实验结果

实验1.

TF-IDF(6000个词表,max_df=0.4, min_df=0.001)

	#Train.acc	#Test.acc
RandomForestClassifier	0.483	0.4607
LogisticRegression	0.7606	0.7226
MultinomialNB	0.6835	0.6637
SVC		0.7641
LightGBM		

实验2.

TF-IDF(73351个词,max_df=0.4, min_df=0.0001)

	#Train.acc	#Test.acc
RandomForestClassifier	0.3957	0.3838.
LogisticRegression	0.8127	0.7545
MultinomialNB	0.6745	0.6453
SVC		

LightGBM		

实验3.

embedding(参数min-count=2，windows=3,(Train-206316,Test-29474),词个数 217943)

3.1基于w2v的文本分类结果 （词表个数：217943）

	#Train.acc	#Test.acc
RandomForestClassifier	0.4687	0.4484
LogisticRegression	0.7210	0.6683
MultinomialNB	0.434	0.459

3.2基于fast的文本分类结果

	#Train.acc	#Test.acc
RandomForestClassifier	0.4475	0.42
LogisticRegression	0.7098	0.6270
MultinomialNB	0.394	0.4067

实验4.

embedding(参数min-count=2，windows=2 (Train-

206316,Test-29474))

4.1基于w2v的文本分类结果 （26119）

	#Train.acc	#Test.acc
RandomForestClassifier	0.4661	0.4278
LogisticRegression	0.7171	0.6218
MultinomialNB	0.4459	0.4477
SVC	0.7972	0.7076
LightGBM	0.7477	0.6458

4.2基于fast的文本分类结果(19592个词表)

	#Train.acc	#Test.acc
RandomForestClassifier	0.4443	0.4156
LogisticRegression	0.706	0.5525
MultinomialNB	0.3859	0.3732
SVC	0.7597	0.6565
LightGBM	0.7317	0.6222

实验5.

embedding_all(参数min-count=1，windows=2，词个

数369974)

5.1基于w2v的文本分类结果

	#Train.acc	#Test.acc
RandomForestClassifier	0.4617	0.4464
LogisticRegression	0.7209	0.6901
MultinomialNB	0.4344	0.4486

5.2基于fast的文本分类结果

	#Train.acc	#Test.acc
RandomForestClassifier	0.4383	0.4264
LogisticRegression	0.7093	0.6457
MultinomialNB	0.4016	0.4234

实验6.

将样本中的词向量中**最大值**作为该样本的表示

6.1Word2vec_max的词向量(参数min-count=2, windows=3)

	#Train.acc	#Test.acc

RandomForestClassifier	0.3912	0.376
LogisticRegression	0.6311	0.5986
MultinomialNB	0.3519	0.3375

6.2FastText_max词向量

	#Train.acc	#Test.acc
RandomForestClassifier	0.3713	0.3631
LogisticRegression	0.5512	0.4958
MultinomialNB	0.3227	0.3208

实验7.

将样本中的词向量的**和**作为该样本的表示
7.1Word2vec_sum的词向量(参数min-count=2, windows=3)

	#Train.acc	#Test.acc
RandomForestClassifier	0.4448	0.4306
LogisticRegression	0.7137	0.6796
MultinomialNB	0.3959	0.4129

7.2FastText_sum词向量

	#Train.acc	#Test.acc
RandomForestClassifier	0.4354	0.4241
LogisticRegression	0.7032	0.6546
MultinomialNB	0.369	0.3818

FastText模型用于分类的结果

	#Train.acc	#Test.acc
系统词嵌入	0.8013	0.7389
w2v	0.7740	0.7748
fast	0.8223	0.7719

优化方案

数据方面

目前训练集数据有20万条左右，训练集3万条左右，一共有33个类别。属于文本的多分类项目。
根据实验结果分析，准确率不高在数据方面的主要原因是很多样本的标签都十分接近，很容易分错。
以TF-IDF的MultinomialNB（）分类模型对样本数据分析为例：
第一种情况：数据样本可以归纳属于多个标签，实际上却只给了一个标签。从而导致模型预测错误。

比如：医学类别被错误预测为健身保健类别（样本中含有‘药物，'艾灸', '灸法''等容易误导模型的词）；文学被预测为国学/古籍类别，动漫被错误预测为青春文学（样本中含有‘小说，浪漫爱情，唯美，仙气，’等误导模型的词），社会科学
第二种情况： 数据样本没有问题，没有歧义词。但是模型预测错误，即模型没有学会该样本。
第三种：数据中可能存在小部分样本的标签标记错误。即存在标签噪声。
比如：某个样本中含有“党的建设，党的作风，执政能力“等，该样本却被标为文学标签，但预测为”政治/军事“标签。
对于这种错误很可能需要手动进行调整。

词向量模型

- TF-IDF
  TfidfVectorizer（max_df, min_df)根据调节max_df, 和min_df二个参数，在该模型上分别基于6000个词和70000个词进行了实验（如实验1和实验2所示）。根据实验结果分析可得，实验二中的逻辑回归分类器模型准确率要高于实验一，并且在该模型上有一定程度的过拟合现象。总的来说，实验1，2的实验结果相差不大，即该模型中词的个数大小对模型的影响不是很大。

- Word2vec 和fasttext词嵌入模型

  目前设置的词向量的维度是300.可以通过min_count和max_vocab_size参数来限定词表的大小。从而去掉一些频率较低的词。

第一:这二种词嵌入模型，我们分别在词个数为20000左右（实验三），210000（实验四），369974（实验五）的条件下进行实验。根据实验三，实验四，实验五分析发现，这二种词嵌入模型在不同词个数的情况下产生的样本表示对样本分类结果影响不大。因为样本中对样本决策具有关键作用的词较小，大部分的词对样本分类的影响都比较小。因此减少或者增加词的个数来构建样本表示不会对结果有太大影响。

第二：在构建每条样本表示的时候，我们分别测试使用样本中所有词表示的平均值（mean），最大值(max)，加和(sum)作为样本的词嵌入表示。根据实验六和实验七结果显示，通过最大化的方法来构造样本表示会导致实验效果下降。这可能是因为样本表示中的每一个元素都去样本词中对应元素的最大值，从而减少了样本不同特征之间的区分度。进而导致模型更难分类正确该样本。而使用加求与平均的方法的实验结果差不多，这是因为相对样本表示中的其他特征来说，不管是对元素加和还是平均，都可以较大程度上保留样本中特征的区分度，从而使得模型的效果稳定。

第三： 根据实验结果显示，发现基于FastText的词向量表示的分类模型结果稍低于基于Word2vec的试验结果。 以实验四为例，这是由于二者在生成词向量时模型设置的参数不同导致二者的词表个数不同导致的，Word2vec模型中有26119个词表，而FastText模型中只有19592个词表。因此FastText构造的样本表示质量要稍低于Word2vec，从而导致基于FastText的分类结果略低于基于Word2vec的分类模型。然而，如果单独

看不同词表产生的FastText样本表示结果，根据实验三（词表个数217943），四（词表个数19592），五（词表个数39979）中的Fasttext结果显示，可以发现实验四中由于词表个数最少，其分类结果也最差。从而验证了词表个数会对FastText构造的样本表示影响较大。

机器学习模型

- lgb.LGBMClassifier(num_leaves=30, reg_alpha=1, reg_lambda=10, max_depth=3)
  该模型可以调整的参数很多，而且很容易产生过拟合，如果产生过拟合，可以过降低树的叶子结点个数（num_leaves）,调整L1正则化（reg_alpha）和L2正则化（reg_lambda）参数以及降低树的深度（max_depth）。

- RandomForestClassifier(n_estimators=500, max_depth=5)
  根据实验结果显示，该模型的准确率大概在0.4左右，这个结果相对来说是比较低的，可以通过调节随机森林模型中n_estimators,max_depth等参数来提升模型性能。（补充实验）

- LogisticRegression_test
  实验结果显示逻辑回归模型的测试集的准确率大概在0.7左右，但是实验中训练集的准确率通常要高于测试集，因此可以认为该模型产生了一定的过拟合现象。比

如在实验二中的TF-IDF中，逻辑回归在训练集的准确率是0.8127，在测试集上的准确率是0.7545，可以看出产生了一定的过拟合现象，因此可以在该模型中使用一些防止过拟合策略来提高模型的性能。比如正则化策略等。

- MultinomialNB_test
  根据实验结果显示，该模型的实验效果大概在0.35-0.45之间，总体上低于其他分类模型。该模型效果较差主要是因为朴素贝叶斯模型假设样本特征之间是相互独立的，因此在特征之间关联度比价高的情况下该模型表现的会比较差。

- SVC_test accuracy
  根据实验四结果显示，该分类模型的分类结果较好。可以发现SVC模型在训练集上的准确率要高于测试集，说明该模型同样具有一定程度的过拟合现象。同样可以考虑加入正则化操作来提高模型的泛化能力。

总的来说，朴素贝叶斯模型由于特征之间相互独立性的假设，导致其在大规模数据集上的效果最差。以实验四为例。分类准确率
SVC>LightGBM>LR>MultiNB>RandomForesst

深度学习模型结果

CNN文本分类结果

	#Train	#val	#Test
fast_embedding	0.8125	0.7789	0.7769
W2v_embedding			

RCNN文本分类结果

	#Train	#val	#Test
fast_embedding	0.8125	0.7644	0.7596
W2v_embedding			