文本数据向量化以及不同的机器学习算法在文本分类的应用——实验报告

1. 方法

1.1. 数据

本次实验使用 scikit-learn 1.4.2（Pedregosa et al., 2011）中的 20 newsgroups 数据集。该数据集共 18846 篇文档。其中训练集共 11314 篇，测试集共 7532 篇。每篇文档均有分类标注。标注共五种粗分类别，20 种细分类别。围绕 comp、rec、sci、talk、other 五种粗分类别，本次实验测试了不同机器学习算法的文本分类能力。

1.2. 数据处理

1.2.1. 文本预处理

为保证文本特征抽取和向量化的质量，该实验使用 spaCy 3.7.5（Honnibal & Johnson, 2015）对训练集和测试集文档都进行了文本预处理。具体来说，实验去除了文档中的空白字符、标点符号、停用词，并对单词进行了词元化（lemmatization）。训练集与测试集预处理结果均以文本文件（UTF-8 编码）保存在本地。

1.2.2. 文本向量化

文本向量化的特征首先为 20 newsgroups 训练集文档中空白字符、标点符号、停用词之外单词的词元（lemma），共 138540 项。为控制特征维度，实验统计了这些词元在处理后训练集的出现频数，并过滤了频数小于五的词元。词元过滤后总计 24666 项。本实验为这些词元分配了索引（ID），保存在大小为 24666 的字典中。同时，字典中的词元按出现频数从大到小排列，便于后续选择 1000，2000，5000，8000，10000 的 top-K 特征维度进行模型调参。

基于所得字典，实验将 20 newsgroups 训练集与测试集转化为两项形状是（文档数* 特征数）的 TF-IDF 矩阵。首先，实验为字典中的每一个词元计算了其在所选集全部文档的 IDF 值，得到 IDF 一维密集数组。接下来，实验计算了字典中每个词元在所选集每篇文档的 TF 值，得到 TF 稀疏矩阵。通过将 TF 矩阵的每个行向量与 IDF 数组向量进行元素级别的乘法运算，便得到了 TF-IDF 矩阵。同时，为避免文本长度影响，实验为 TF-IDF 矩阵的每个行向量，即每个文档向量，进行了 L2 归一化。训练集与测试集向量化结果分别以形状为（11314*24666）和（7532*24666）的 TF-IDF 稀疏矩阵（CSR 格式）保存在本地。

1.2.3. 模型分类比较

针对文本分类任务，本实验在 scikit-learn 中分别调用了 MultinomialNB、GaussianNB、DecisionTreeClassifier、KNeighborsClassifier、SVC、LogisticRegression 六种机器学习算法，对它们展开比较。

为对六类方法进行训练与测试，实验准备了训练集与测试集的 TF-IDF 矩阵、训练集与测试集的文档粗分类别编码向量、粗分类别名称。需要注意的是，20 newsgroups 原先的类别编码向量、类别名称为 20 种细分类，而该实验基于五种粗分类。因此，原先的细分类别编码向量、细分类别名称都映射到了粗分类别，即{‘comp’:0, ‘rec’:1, ‘sci’:2, ‘talk’:3, ‘other’:4}。不过该实验仍然保留了细分类别编码向量与类别名称，旨在观察上述算法在粗、细任务分类下的表现浮动。

在模型训练上，本实验主要使用训练集 TF-IDF 矩阵和训练集文档粗分类别编码向量。训练参数方面，实验按特征频数为训练集矩阵准备了 1000、2000、5000、8000、10000 五种 top-K 维度，并使用 GridSearchCV 针对不同机器学习算法进行网格搜索交叉验证。此处，GridSearchCV 均使用十折交叉验证，得分标准采用“f1_macro”。上述条件下，五种 top-K 特征的训练数据均得到其最佳模型与最佳模型参数。其中，“f1_macro”得分最

高的 top-K 特征训练数据的维度便为最佳维度，其训练模型为最佳模型，其训练模型参数为最佳模型参数。各模型最佳训练参数与得分如表一所示。值得一提的是，本实验在网格搜索参数外还对特定分类模型设置了部分参数：DecisionTreeClassifier、SVC、LogisticRegression 的 random_state 均设置为 42 以复现实验；LogisticRegression 的 solver 设置为适合大规模文本分类任务的 saga，max_iter 设置为 1000 保证收敛效果。接下来，这些参数下训练的最佳模型便可用于测试集预测。此外，实验还以同样的方法训练了细分类任务模型，用来观察模型在粗、细任务下的不同表现。

表一：最佳训练参数与交叉验证得分

Classifiers	Parameters	F1_Macro (CV)
MultinomialNB	alpha: 0.01; dims: 10000	0.8910
GaussianNB	var_smoothing: 1e-05; dims: 10000	0.8427
DecisionTreeClassifier	criterion: gini; dims: 10000; (random_state: 42)	0.6857
KNN	n_neighbors:1; metric: cosine; dims: 10000	0.8570
SVC	kernel: linear; degree: 2; dims: 10000; (random_state=42)	0.9012
LogisticRegression	tol: 0.0001; C: 2.5; dims=10000; (solver: saga; random_state:42; max_iter:1000)	0.9031

模型测试上，实验主要使用测试集 TF-IDF 矩阵和测试集文档粗分类编码向量。首先，测试集矩阵按所得最佳维度切片（均为 10000 维）。切片后矩阵传入最佳模型得到预测分类编码向量。通过对比预测分类编码向量与测试集实际分类编码向量，实验分别为六种算法分类模型保存了分类结果报告、混淆矩阵、宏平均得分。同样，实验也计算了模型细分类任务下的宏平均得分，用于比较。

2. 结果

2.1. 粗分类任务表现

对六类器学习算法最佳模型测试后，实验得到了每种模型的宏平均精确率、召回率、f1 分数，如表二所示。总体看来，六种模型在五分类任务的 f1 得分在 0.63 到 0.87 之间，均表现较好，从高到低分别为 LogisticRegression（0.8662）、SVC（0.8597）、MultinomialNB（0.8484）、GaussianNB（0.7671）、KNN（0.7241）、DecisionTreeClassifier（0.6386）。其中 LogisticRegression、SVC、MultinomialNB 表现相对优秀（f1>0.80），DecisionTreeClassier 表现相对较差（0.7>f1>0.6）。而在精确率、召回率上，六种模型在分类时对精确率的追求都高于召回率。同时，在精确率和召回率差值上，f1 表现更好的 LogisticRegression、SVC、MultinomialNB 的平均差值（0.0331）也比 GaussianNB、KNN、DecisionTreeClassifier 的平均差值（0.0055）更高。因此，在目标文本粗分类性能上，LogisticRegression、SVC 、MultinomialNB 较好，DecisionTreeClassifier 相对不足，该性能还可能与分类器对精确性的追求有所关系。

表二：模型粗分类宏平均得分

Classifiers	MacroAvg-Precision	MacroAvg-Recall	MacroAvg-F1
MultinomialNB	0.8835	0.8351	**0.8484**
GaussianNB	0.7753	0.7641	0.7671
DecisionTreeClassifier	0.6411	0.6367	**0.6386**
KNN	0.7278	0.7269	0.7241
SVC	0.8740	0.8498	**0.8597**
LogisticRegression	0.8826	0.8559	**0.8662**

2.2. 粗、细分类任务模型表现浮动

实验还计算了模型在 20 分类任务的最佳宏平均 f1 得分，并得到了模型在粗、细分类任务下 f1 的得分差值，如表三所示。可见，更严格的分类条件下，MultinomialNB（0.8142）、LogisticRegression（0.8038）、SVC（0.7896）表现仍相对较好，DecisionTreeClassifier（0.5474）相对较差。同时，通过 f1 得分差值还能发现，在 MultinomialNB（0.0342）、LogisticRegression（0.0624）、SVC（0.0701）三项较优模型中，MultinomialNB 的性能变化更低，鲁棒性更好。

表三：模型粗、细分类任务表现浮动

Classifiers	MacroAvg-F1 (5 classes)	MacroAvg-F1 (20 classes)	Difference
MultinomialNB	**0.8484**	**0.8142**	**0.0342**
GaussianNB	0.7671	0.6793	0.0878
DecisionTreeClassifier	**0.6386**	**0.5474**	0.0912
KNN	0.7241	0.6257	0.0984
SVC	**0.8597**	**0.7896**	0.0701
LogisticRegression	**0.8662**	**0.8038**	0.0624

2.3. 模型粗分类错误分析

针对不同模型，实验还记录它们在 comp、rec、sci、talk、other 五类具体类别上的 f1 得分，如表四所示。可以发现，六种模型在“other”类别的得分均为最低。这说明模型在文档分类“other”时错误最多。

表四：模型在具体类别的 f1 得分

Classifiers	Comp	Rec	Sci	Talk	Other
MultinomialNB	0.90*	0.95	0.85	0.89	**0.66**
GaussianNB	0.84	0.90	0.76	0.82	**0.51**
DecisionTreeClassifier	0.70	0.72	0.58	0.63	**0.57**
KNN	0.79	0.76	0.74	0.72	**0.62**
SVC	0.89	0.93	0.85	0.86	**0.77**
LogisticRegression	0.91	0.94	0.86	0.87	**0.75**

*f1 得分

因此，实验通过各模型的混淆矩阵观察了“other”分类的错误情况，分别从精确率（模型预测为 other 的文档是否实际为 other）和召回率（实际为 other 的文档是否被模型预测为 other）的角度记录在了表五中。这里，我们将错误数量占比大于等于 10%的类

型视为主要错误类型。

首先，从精确率来看：GaussianNB 分类为 other 的主要错误类型为 comp（14%）和 sci（13%）；DecisionTreeClassifier 主要错误类型为 comp（12%）、sci（14%）和 talk（10%）；KNN 主要错误类型为 comp（13%）和 sci（11%）。三种分类器均易将 comp 和 sci 视为 other。而从召回率来看：MultinomialNB 主要将 other 错误分类为了 comp（19%）和 sci（13%）；GaussianNB 主要错分为 comp（15%）、sci（17%）和 talk（15%）；DecisionTreeClassifier 也主要错分为 comp（14%）和 sci（15%）；KNN 主要错分为 comp（13%）和 rec（11%）；SVC 主要错分为 sci（12%）；LogisticRegression 也主要错分为 sci（13%）。六种分类器基本上容易把 other 错误分类成 comp 或 sci。可见，不论在精确率还是召回率上，模型在学习区分“other”和“camp、sci”上都存在一定问题。

表五：文档在 other 类别的分类情况

Metrics	**Classifiers**	**comp**	**rec**	**sci**	**talk**	**other**
Precision	MultinomialNB	6*	3	10	22	366
	GaussianNB	**70(14%)**	28	**70(13%)**	43	321
	DecisionTreeClassifier	**80(12%)**	41	**93(14%)**	**65(10%)**	391
	KNN	**86(13%)**	24	**73(11%)**	62	424
	SVC	21	14	27	23	494
	LogicsticRegression	19	10	23	19	468
Recall	MultinomialNB	**137**(19%)**	48	**94(13%)**	64	366
	GaussianNB	**107(15%)**	53	**124(17%)**	**104(15%)**	321
	DecisionTreeClassifier	**98(14%)**	57	**104(15%)**	59	391
	KNN	**91(13%)**	**77(11%)**	64	53	424
	SVC	59	17	**86(12%)**	53	494
	LogicsticRegression	69	25	**92(13%)**	55	468

*模型预测为 other 类别，实际为当前列类别的文档的数量

**实际类别为 other，模型预测为当前列类别的文档的数量

3. 结论

本次实验对 20 newsgroups 数据进行了文本预处理，特征选择和 TF-IDF 矩阵向量化。同时，实验比较了 MultinomialNB、GaussianNB、DecisionTreeClassifier、KNeighborsClassifier、SVC、LogisticRegression 六种机器学习算法在该数据文本分类上的表现。总的来说，实验发现：（1）在五分类任务下，LogisticRegression、SVC、MultinomialNB 表现较好，DecisionTreeClassifier 相对不足。模型对精确性的追求也可能与最终性能相关；（2）二十分类下的模型比较与五分类一致，同时着重反映了 MultinomialNB 的鲁棒性；（3）在 comp、rec、sci、talk、other 五种类型上，全部模型都在 other 上错误最多。且“other”这一类别不论在精确率还是召回率角度都容易与“camp、sci”混淆。

参考文献：

Honnibal, M., & Johnson, M. (2015). An improved non-monotonic transition system for dependency parsing. In L. Màrquez, C. Callison-Burch, & J. Su (Eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1373-1378). Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1162

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *JMLR, 12*, 2825-2830.