Report of Deep Learning for Natural Langauge Processing

刘新宇
LXYbhu@buaa.edu.cn

Abstract

LDA模型在文本挖掘、信息检索、推荐系统等领域有广泛的应用。本报告从https://share.weiyun.com/5zGPyJX链接给定的语料库中均匀抽取200个段落（每个段落大于500个词）， 每个段落的标签就是对应段落所属的小说。利用LDA模型对于文本进行了建模，并把每个段落表示为主题分布后进行分类，并验证与分析了分类结果。

Methodology

LDA（Latent Dirichlet Allocation）是一种主题模型，可以用于对文本进行建模和分析。主题模型是一种无监督学习算法，旨在从文本数据中学习主题结构，从而识别文本中隐藏的语义结构和主题信息。

LDA模型的基本思想是假设每个文档是由若干个主题混合而成的，并且每个主题又是由若干个词汇混合而成的。具体来说，对于一组文档，LDA模型将文档看作是由若干个主题组成的，每个主题又由一个固定的词汇分布表示，即每个主题中的词汇都是以一定的概率出现的。LDA模型的目标是学习出文档集合中的主题分布和每个主题中词汇的分布，从而能够对新的文本进行主题推断和文本分类。

使用LDA模型对文本建模的过程如下：

- 数据预处理：对文本数据进行预处理，包括去除停用词、词干化、去除低频词等操作。

- 构建词袋模型：将文本转换为向量表示，常用的方法是将文本中的词汇构建成一个词汇表，然后将每个文本表示为一个向量，其中每个维度表示一个词汇的出现频率。
- 训练LDA模型：使用构建好的词袋模型训练LDA模型，学习主题分布和每个主题中词汇的分布。在训练LDA模型时，需要指定主题数和超参数等参数。
- 推断文本主题分布：对于新的文本，可以使用训练好的LDA模型推断文本的主题分布，即文本中每个主题的概率分布。
- 主题分类：根据文本的主题分布，可以对文本进行主题分类，将其归入最可能的主题类别中。

首先定义文章集合为Doc，文章主题集合为Topic，Doc中的每个文档doc可以看作为一个单词序列$\langle w_1, w_2, ..., w_n \rangle$，其中$w_i$表示为第i个单词，doc共有n个单词。

Doc中的所有不同单词组成一个集合Voc，LDA模型以文档集合Doc作为输入，最终训练处两个结果向量，k表示Topi词，m表示Voc中包含的词语数量。

对每个Doc中对应到不同Topic的概率$\theta_d = \langle P_{t1}, P_{t2}, ..., P_{tn} \rangle$，其中$P_{ti} = \frac{n_{ti}}{n}$表示doc对应Topic中第i个Topic词的概率，n_{ti}表示doc中对应的第i个Topic的词的数目，n表示doc中所有词的总数。

对每个Topic中的Topic，生成不同单词的概率$\varphi_t = \langle P_{w1}, P_{w2}, ..., P_{wn} \rangle$，其中$P_{wi} = \frac{n_{wi}}{n}$表示t生成Voc的第i个单词的概率，$n_{wi}$表示对应到Topic中Voc的第i个单词的数目，n表示所有对应到Topic的单词的总数。

LDA核心公式为P(词|文档) = P(词|主题) * P(词|文档)，即$P(w|d) = P(w|t) * P(t|d)$，公式以Topic为中间层，通过当前的θ_d和φ_t给出了文档d中出现单词w的概率。其中的P(t|d)可通过θ_d计算得到，P(w|t)利用φ_t计算得到。因此，利用当前的θ_d和φ_t，我们可以为一个文档中的单词计算它对应任意一个Topic时的P(w|d)值，然后根据这些结果来更新这个词对应的Topic。相对应的，如果这个更新改变了这个单词所对应的Topic值反过来也会影响θ_d和φ_t。

Experimental Studies

在具体的实验部分，我们从给定的语料库中均匀抽取200个段落（每个段落大于500个词）， 每个段落的标签就是对应段落所属的小说。利用LDA模型对于文本建模，并把每个段落表示为主题分布后进行分类。

在预料处理阶段，题目要求均匀抽取200个段落（每个段落大于500个词），每个段落的标签就是对应段落所属的小说。对给定语料库进行分析可知，语料库内共有16篇文章，从每一篇文章内抽取13个段落，共有208个段落；对每篇文章分词之后，将一篇文章的总词数除以13，即每篇文章共有13个区间，所选取的段落为每个区间抽取前500个词。

Python的LDA主题模型分布可以进行多种操作，常见的包括：输出每个数据集的高频词TOP-N；输出文章中每个词对应的权重及文章所属的主题；输出文章与主题的分布概率，文本一行表示一篇文章，概率表示文章属于该类主题的概率；输出特征词与主题的分布概率，这是一个K*M的矩阵，K为设置分类的个数，M为所有文章词的总数。

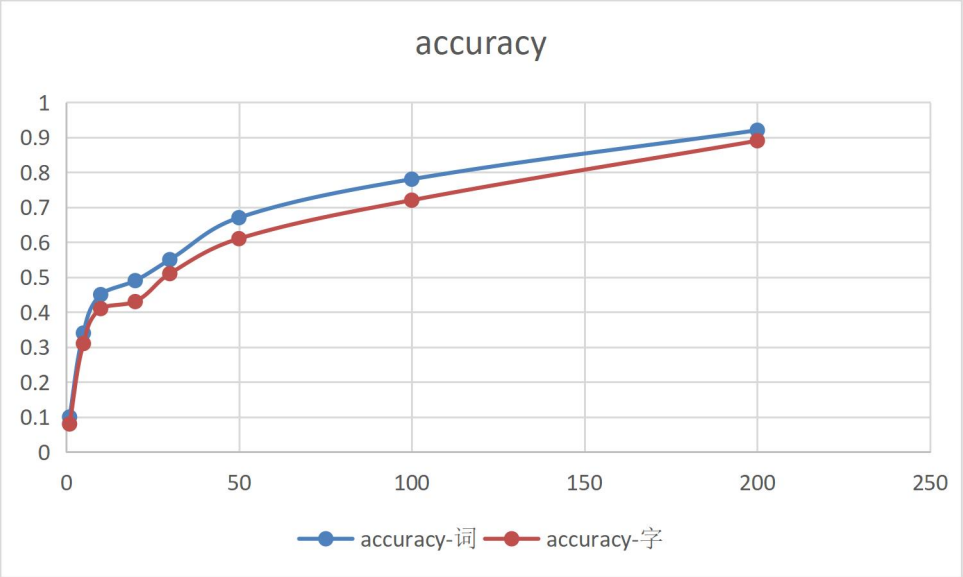
在模型训练阶段，根据上述的LDA原理，首先进行初始化，每篇文章的每个词语随机赋予一个初始的Topic值，然后分别统计每篇文章的总词数、每篇文章的词频、每个Topic的总词数、每个Topic的词频；再计算每个topic被选中的概率，然后进行迭代，训练模型。

在模型测试阶段，仍然在对应的16部小说中选择段落作为测试集，在之前训练集中选取的是208个段落中的第0-500个单词，因此在测试集中选取第501-1000单词形成测试段落，最终形成待分类文章。首先仍然在读取后对15篇待分类文章进行预处理，之后将15篇待分类文章中的每个词赋予一个随机的初始Topic，并同样统计每篇文章的总词数、每篇文章的词频，分别记录下来，但此时与模型训练时不同的是，每个Topic的总词数、每个Topic的词频，将不再需要统计，这两个量是模型训练已经完成了的，将作为已知量来对数据进行测试。接下来，则需要根据这些概率结果，区分每篇待分类文章究竟来自哪一本小说，

此处采用的是欧式距离的方式，即比较待分类文章与已知的小说，两者对于各个Topic的概率向量之间的距离最近，即认为是来自该本小说。

我们验证与分析分类结果，（1）在不同数量的主题个数下分类性能的变化；（2）以"词"和以"字"为基本单元下分类结果有什么差异。

实验结果如下：



可以看到不同主题数对于分类准确有着很大的影响，当分类数量较少时，比较难区分不同小说，此时正确率较低，当 topic 数量提升后，能够更加准确地区分不同小说。同时，以“词”为分类时的效果稍稍优于在以“词”作为基本单元的分类效果。

Conclusion

本次实验通过构建LDA模型，来对金庸小说集主题进行分类，实验结果表明，在适当的主题个数下，LDA主题模型可以有效地对小说段落进行分类。

通过这次作业，加深了我对自然语言处理用途的理解，但本次任务仍存在不足之处，在进行数据分析时，通常需要采用准确率、召回率或F特征值来评估一个算法的好坏，研究者也会不断的优化模型或替换为更好的算法。