中国科学院大学
University of Chinese Academy of Sciences

自 然 语 言 处 理

技术报告

班级：2023 春季自然语言处理班

组号：--

姓名：唐嘉良

学号：2020K8009907032

报告主题：文本分类——基于搜狗实验室 2006 年数据集

2023 年 7 月 10 日

一．报告摘要

本次实验中，我基于搜狗实验室在2006年发布的文本分类数据集，在开源文本分类模型框架（https://github.com/lijqhs/text-classification-cn）的基础上，运用传统机器学习模型中不同的文本分类器完成分类任务，并对这几种分类器的性能进行了评估。实验结果表明，朴素贝叶斯分类器、Logistic Regression分类器和SVM分类器在该数据集上的分类准确率分别达到了84%，88%和89%。进一步地，我探究了不同分类器在不同种类文本上的性能差异，进行了相关结论的总结。

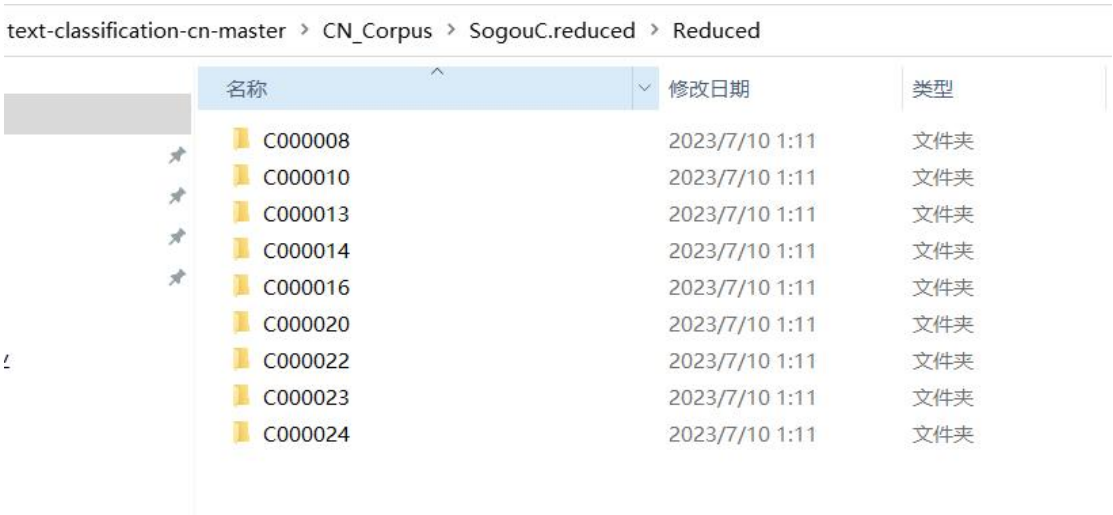
二．实验环境

环境配置：python 3.9.7，tensorflow == 2.12.0，protobuf == 3.20.3，keras == 2.12.0，wordcloud 1.9.2，sklearn 0.0，jieba 0.42.1等

三．实验过程与方法

1.数据集准备与预处理

本实验的数据集我们采用搜狗实验室在2006年发布的文本分类数据集，包括财经、IT、健康、体育、旅游、教育、招聘、文化和军事9个领域各1990个文件，并将这些原始语料按照类似C000008的标签组织子目录，存放在CN_Corpus/SogouC.reduced/Reduced目录下。



对上述原始语料进行如下预处理：将每一类语料的前80%的文件作为训练语料，后20%的文件作为评测语料，进行完这样的切分之后将训练语料和评测语料分开存放在目录data中。利用load_datasets()函数进行文本数据和对应标签的读取，读取完毕后使用jieba开源库进行汉语分词，得到分词语料。数据集切分结构、切分细节和分词代码分别如下：

```
data
├──test
|    ├──_08_Finance
|    ├──_10_IT
|    ├──_13_Health
|    ├──_14_Sports
```

```
|   ├──_16_Travel
|   ├──_20_Education
|   ├──_22_Recruit
|   ├──_23_Culture
|   └──_24_Military
└──train
    ├──_08_Finance
    ├──_10_IT
    ├──_13_Health
    ├──_14_Sports
    ├──_16_Travel
    ├──_20_Education
    ├──_22_Recruit
    ├──_23_Culture
    └──_24_Military
```

```
label: _08_Finance, len: 1500
Building prefix dict from the default dictionary ...
Dumping model to file cache C:\WINDOWS\TEMP\jieba.cache
Loading model cost 0.555 seconds.
Prefix dict has been built successfully.
label: _10_IT, len: 1500
label: _13_Health, len: 1500
label: _14_Sports, len: 1500
label: _16_Travel, len: 1500
label: _20_Education, len: 1500
label: _22_Recruit, len: 1500
label: _23_Culture, len: 1500
label: _24_Military, len: 1500
train corpus len: 13500

label: _08_Finance, len: 490
label: _10_IT, len: 490
label: _13_Health, len: 490
label: _14_Sports, len: 490
label: _16_Travel, len: 490
label: _20_Education, len: 490
label: _22_Recruit, len: 490
label: _23_Culture, len: 490
label: _24_Military, len: 490
test corpus len: 4410
```

```
def preprocess(text):
    text1 = re.sub(' ', ' ', text)
    str_no_punctuation = re.sub(token, ' ', text1)  # 去掉标点
    text_list = list(jieba.cut(str_no_punctuation))   # 分词列表
    text_list = [item for item in text_list if item != ' '] # 去掉空格
    return ' '.join(text_list)
```

2.文本特征提取

为提取文本特征，我们使用 sklearn 这一机器学习库，使用对应方法对训练数据进行 TF-IDF 特征提取，这里 TfidfVectorizer 将文档集合转为 TF-IDF 矩阵。

```
# TF-IDF feature extraction
tfidf_vectorizer = TfidfVectorizer(stop_words=stopwords)
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train_data)
words = tfidf_vectorizer.get_feature_names()
```

3.训练分类器——基于朴素贝叶斯、Logistic Regression 和 SVM

分类器的构建可以直接引用 scikit-learn 库中提供的分类器封装，其中我们引用 MultinomialNB 作为朴素贝叶斯分类器的实例，LogisticRegression 作为逻辑回归分类器实例，SGDCClassifier 作为 SVM 分类器实例。

```
text_clf = Pipeline([
    ('vect', TfidfVectorizer()),
    ('clf', MultinomialNB()),
])
```

```
text_clf_lr = Pipeline([
    ('vect', TfidfVectorizer()),
    ('clf', LogisticRegression()),
])
```

```
text_clf_svm = Pipeline([
    ('vect', TfidfVectorizer()),
    ('clf', SGDClassifier(loss='hinge', penalty='l2')),
])
```

对三种分类器采用同一批训练数据和评测数据进行训练与评测，生成评测报告，进行对比分析。

四．实验结果分析

1. 三种分类器评测结果

三种分类器评测结果如下（使用 **sklearn** 库中的 **classification_report** 生成）：

	precision	recall	f1-score	support
_08_Finance	0.88	0.88	0.88	488
_10_IT	0.72	0.87	0.79	403
_13_Health	0.82	0.84	0.83	478
_14_Sports	0.95	1.00	0.97	466
_16_Travel	0.86	0.92	0.89	455
_20_Education	0.71	0.87	0.79	401
_22_Recruit	0.91	0.65	0.76	690
_23_Culture	0.80	0.77	0.79	513
_24_Military	0.94	0.89	0.92	516
accuracy			0.84	4410
macro avg	0.84	0.86	0.84	4410
weighted avg	0.85	0.84	0.84	4410

朴素贝叶斯分类器性能报告

	precision	recall	f1-score	support
_08_Finance	0.87	0.91	0.89	465
_10_IT	0.77	0.86	0.81	439
_13_Health	0.91	0.81	0.86	550
_14_Sports	0.98	0.99	0.99	484
_16_Travel	0.90	0.90	0.90	489
_20_Education	0.80	0.91	0.85	431
_22_Recruit	0.86	0.86	0.86	487
_23_Culture	0.87	0.76	0.81	562
_24_Military	0.95	0.92	0.93	503
accuracy			0.88	4410
macro avg	0.88	0.88	0.88	4410
weighted avg	0.88	0.88	0.88	4410

逻辑回归分类器性能报告

	precision	recall	f1-score	support
_08_Finance	0.87	0.92	0.90	464
_10_IT	0.78	0.86	0.81	445
_13_Health	0.93	0.82	0.87	559
_14_Sports	0.99	0.99	0.99	488
_16_Travel	0.91	0.91	0.91	490
_20_Education	0.81	0.91	0.86	435
_22_Recruit	0.89	0.85	0.87	515
_23_Culture	0.84	0.83	0.84	500
_24_Military	0.96	0.91	0.93	514
accuracy			0.89	4410
macro avg	0.89	0.89	0.89	4410

SVM 分类器性能报告

从上述结果中可以看出：朴素贝叶斯分类器、Logistic Regression 分类器和 SVM 分类器在该数据集上的平均分类准确率分别达到了 84%，88%和 89%。这对于传统机器学习方法来说已经比较可观。

2.三种分类器在不同类型文本的识别准确率对比

朴素贝叶斯分类器尽管是理论上的最优分类器，但是在实际应用中仍然与理论性能相距较远，平均性能是三种分类器中最差的，特别地，该分类器在 IT 领域和教育领域文本分类任务中表现很差。

LogisticRegression 分类器相比朴素贝叶斯方法提升了些许平均性能，并且在某些领域的表现改观很多，例如健康领域文本分类准确率相比朴素贝叶斯分类器提升了 9%。但是另一方面该分类器仍然存在在特别领域文本上性能较差的情况，例如 IT 文本正确率为 77%，教育文本正确率为 80%。

SVM 的分类器在三种分类器中表现最突出，整体性能相比 LogisticRegression 分类器又有较大提升，但是其仍然无法避免在 IT 领域和教育领域文本分类任务上有较大缺陷。

3.实验特征与缺陷分析

通过上面对实验结果的分析可知，传统机器学习方法的三种分类器整体性能不错，但是在某些特定领域数据集上会丢失不少准确度。推测是因为所选取的数据集有“多标签”的特性，例如某一段文字可能和多个领域均有较大关系，这种情况下文本数据的标签具有一定模糊性，从模糊标签文本中训练出的单标签的分类器难以做到十分精确地分类文本，尤其是在评测文本数据也具有模糊标签的情况下。

此外，2006 年搜狗实验室的文本数据量在如今看来并不足够庞大，标签种类丰富程度低，并且数据总量也仅有 10MB 量级，这在一定程度上影响了分类器性能。当然，本实验仅仅是基于传统机器学习方法，它已经被无数人验证过在性能上难以达到神经网络方法的程度，因此数据量的边际效应或许也并不会那么明显。