Report #2 of Deep Learning for Natural Langauge Processing

孟逸飞
mengyifei@edu.iwhr.com

Abstract

本研究的目的是探讨使用 LDA 主题模型对中文文本进行分类的有效性，并分析不同主题数、基本单位（词和字）、以及段落长度对模型性能的影响。首先，通过对来自经典中文小说的大型语料库进行系统的文本预处理，包括去除非中文字符、分词、并删除停用词，从而构建了一个包含 1000 个段落的数据集。每个段落被标记为其所属小说的标题。然后，利用生成的数据集训练了 LDA 模型，并将每个段落表示为主题分布。这些主题分布随后被用作输入特征，通过随机森林分类器进行文本分类。采用 10 次交叉验证方法评估模型的分类性能。实验结果表明，随着主题数量的增加，分类精度存在变化；不同的基本单位（词与字）对分类结果有显著影响；段落长度对主题模型的性能也显示出差异。

Introduction

在自然语言处理（NLP）领域，文本分类是一个基础且关键的任务，广泛应用于情感分析、主题识别、新闻归类等多种场景。随着机器学习技术的发展，多种模型被提出以提高文本分类的准确性和效率。潜在狄利克雷分配（LDA）作为一种主题模型，因其在文本数据上提取主题的能力而被广泛研究和使用。尽管 LDA 模型主要用于文档级别的主题发现，其在文本分类中的潜力和表现仍然是研究的重要内容。

本研究旨在探讨 LDA 模型在中文文本分类上的应用效果，特别是考察模型在处理不同长度、不同基本单位（词和字）的文本时的表现。中文文本处理具有其特殊性，如分词的不确定性和语言的结构复杂性，这些特点使得模型的配置和选择显得尤为重要。

此外，本研究通过分析不同主题数量的设置对分类效果的影响，旨在找出最优的模型参数配置，以提升分类性能。通过构建一个均衡的数据集，每个段落根据其所属的小说进行标注，并使用这些数据来训练和验证 LDA 模型结合随机森林分类器的组合。

Methodology

1 数据集的构建与预处理(tokenize_extract_words.py)

为了保证分类任务的准确性和有效性，数据预处理是关键步骤。

1.1 停用词的加载与去除：

load_stopwords 函数的核心是创建一个集合（Set），其中包含所有从指定文件中读取的停用词。停用词通常包括最常见的、语义贡献较低的词，如“的”、“了”等。这一步是通过打开文件、读取内容并将其分割成单独的词汇来完成的，最终这些词汇存储在一个集合中以便快速查找和过滤。

1.2 文本预处理：

preprocess_text 函数的目标是将原始文本转换为清洗后的、分词的 Token 列表。首先，使用正则表达式替换掉所有非中文字符，确保处理的文本纯粹是中文。接着，利用 jieba 库

对清洗后的文本进行分词。最后，遍历分词结果，并过滤掉所有包含在停用词集中的词汇。这个过程保证了输出的 Token 是具有实际意义的中文词汇。

1.3 段落抽取：

extract_paragraphs 函数实现了将一长串 Token 分割为指定数量和长度的段落。函数首先计算每个段落应含 Token 的数量，然后按此规模进行切分。该步骤关键在于保证每个段落内 Token 的数量符合预设标准，同时保持段落数量的均衡，这对后续的模型训练极为重要。

2、主题建模、特征提取、分类与评估（LDA_RF.py）

2.1 LDA 模型训练

潜在狄利克雷分配（LDA）模型是一种统计模型，用于发现大量文档集合中的共享主题。在本研究中，我们使用 LDA 模型来提取文本数据的潜在主题分布，作为特征用于后续的分类任务。

字典和语料库的创建：首先，利用所有文本数据创建一个 Dictionary 对象，该对象将每个独特的词映射到一个整数 ID。随后，将文本转换成词袋模型，每个文本被表示为一个词的频率向量。

模型训练：利用 gensim 的 LdaModel 类，我们根据语料库和字典训练 LDA 模型。通过设置 num_topics 参数，我们指定希望模型学习的主题数量。passes 参数控制模型在整个语料库上迭代的次数，以确保模型的收敛和稳定性。

主题分布的提取：训练完成后，每个文档的主题分布可以通过模型转换其对应的词袋向量来获取。这些主题分布向量将用作后续分类任务的特征输入。

2.2 随机森林分类器

随机森林是一种强大的集成学习方法，用于分类和回归。它操作简单且通常能达到很高的准确度。

分类器初始化：使用 sklearn.ensemble.RandomForestClassifier，初始化一个随机森林分类器。此分类器将用于基于文档的主题分布特征来预测文档的类别。

参数优化：为了找到最优的分类器配置，使用 RandomizedSearchCV 进行随机参数搜索。定义参数分布 param_dist，包括树的最大深度、拆分所需的最小样本数、叶节点的最小样本数等。这个过程将自动测试多种随机森林配置，以找到最佳的参数组合。

模型训练和评估：通过 RandomizedSearchCV 在训练数据上训练多个随机森林模型的不同配置，并使用交叉验证来评估每个模型的性能。

2.3 交叉验证

交叉验证是一种评估模型泛化能力的方法，特别是在数据集不是非常大的情况下。

数据分割：使用 StratifiedKFold 确保每次分割后各类别样本的比例保持一致，特别是在处理类别不平衡的数据集时。将数据集分为 10 个子集，每个子集保持原始数据类别的比例。

迭代过程：在 10 次迭代中，每次迭代选择一个子集作为测试集，其余九个子集作为训练集。这样可以确保每个数据点都有一次作为测试集的机会。

学习与验证：每次迭代中，使用训练集数据来训练模型，并在测试集上评估其性能。记录每次迭代的评估指标，如准确率。

总结结果：所有迭代完成后，计算所有单次迭代的评估指标的平均值，得到模型的交叉

验证得分。这种方法有效地利用有限的数据进行多次独立的训练和测试，提高评估的稳定性和可靠性。

2.4 可视化 LDA 模型

使用 pyLDAvis 库对 LDA 模型进行可视化，这有助于更好地理解模型学到的主题和它们之间的关系。通过可视化，研究者可以直观地看到每个主题的词分布和主题间的区分度，这对于调整模型参数和解释模型结果都非常有帮助。

Experimental Studies

1. 不同主题数量（T）对分类性能的影响

我们实验的第一个方面旨在探究不同主题数量（T）对基于 LDA 的文本建模方法的分类性能的影响。我们通过设置不同的 T 值进行实验，并使用 10 折交叉验证评估所得到的分类性能。

主题数量（T）	**交叉验证准确率**
5	0.155
10	0.199
15	0.181
20	0.179
25	0.187
30	0.153
50	0.143
100	0.106
200	0.102
500	0.094
1000	0.158
2000	0.064

从结果可以看出，分类性能随着主题数量的不同呈现出波动的趋势。一开始，性能随着主题数量的增加而提高，在 T=10 左右达到峰值，随后随着 T 的继续增加而逐渐降低。这表明对于我们的分类任务来说，存在一个最佳的主题数量，在此之外增加更多的主题可能会引入噪音并降低模型的效果。

2. 基于词和字符的基本单位分类结果的差异

我们实验的第二个方面旨在比较使用词和字符作为文本表示的基本单位时所获得的分类结果。我们在 T=10 的情况下进行实验，并评估基于词和字符表示的分类性能。

主题数量（T）	**文本长度（K）**	**交叉验证准确率**	
		按词	**按字符**
T=10	K=100	0.171	0.264
T=10	K=500	0.376	0.533

结果表明，基于字符的表示优于基于词的表示，这表明字符级别的信息捕获了文本中更细粒度的细节和差异，对于分类任务是有益的。

3. 文本长度（K）对主题模型性能的影响

最后，我们探讨了文本长度（K）对我们的主题建模方法性能的影响。我们分析了不同长度文本的分类性能，从短到长进行了对比。

文本长度（K）	交叉验证准确率
20	0.103
100	0.171
500	0.300
1000	0.268
3000	0.686

结果显示，文本长度对主题模型的性能有显著影响。随着文本长度的增加，分类性能通常会提高。这表明长文本提供了更多的上下文和信息，使得模型更能够捕捉到文本的语义特征，从而提高分类准确率。

Conclusions

通过本研究，我们对文本建模和分类的关键因素进行了探究，并取得了以下结论：

主题数量是影响分类性能的重要因素。我们发现在适当范围内增加主题数量可以提高分类性能，但过多的主题数量可能会导致性能下降。使用字符级别的表示比词级别的表示更有利于分类任务，这表明字符级别的信息更丰富、更细致。文本长度对主题模型的性能有显著影响。随着文本长度的增加，分类性能通常会提高，因为长文本提供了更多的语义信息和上下文。

综上所述，研究结果为进一步优化文本建模和分类方法提供了重要的参考，有助于提高模型的性能和泛化能力。