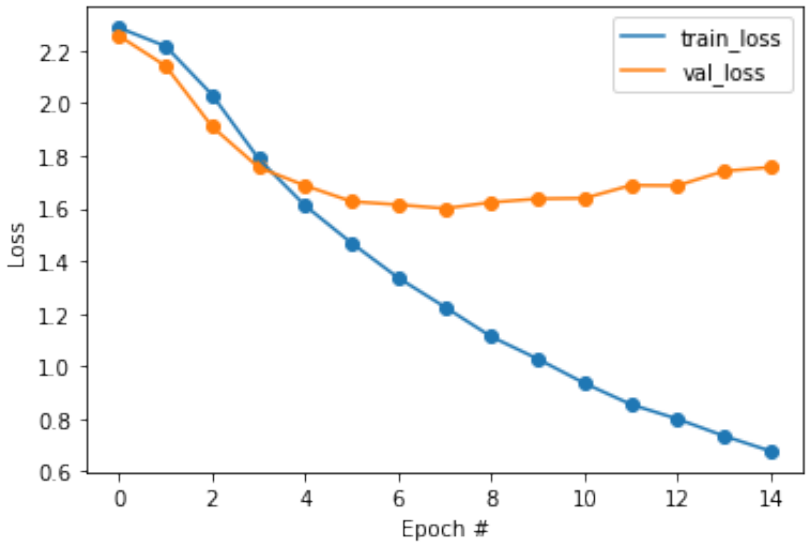


- 最终测试集的labels:



- 对比两个图可以看出模型对lable 9 的选择存在较大的偏好。 可能是数据中label9过多并且在模型训练中放大了这种效应。 可以考虑通过采样构造出各个类相对均匀的训练集合来提高模型准确度

影响效果的主要问题:

- 从理论上来说，BiLSTM模型能更好地捕捉双向的语义关系，从而取得不错的效果。但在此实验中，BiLSTM模型在这个数据集合上有严重的overfitting 问题，即使经过10-folds 的交叉训练和投票决定仍然对模型的overfitting问题解决帮助不大。 这也是导致本次实验准确率一直较低的主要问题所在。
    - Overfitting曲线1:
    - 

    - 可以看出在训练集合上train_loss还有对应的accuracy一直提升（可以到达80+），但在验证集上val_loss到1.6左右就不在变换，accuracy也维持在0.46-0.48的水平(略高于最终的测试集结果)
- TF-IDF模型表现力不够强，仅通过简单的词向量化和编码准确率并不是很高。因为形式非常的简单(非常类似线性模型)，很难去拟合数据的真实分布，同时也很难处理数据不平衡的问题，并且逻辑回归本身无法师选特征。所以单独使用此模型仅能得到接近于40%的accuracy。 而单个BiLSTM模型的准确率大概为41-42%。 通过训练多个BiLSTM网络并与TF-IDF模型融合最终得到的模型准确率为45.25%.

实验结果:

Filename	ACC-Relation	ACC-NER
张越群-PB17051070-3.txt	0.4525	0.0

分工情况

- **骆霄龙 PB18151853**：实现数据预处理， TF-IDF模型以及前期实验
- **张越群 PB17051070**：实现BiLSTM模型以及模型融合
- 基于实验的分工和合作，我们一致认为在本次实验中两个人贡献平等。

参考资料:

1. https://lab.datafountain.cn/forum?id=154
2. https://lab.datafountain.cn/forum?id=156
3. 基于 **CNN** 和双向 **LSTM** 融合的实体关系抽取;张晓斌，陈福才，黄瑞阳;网络与信息安全学报