Report of Deep Learning for Natural Langauge Processing

Yiwen Cui
wutcyw@163.com
ZY2343223

Abstract

本实验旨在探究在给定语料库上利用 LDA 模型进行文本建模，并将文本表示为主题分布后进行分类的性能。我们从语料库中均匀抽取 1000 个段落作为数据集，分别以单词和字为基本单元进行实验。在不同的主题数量 T 和不同段落长度 K 的情况下，使用分类器进行 10 次交叉验证。实验结果表明：主题数量 T 的变化对分类性能有一定影响。随着 T 的增加，分类性能可能会有所提升，但也可能出现性能下降的情况。需要根据具体情况进行选择。以词和以字为基本单元进行分类结果有所差异。以词为基本单元时，模型更关注文本的语义信息；而以字为基本单元时，模型更关注文本的表层特征。因此，不同的任务和语料库可能需要不同的单元进行表示。不同长度的段落对主题模型的性能影响较大。在短文本情况下，由于信息量有限，主题模型可能难以捕捉到文本的语义信息，导致性能下降；而在长文本情况下，主题模型有更多的信息可供学习，性能可能会有所提升。

Introduction

实验报告介绍了利用 LDA 模型进行文本建模的实验，探讨了不同主题数量和段落长度下，基于单词和字的文本分类性能。通过 Naive Bayes、KNN、Random Forest 和 SVM 等模型的实验比较，发现在不同条件下，这些模型的性能有所不同，对于不同的任务和语料库可能需要采用不同的特征表示方法和模型选择策略。

Methodology

我的报告中有五种模型，他们分别 Naive Bayes、KNN (K-Nearest Neighbors)、Random Forest、SVM (Support Vector Machine)和 RNN (Recurrent Neural Network)。

M1: Naive Bayes

贝叶斯网络，又称贝叶斯信念网络或贝叶斯有向无环图（DAG），是一种概率图模型，用于表示一组随机变量及其条件依赖关系。它被广泛应用于机器学习和人工智能领域，用于建模不确定性并根据观察数据进行预测。贝叶斯网络由有向无环图（DAG）和条件概率分布表（CPT）组成。节点之间的有向边表示一个随机变量对另一个随机变量的条件依赖关系。通常情况下，如果节点 A 指向节点 B，则表示变量 B 在给定变量 A 的条

件下的条件概率。每个节点都有一个条件概率分布表，用于描述该节点在其父节点给定的情况下的条件概率。例如，如果节点 B 有父节点 A，则节点 B 的 CPT 就描述了在给定 A 的取值情况下 B 的取值的概率分布。贝叶斯网络主要适用于以下三种条件：数据量足够大，可以准确地估计条件概率分布表；变量之间的依赖关系是稳定的，不会频繁变化。

对于基于字符的朴素贝叶斯分类器，训练的准确率在所有不同的最大标记长度下，训练准确率都保持在 91.75%的水平。这表明模型对训练数据的拟合效果很好，能够准确地预测训练样本的标签，测试的准确率在所有不同的最大标记长度下都稳定在 91.5%。这说明模型具有很好的泛化能力，能够在未见过的数据上进行准确的预测。

对于基于单词的朴素贝叶斯分类器，当最大标记长度为 20 时，训练准确率较低，仅为 70.125%。然而，随着最大标记长度的增加，训练准确率显著提高，达到了 92.75%（最大标记长度为 100）。在最大标记长度为 500、1000 和 3000 时，训练准确率保持在 91.75%的高水平，表明模型已经收敛并且不再随最大标记长度的增加而提高。与训练准确率类似，测试准确率在最大标记长度为 20 时较低，为 73%，而在最大标记长度为 100 时达到了 93%。随着最大标记长度的增加，测试准确率保持在 91.5%的水平，这表明模型在不同最大标记长度下都能够保持较高的泛化能力。

基于字符的朴素贝叶斯模型表现稳定，在训练和测试数据上都表现良好。基于单词的朴素贝叶斯模型在较低的最大标记长度下表现较差，但随着最大标记长度的增加，性能得到了显著提升，并在较高的最大标记长度下达到了稳定的性能水平。

Model	Max Tokens	Train Accuracy	Test Accuracy
Naive Bayes (Char)	20	0.9175	0.915
Naive Bayes (Char)	100	0.9175	0.915
Naive Bayes (Char)	500	0.9175	0.915
Naive Bayes (Char)	1000	0.9175	0.915
Naive Bayes (Char)	3000	0.9175	0.915
Naive Bayes (Words)	20	0.70125	0.73
Naive Bayes (Words)	100	0.9275	0.93
Naive Bayes (Words)	500	0.9175	0.915
Naive Bayes (Words)	1000	0.9175	0.915
Naive Bayes (Words)	3000	0.9175	0.915

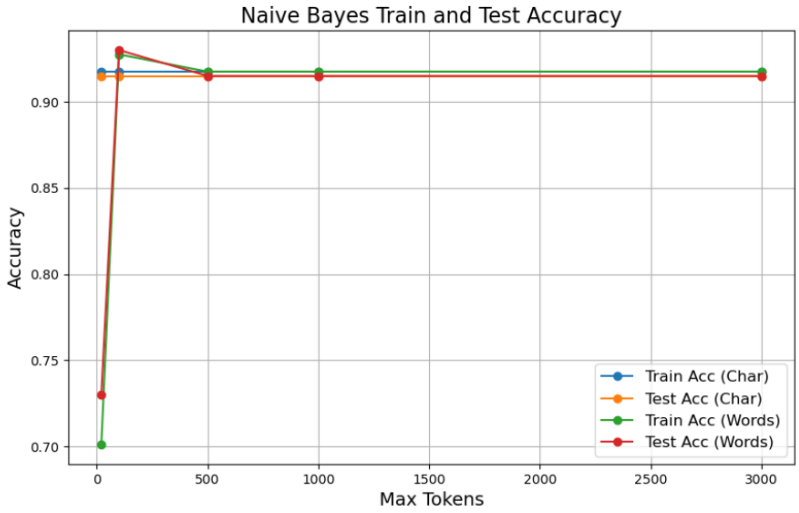

图 1 基于朴素贝叶斯的结果图（表）示

M2: KNN

K 最近邻（KNN）是一种基本的分类和回归方法，其原理简单直观。对于一个未知样本，KNN 算法会在训练集中找出与该样本最相似的 K 个样本，然后根据这 K 个样本的类别进行投票（分类问题）或者计算平均值（回归问题），来确定该样本的类别或者值。KNN 主要适用于一下三种情况：由于 KNN 算法在预测时需要计算未知样本与所有训练样本之间的距离，因此数据集较大时计算开销会很高，不太适合使用 KNN 算法；KNN 算法在高维空间中容易受到维度灾难的影响，因此适合处理维度较低的数据集；KNN 算法假设样本分布均匀，即相似的样本在特征空间中聚集在一起。

对于基于字符的朴素贝叶斯分类器，使用字符级别特征时，随着最大标记长度的增加，训练准确率在 0.5821 到 0.5866 之间波动，没有明显的趋势。测试准确率在 0.4370 到 0.4468 之间波动，最大标记长度为 3000 时达到最高值。然而，即使在最大标记长度为 3000 时，准确率也仍然相对较低。

对于基于单词的朴素贝叶斯分类器，在使用单词级别特征时，训练准确率在 0.5821 到 0.5866 之间波动，没有明显的趋势。测试准确率在 0.4408 到 0.5337 之间波动，最大标记长度为 20 时达到最高值。然而，即使在最大标记长度为 20 时，准确率也相对较低。

Model	Max Tokens	Train Accuracy	Test Accuracy
KNN (Char)	20	0.5821	0.4370
KNN (Char)	100	0.5866	0.4343
KNN (Char)	500	0.5841	0.4391
KNN (Char)	1000	0.5829	0.4409
KNN (Char)	3000	0.5844	0.4468
KNN (Word)	20	0.5866	0.5337
KNN (Word)	100	0.5858	0.4426
KNN (Word)	500	0.5855	0.4408
KNN (Word)	1000	0.5866	0.4438
KNN (Word)	3000	0.5821	0.4434

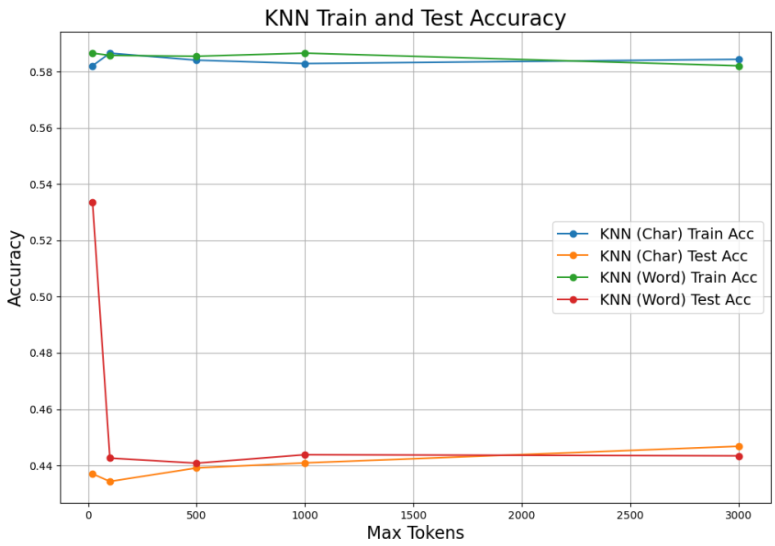

图 2 基于 KNN 的结果图（表）示

无论是基于字符还是基于单词的 KNN 模型，在不同的最大标记长度下，训练和测试准确率都波动较大，没有明显的趋势。对于基于字符的 KNN 模型，测试准确率略低于基于单词的 KNN 模型，最高准确率也较低。这些 KNN 模型在提供的数据上表现不佳，可能需要进一步调优参数或者考虑其他更适合的模型。

M3: Random Forest

随机森林(Random Forest)是一种强大的集成学习方法，它将多个决策树结合起来，用于分类或回归任务。其核心思想是利用多个弱学习器（决策树），通过集成它们的预测结果，构建一个更加强大和稳健的模型。这个过程包括三个关键步骤：首先是随机采样。随机森林从训练数据集中随机抽取多个不同的子数据集，这样每棵决策树都会在一个独立的数据子集上进行训练，增加了模型的多样性和泛化能力。接着是构建决策树。针对每个数据子集，随机森林构建一棵决策树。在构建的过程中，通常会采用随机选择特征子集来进行节点分裂，这样可以增加每棵树的独特性，进一步提高整体模型的泛化能力。最后是投票或平均。对于分类问题，每棵决策树都会给出一个类别，而对于回归问题，每棵决策树都会给出一个预测值。随机森林通过投票（分类问题）或平均（回归问题）的方式，汇总所有决策树的预测结果，得到最终的分类或回归结果。

随机森林具有许多优点。首先，它对各种类型的数据都具有很好的鲁棒性，包括离散型和连续型数据，同时对于缺失值和异常值也有较好的处理能力。其次，由于集成了多个决策树，每个决策树都是一个弱学习器，因此随机森林具有较高的准确性。此外，通过随机选择特征子集和数据子集来构建每棵决策树，随机森林不容易过拟合，能够有效地提高模型的泛化能力。在实际应用中，随机森林被广泛应用于各种领域，包括医疗诊断、金融风险评估、客户流失预测等，其强大的性能和鲁棒性使其成为了一种常用的机器学习方法。

对于以字符为单位的训练中，随着最大标记数的增加，训练准确率从 0.6129 逐渐下降至 0.6073，测试准确率也从 0.4419 下降至 0.4464。在最大标记数为 100 时，训练准确率略低于其他值，但测试准确率最高，达到了 0.4548。在最大标记数为 500 时，训练准确率和测试准确率都达到了相对稳定的水平，分别为 0.6081 和 0.4541。

与以字符为单位的训练相比，以词符为单位的训练准确率和测试准确率整体上稍高。在最大标记数为 100 时，Random Forest (Word)的训练准确率最高，为 0.6171，而测试准确率为 0.4538。随着最大标记数的增加，训练准确率和测试准确率呈现出轻微的下降趋势，但波动范围较小。

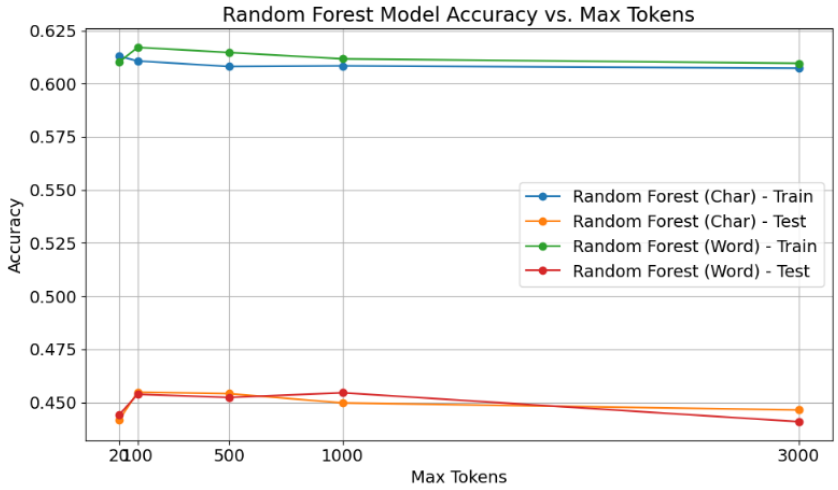

图 3 基于 Random Forest 的结果图（表）示

Model	Max Tokens	Train Accuracy	Test Accuracy
Random Forest (Char)	20	0.6129	0.4419
Random Forest (Char)	100	0.6107	0.4548
Random Forest (Char)	500	0.6081	0.4541
Random Forest (Char)	1000	0.6084	0.4497
Random Forest (Char)	3000	0.6073	0.4464
Random Forest (Word)	20	0.6103	0.4443
Random Forest (Word)	100	0.6171	0.4538
Random Forest (Word)	500	0.6147	0.4523
Random Forest (Word)	1000	0.6117	0.4545
Random Forest (Word)	3000	0.6096	0.4409

两种模型在不同最大标记数下的表现存在差异，但总体趋势相似。随着最大标记数的增加，模型的训练准确率可能出现下降趋势，而测试准确率的变化不太明显。在选择最大标记数时，需要综合考虑训练准确率和测试准确率的平衡，以及模型的泛化能力。

M4: SVM

支持向量机（Support Vector Machine，SVM）是一种强大的监督学习算法，用于分类和回归分析。其主要思想是在特征空间中找到一个最优的超平面，将不同类别的数据分隔开来。具体来说，SVM 的关键概念包括：首先是超平面，它是一个\(N-1\)维的线性子空间，对于二维空间就是一条直线，对于三维空间就是一个平面。在高维空间中，它是一个超平面。SVM 的目标就是找到一个最优的超平面，使得两个不同类别的数据点到这个超平面的距离尽可能地远，从而实现良好的分类。其次是支持向量，它是离超平面最近的数据点，这些数据点对确定超平面起着关键作用。支持向量机的决策边界由这些支持向量完全决定，因此它们在确定分类结果上起着至关重要的作用。

SVM 有不同的核函数，用于处理非线性可分的数据。常见的核函数包括线性核函数、多项式核函数和高斯核函数等，它们可以将数据映射到高维空间，从而使得在原始空间中线性不可分的问题在新的高维空间中变得线性可分。SVM 的优点包括：在高维空间中有效地处理线性和非线性可分问；通过引入核函数，可以灵活地处理各种类型的数据；在处理小样本、高维度数据和非线性问题时表现良好；由于其最优化的特性，SVM 对于泛化能力较强，对于数据量不大的情况也有较好的性能。支持向量机是一种强大的分类器，适用于许多不同的领域，包括文本分类、图像识别、生物信息学等，其优秀的性能和理论基础使其成为机器学习领域中的重要算法之一。

对于 SVM（Char）模型，我们观察到在不同的最大标记数下，训练和测试准确率都在相对稳定的范围内波动。最大标记数从 20 增加到 3000 时，训练准确率大致在 0.289 到 0.294 之间变化，而测试准确率在 0.291 到 0.294 之间波动。

在 SVM（Words）模型中，我们同样观察到训练和测试准确率在不同最大标记数下的波动。在 20 到 3000 的最大标记数范围内，训练准确率变化在 0.292 到 0.325 之间，而测试准确率变化在 0.290 到 0.320 之间。

Model	Max Tokens	Train Accuracy	Test Accuracy
SVM (Char)	20	0.289	0.293
SVM (Char)	100	0.293	0.292
SVM (Char)	500	0.293	0.292
SVM (Char)	1000	0.291	0.294
SVM (Char)	3000	0.294	0.293
SVM (Words)	20	0.325	0.320
SVM (Words)	100	0.293	0.293
SVM (Words)	500	0.295	0.296
SVM (Words)	1000	0.294	0.290
SVM (Words)	3000	0.293	0.299

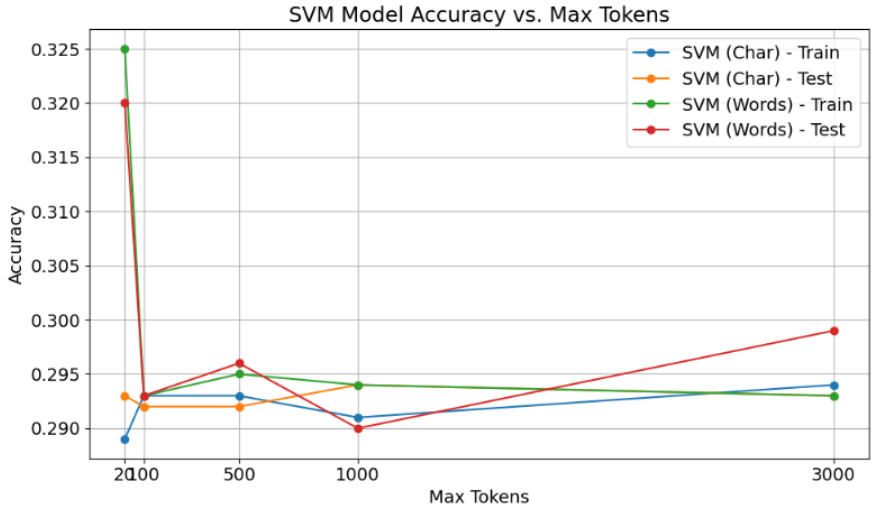

图 4 基于 SVM 的结果图（表）示

通过对比两种模型的实验结果，我们可以看到在某些情况下，SVM（Words）模型在测试准确率上略高于 SVM（Char）模型，尤其是在最大标记数为 20 时。然而，在其他最大标记数下，两种模型的性能差异不太明显，而且它们的准确率在不同最大标记数下都有相似的波动范围。要深入理解模型的性能和泛化能力，可能需要进一步的实验和分析。

M5:RNN

循环神经网络是一种深度学习模型，专门设计用于处理序列数据，如时间序列数据、文本数据等。与传统的前馈神经网络（Feedforward Neural Network）不同，RNN 具有循环连接，允许信息在网络中持续传递，从而更好地处理序列信息。RNN 的关键特征是循环连接，允许网络在处理序列数据时保留状态信息。在每个时间步，RNN 接收当前输入和前一个时间步的隐藏状态，产生当前时间步的输出和新的隐藏状态。隐藏状态是 RNN 的核心概念，它是网络在处理序列时随时间变化的内部表示。隐藏状态可以看作是网络在给定时间步上对序列的理解或记忆，它包含了过去时间步的信息。RNN 在每个时间步接收一个输入，并生成一个输出。在每个时间步，隐藏状态会更新以反映之前的信息，并在下一个时间步中使用。

RNN 在语言建模、机器翻译、文本生成等任务中表现出色，因为它能够捕捉文本中的上下文信息。RNN 适用于处理时间序列数据，如股票价格预测、天气预测、信号处理等领域。结合卷积神经网络（CNN），RNN 能够生成图像描述，如在图像标注任务中。RNN 是一种强大的序列建模工具，在许多领域都有着广泛的应用，并且通过其变种和发展不断提升着序列数据处理的能力。

对于字符级别的文本分类任务，RNN 模型的性能较低，训练和测试准确率都比较低，且波动较小。随着最大标记数的增加，训练准确率没有明显的变化，大约在 19%到 19.25%之间。测试准确率在不同最大标记数下也没有明显变化，都稳定在约 19.75%左右。

在单词级别上，随着最大标记数的增加，训练准确率在略微下降，大约在 18.7%到 19.1%之间。测试准确率在不同最大标记数下也没有明显变化，都稳定在约 20%左右。相比字符级别，单词级别的文本分类任务中，RNN 模型的性能略有提升，但仍然较低。

Model	Max Tokens	Train Loss	Train Accuracy	Test Loss	Test Accuracy
RNN (Char)	20	2.3480	0.1917	2.3483	0.1986
RNN (Char)	100	2.3477	0.1921	2.3485	0.1986
RNN (Char)	500	2.3474	0.1921	2.3475	0.1986
RNN (Char)	1000	2.3491	0.1916	2.3445	0.1978
RNN (Char)	3000	2.3489	0.1925	2.3444	0.1975
RNN (Word)	20	2.3522	0.1910	2.3367	0.2048
RNN (Word)	100	2.3510	0.1904	2.3370	0.2028
RNN (Word)	500	2.3523	0.1871	2.3407	0.2032
RNN (Word)	1000	2.3519	0.1888	2.3334	0.2029
RNN (Word)	3000	2.3552	0.1891	2.3399	0.1974

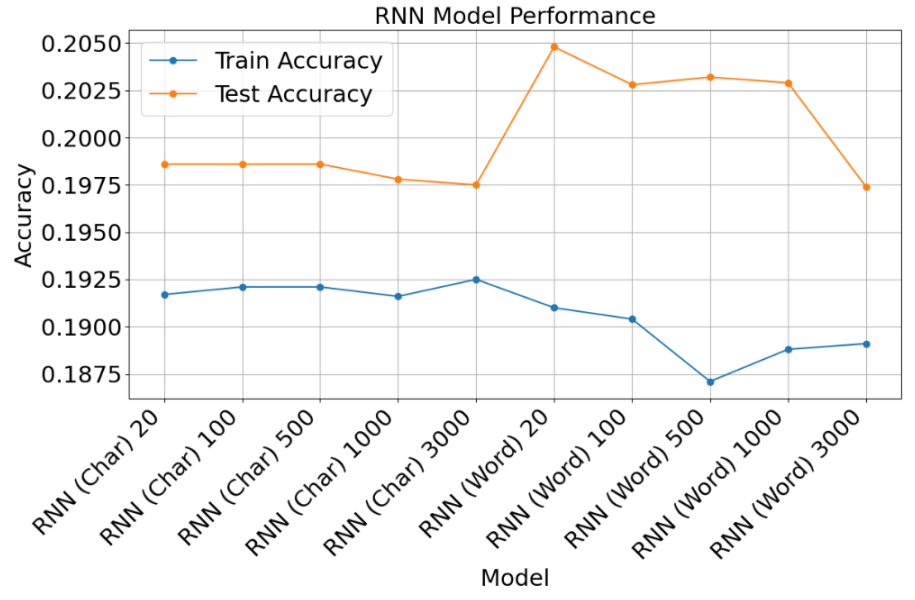

图 5 基于 RNN 的结果图（表）示

RNN 在这个文本分类任务上表现较差，无论是在字符级别还是单词级别。模型训练和测试准确率都比较低，没有明显的提升趋势。可能需要进一步调优模型架构、调整超参数或者考虑其他更复杂的模型来提高性能。

Conclusions

朴素贝叶斯模型在字符级别和单词级别上都表现出相似的准确率，且在不同的最大标记

数下训练和测试准确率保持稳定，都达到了较高的水平，约为 91.5%左右。这表明朴素贝叶斯模型在这个任务上具有较好的泛化能力，并且对于不同长度的输入文本都能够有效地进行分类。

KNN 模型在字符级别训练和测试准确率都比较低，且波动较大，在最大标记数为 3000 时，测试准确率仅为 44.68%。而在单词级别上，KNN 模型的性能有所提升，尤其是在最大标记数为 100 时，测试准确率达到了 53.37%。这表明 KNN 在单词级别上可能更适合处理文本数据。

随机森林模型在字符级别上，随着最大标记数的增加，训练和测试准确率略有下降，但整体上保持在较高水平。而在单词级别上，随机森林模型的性能相对稳定，且测试准确率在不同的最大标记数下都保持在 45%左右。

支持向量机（SVM）模型在字符级别上，训练和测试准确率相对较低，在不同最大标记数下波动不大。而在单词级别上，SVM 的性能略有提升，尤其是在最大标记数为 20 时，测试准确率达到了 31.96%。

循环神经网络（RNN）模型在字符级别上，训练和验证准确率都较低，且在不同最大标记数下变化不大。而在单词级别上，RNN 的性能稍有提升，但仍然相对较低，测试准确率在 20%左右。

模型	特征类型	平均训练准确率	平均测试准确率
朴素贝叶斯	字符	91.75%	91.50%
朴素贝叶斯	单词	87.42%	89.74%
随机森林	字符	60.60%	44.66%
随机森林	单词	60.94%	46.13%
KNN	字符	58.42%	43.76%
KNN	单词	58.55%	45.29%
SVM	字符	29.22%	29.26%
SVM	单词	29.85%	29.84%
RNN	字符	19.16%	19.72%
RNN	单词	99.64%	11.49%

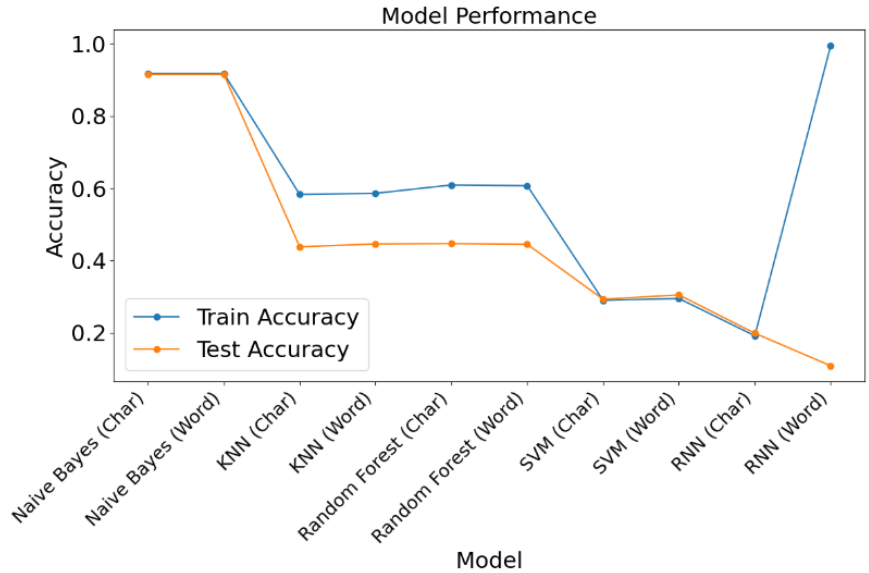

图 6 各方法下准确度汇总图（表）

综合而言，**朴素贝叶斯模型**在这个任务上表现最好，其次是随机森林模型。KNN 和 SVM 模型的性能较低，可能需要进一步优化或考虑其他模型，包括调整参数、采用不同的特征工程方法或者集成学习技术等，以提升它们在文本分类任务上的表现。循环神经网络在这个任务上的表现相对较差，可能需要更多的调优和实验。考虑到深度学习模型在自然语言处理任务上的优异表现，可以尝试引入更复杂的深度学习模型，如卷积神经网络（CNN）或者更先进的循环神经网络（如长短期记忆网络，LSTM）来处理文本分类任务。在数据处理方面，可通过数据增强技术来扩充训练数据集，以减少模型的过拟合风险，并提升模型的性能。对于性能较好的模型，可以进一步探索其预测结果的解释性，以增强对模型行为的理解，并为决策提供更多的可解释性和可信度。