机器学习（进阶）纳米学位

文档归类 开题报告

吕奇峰

开题报告

项目背景

自然语言处理（NLP）[1]是人工智能极为重要的一部分。

是计算机科学，人工智能，语言学关注计算机和人类（自然）语言之间的相互作用的领域。因此，自然语言处理是与人机交互的领域有关的。在自然语言处理面临很多挑战，包括自然语言理解，因此，自然语言处理涉及人机交互的智能。在 NLP 诸多挑战涉及自然语言理解，即计算机源于人为或自然语言输入的意思，和其他涉及到自然语言生成。

机器学习下的自然语言处理不同于一般的语言处理算法。在数据量不足，计算机算力也不大的时候，人们普遍大规模的使用规则模型去理解语意。将注意力主要集中在规则上，不同的规则将会输出不同的结果。而机器学习则改变着这种形态。机器学习通过海量的数据，淡化人为的规则，将更多的注意力放在了数据本身。通过巨大的算力，进行巨量的运算而自然生成自然语言的模型。这种做法为自然语言处理带来了跨越性的进展，突破了传统的瓶颈。

一些最早使用的算法，如决策树，产生硬的 if-then 规则类似于手写的规则，是再普通的系统体系。然而，越来越多的研究集中于统计模型，这使得基于附加实数值的权重，每个输入要素柔软，概率的决策。此类模型具有能够表达许多不同的可能的答案，而不是只有一个相对的确定性，产生更可靠的结果时，这种模型被包括作为较大系统的一个组成部分的优点。

自然语言处理研究逐渐从词汇语义进一步的到叙事的理解。然而人类水平的自然语言处理，是一个人工智能的极限问题。它是相当让人工智能和人一样聪明。这是自然语言处理的未来，因此密切结合人工智能发展。

问题描述

自然语言处理有个非常复杂的问题解决过程。首先是一个难点是分词，只是分词现在已经有非常多的库可以使用，也已经有相对比较成熟的算法，得到比较好的成功率。分词对于不同语言的含

义是不一样的，比如中文，正向思维是把一个句子分成短语，短语分成词。而逆向思维则是字组成词，词组成短语，短语组成句子。而英文则不太一样，因为的最小单位是字母，字母组成词，词组成习惯短语，然后再组成句子。

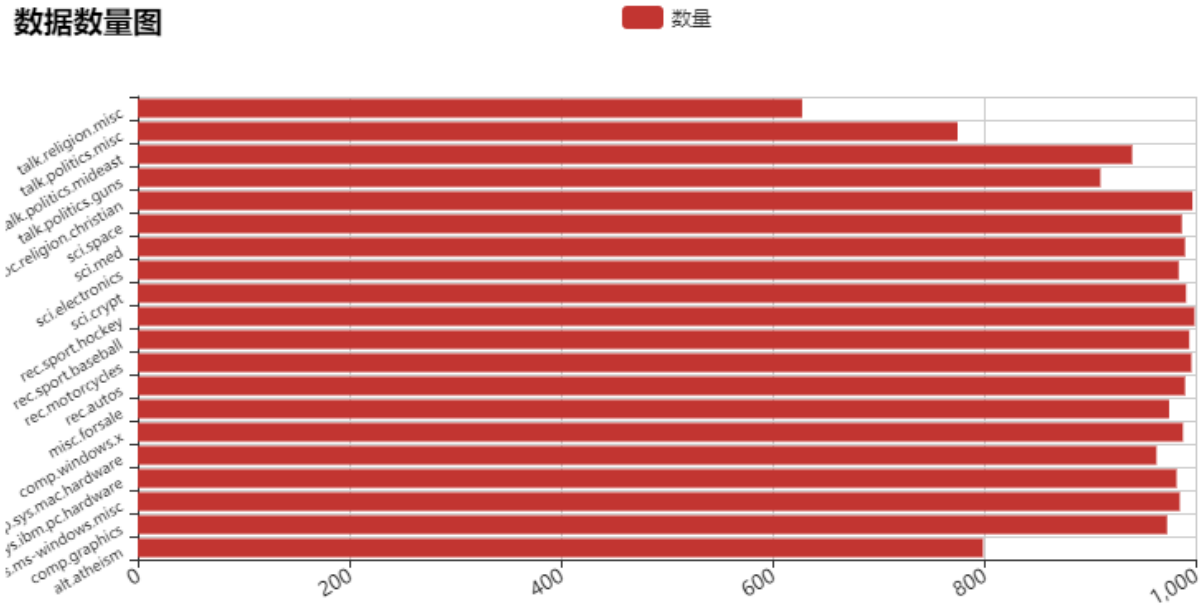
分词之后就是词、语句以及文章的表达。以英文为例，最常见的词语表述方式比如”cat“、”dog“，这些都是利用字母表示意思。统计语言处理里面，比较容易利用字母来描述概率模型，比如 ngram 模型 ，计算两个单词或者多个单词同时出现的概率，但是这些符号难以直接表示词与词之间的关联，也难以直接作为机器学习模型输入向量。对句子或者文章的表示，可以采用词袋子模型，即将段落或文章表示成一组单词，例如两个句子：”She loves cats.“、”He loves cats too.“ 我们可以构建一个词频字典：{"She": 1, "He": 1, "loves": 2 "cats": 2, "too": 1}。根据这个字典，我们能将上述两句话重新表达为下述两个向量: [1, 0, 1, 1, 0]和[0, 1, 1, 1, 1]，每 1 维代表对应单词的频率。

近年来，借助深度学习概念和性能强劲的硬件平台，Geofrey Hinton, Tomas Mikolov, Richard Socher 等学者深入开展了针对词向量的研究，将自然语言处理推向了新的高度。以词向量为基础，可以方便引入机器学习模型对文本进行分类、情感分析、预测、自动翻译等。最简单的词向量就是独热编码(one-hot encoder)，比如有三个单词”man"、”husband“、”dog“，将之分别表示为[0,0,1]，[0,1,0]，[1,0,0]，这些词向量可以作为机器学习模型的输入数值向量，但是它们依然难以表达关联性，而且当词库单词量庞大时，独热编码的维度也会十分巨大，给计算和存储带来不少问题。Mikolov、Socher 等人提出了 Word2Vec、GloVec 等词向量模型，能够比较好的解决这个问题，即用维数较少的向量表达词以及词之间的关联性。关于这些词向量模型的具体原理，可以阅读他们所发表的论文，主要是英文，中文网站上也出现了不少精彩的翻译和解读，可以参考某些关于自然语言处理的中文博客。

本项目目的就是利用上述自然语言处理技术结合所学机器学习知识对文档进行准确分类。

输入数据

分类文本数据使用经典的 20 类新闻包，里面大约有 20000 条新闻，比较均衡地分成了 20 类，是比较常用的文本数据之一。每个类的数量如下图所示：



数据以 txt 文件保存，除了正文，还有 Header，Footer 等各种附加信息。正文的风格，字数都有很大差异。随机选取一节加以分析：

Path: cantaloupe.srv.cs.cmu.edu!magnesium.club.cc.cmu.edu!news.sei.cmu.edu!cis.ohio-state.edu!zaphod.mps.ohio-state.edu!cs.utexas.edu!uunet!news.tek.com!vice!bobbe
From: bobbe@vice.ICO.TEK.COM (Robert Beauchaine)
Newsgroups: alt.atheism
Subject: Re: Amusing atheists and agnostics
Message-ID: <11860@vice.ICO.TEK.COM>
Date: 20 Apr 93 15:37:10 GMT
References: <timmbake.735204406@mcl> <madhausC5rFqo.9qL@netcom.com>
Organization: Tektronix Inc., Beaverton, Or.
Lines: 18

In article <madhausC5rFqo.9qL@netcom.com> madhaus@netcom.com (Maddi Hausmann) writes:
>
>"Clam" Bake Timmons = Bill "Shit Stirrer Connor"
>

Sorry, gotta disagree with you on this one Maddi (not the
resemblence to Bill. The nickname).

I prefer "Half" Bake'd Timmons

/\

Bob Beauchaine bobbe@vice.ICO.TEK.COM

They said that Queens could stay, they blew the Bronx away,
and sank Manhattan out at sea.

^^

其文本内容与 E-mail 类似。有一系列头标签，比如 Path,From,Newsgroups,Subject,Message-ID,Date,References,Organization,还有行数 Lines. 文本内容短的只有 10 行左右，而长的则有上千行。

解决方法

选择使用词袋子模型来表示每篇文档，思路是首先将文本进行分词，也就是将一个文本文件分成单词的集合，建立词典，每篇文档表示成特征词的频率向量或者加权词频 TF-IDF 向量，这样可以得到熟悉的特征表。接下来，就可以方便利用机器学习分类模型进行训练。

利用 Word2Vec[2] 方式即词向量模型表示每篇文档，利用文本数据对词向量进行训练，将每个单词表示成向量形式。词向量训练后需要进行简单评测，比如检验一些单词之间相似性是否符合逻辑。

分别在词袋子、词向量表达基础上采用你认为适当的模型对文本分类，优化模型并分析其稳健性。

基准模型

将数据分别使用决策树模型、支持矢量机(SVM)[3] 模型、以及深度学习模型。通过对比选择最优模型作为最终的模型选择。无论是 TF-IDF 还是 Word2Vec，最终都会将数据转化成向量的类型，而向量基本适用于任何模型。所以可以依次初略运算，然后择优者深度优化参数，进而更一步得到理想的结果。

评估指标

本项目使用准确率作为评估指标。虽然二十分类这种多分类的问题可以用混淆矩阵来做细化的评估指标，但是我个人还是喜欢以一个分数来评价整个模型。所以可以将多分类，通过是非判断，简化为二分类的问题。从而建立二分类的评估标准和指标。公式如下：

准确率$= \frac{\sum_{i=1}^{n} A(y_i=Y_i)}{n}$

其中 y_i 表示预测分类，Y_i 表示实际分类。而 A 函数表示如果两者相等，A=1，否则 A=0 。

.

设计大纲

毕业论文将从五个段落分辨阐述整个项目的过程，分别是定义、分析、方法、结果以及结论。

首先，问题的定义非常清晰，就是对文本数据的分类。

而分析的过程，就是对数据本身，包括格式，内容，特征等各种因素进行研究。同时也需要分析算法模型如何与实际数据结合，进而得出更好的结果。

文本处理的预处理同样包含很多内容。对于文本的标签行我觉得可以全部忽略。并且对全部的标点符号，特殊符号都需要去除。停止符和换行符同样全部删掉。对于异常值需要初步分析后才能

确定，预计是去除一些全部文本中重复次数小于 3 次的，作为错别字处理。重复很多次的相同错别字，可以作为相同意思的不同词处理。

进而做最后选择的方法的描述。对于 TF-IDF 实际是以某词在某文出现次数占全文字数的百分比，以及含某词的文数占总文数的百分比的乘积来表示。而 Word2Vec 实际是对于相似词的聚合。我计划通过这两种模型的结合，得出某相似意义的词在某文全文字数百分比，和含某相似意义的词占总文书百分比的乘积来进行优化算法。思路是通过 Word2Vec 找到同义词，然后在同义词中找到最多出现次数的词，然后用最多出现次数的词替换其余的相似词，再进行 TF-EDF 算法分析。希望可以有更好的成绩。

而具体的分类算法我直接是选择 SVM 会有比较高的分数，具体可以在项目正文中进一步研究与确定。进而得到最终的结果。

因为该项目在网上并没有找到太多可以参考的分数，所以我的目标是让自己的正确率达到 85%。

最后的结论总结再进一步描述整个项目过程中的一些难点和体会。

谢谢。

参考

【1】 NLP：https://baike.baidu.com/item/nlp/25220

【2】 word2vec：https://baike.baidu.com/item/Word2vec/22660840?fr=aladdin

【3】 SVM： https://baike.baidu.com/item/svm