

硕士专业学位论文

古典诗词意境的自动识别

Automatic Recognition of Ancient Poetry Artistic Conception

作者：蒋俊成

导师：李宇

北京交通大学

2020年4月

学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。学校可以为存在馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名： 导师签名：

签字日期： 年 月 日 签字日期： 年 月 日

学校代码：10004

密级：

北京交通大学

**硕士专业学位论文**

古典诗词意境的自动识别

Automatic Recognition of Ancient Poetry Artistic Conception

作者姓名：蒋俊成 学 号：17126165

导师姓名：李宇 职 称：副教授

工程硕士专业领域：软件工程 学位级别：硕士

北京交通大学

2020年4月

致谢

首先要感谢我的李宇导师，本文是在李宇老师的悉心指导和尽力帮助下完成的。在做毕业设计期间，李宇老师十分关心我的进度，给了提供了很多的帮助和指导。每当我在研究中遇到困难时，李宇老师总是能提出十分有益的建议和见解。李老师广博的知识，富有前瞻的视野和一丝不苟的治学态度让我仰望敬佩。除此之外，在我找实习期间和找工作期间，李宇老师还提供了许多有价值的信息和参考意见，帮助我找到了高质量的实习和令人满意的工作。在此学业期满之际，我衷心感谢李宇老师在我研究生期间的无私付出。

感谢Florida Atlantic University的Shihong Huang教授，在我刚到美国的时候，是黄教授尽心地帮助我适应了当地环境，安排我尽快进入了学习状态。六个月里她带着我做了十分有意义的学习和研究。黄教授的热情和友善，在学术上的认真严谨和循循善诱给了很大的帮助，让我的美国之行愉快而充实。

感谢北京交通大学的大力栽培，感谢学校给我提供的出国交流访学的机会并资助我完成访学，这对我来说是一生难忘的回忆。从本科到硕士这么多年一直在交大生活、学习，交大浓厚的学术氛围，严谨的治学风气深深地影响了我的品性，临别之际这里的一花一木都让人难以割舍。

最后感谢我的父母，他们以无私的爱给了我最坚强的后盾，以自己的一言一行教导我、支持我，让我在耳濡目染中获得进取的精神与独立的人格。家人的关心与鼓励是我不断前进的动力。

摘要

诗词是我国古典文化皇冠上的明珠，千百年来被无数人传颂和研究。但受限于技术手段的贫乏，对古典诗词的研究一直以来都是靠文人学者们的人力劳作。近年来随着计算机技术的迅猛发展和硬件算力的飞跃式提升，自然语言处理技术在实践运作中取得了令人瞩目的成果。本文将自然语言处理运用于古典诗词的文本分析当中，研究各种算法和工具并整合出一套自动识别古典诗词意境的方法，对诗词的批量化研究和诗词大众推广提供一定的帮助。

首先利用爬虫从互联网上收集大量的分类标注和未标注的古诗词文本语料，以及大量的非诗词古汉语语料。对语料进行数据清洗，入库。为了尽可能贴近原汁原味的古文，研究中一律使用繁体中文作为标准。使用Word2Vec工具进行词向量训练，完成语料的准备工作。

研究不同的机器学习算法在文本分类上的应用，将其和不同的文档向量化方法相结合，比较其在诗词主题分类问题上的表现，总结出在经典机器学习框架下有较好准确率的方法。比较基于字的向量和基于词的向量的效果，得出基于字构建的古诗词向量具有较高准确率的结论。

进一步研究将深度学习方法引入到研究中，分析各类神经网络的优劣，研究学界在自然语言处理实践中具有较好效果的神经网络架构，如Text CNN和Bi LSTM等。引入成熟的自然语言处理预训练模型如BERT进行主题分类和情感分析。

本文对古汉语和现代汉语在自然语言处理中的不同表现做了一定的研究，分析了古诗词在分词、向量构造和文本分类问题中的难点，并提出了部分解决方案。

将自动识别算法应用于大量的未标注诗词，在得到的数据基础上开发出诗词意境分析与相似诗词推荐系统并部署到微信小程序上。

**关键词：**古典诗词，意境，自然语言处理，文本分类。

ABSTRACT

一般为1000个左右实词。

**KEYWORDS：**Ancient Poetry; Artistic Conception; Nature Language Processing; Text Classification.

目录

[摘要 iii](#_Toc35973407)

[ABSTRACT iv](#_Toc35973408)

[1 引言 1](#_Toc35973409)

[1.1 4646 1](#_Toc35973410)

[标题 2](#_Toc35973411)

[标题 3](#_Toc35973412)

[标题 3](#_Toc35973413)

[2 基于词向量的古文本分析 5](#_Toc35973414)

[2.1 Word2Vec简介 5](#_Toc35973415)

[2.1.1 从词嵌入到词向量 5](#_Toc35973416)

[2.1.2 CBOW模型和Skip-gram模型 7](#_Toc35973417)

[2.1.3 句向量和文档向量 8](#_Toc35973418)

[2.2 古文词向量与现代文词向量的差异比较 8](#_Toc35973419)

[2.3 字本位与词本位 8](#_Toc35973420)

[2.4 汉字计算机编码问题研究 9](#_Toc35973421)

[2.4.1 字符，字符集与字符编码 9](#_Toc35973422)

[2.4.2 各字符集及其对繁体中文的支持 10](#_Toc35973423)

[2.5 数据获取 10](#_Toc35973424)

[2.5.1 全唐诗，全宋词和其他古典诗词 10](#_Toc35973425)

[2.5.2 四库全书 11](#_Toc35973426)

[2.5.3 使用爬虫获取文本数据 11](#_Toc35973427)

[2.5.4 数据清洗 13](#_Toc35973428)

[2.5.5 使用MySQL建立语料存储数据库 15](#_Toc35973429)

[2.6 向量训练 16](#_Toc35973430)

[2.6.1 古汉语分词规则研究 16](#_Toc35973431)

[2.6.2 使用自然语言处理工具Gensim进行向量训练 17](#_Toc35973432)

[2.7 本章小结 18](#_Toc35973433)

[3 文本分类算法的使用和结果分析 19](#_Toc35973434)

[3.1 文档向量化方法研究 19](#_Toc35973435)

[3.2 各机器学习分类算法及其效果评估 19](#_Toc35973436)

[3.2.1 朴素贝叶斯分类及其效果评估 20](#_Toc35973437)

[3.2.2 k最近邻算法分类及其效果评估 21](#_Toc35973438)

[3.2.3 支持向量机分类及其效果评估 23](#_Toc35973439)

[3.3 使用字向量集合构建主题域进行文本分类的研究 24](#_Toc35973440)

[3.3.1 主题域的构想 24](#_Toc35973441)

[3.3.2 近似关键字搜索算法 24](#_Toc35973442)

[3.3.3 字向量加权相似计算主题相似度 24](#_Toc35973443)

[3.3.4 结果评估 25](#_Toc35973444)

[3.4 关于各算法优劣的比较 25](#_Toc35973445)

[4 搭建神经网络实现文本分类 26](#_Toc35973446)

[4.1 标题 26](#_Toc35973447)

[4.1.1 标题 26](#_Toc35973448)

[结论 27](#_Toc35973449)

[参考文献 28](#_Toc35973450)

[附录 A 29](#_Toc35973451)

[索引 30](#_Toc35973452)

[作者简历及攻读硕士/博士学位期间取得的研究成果 31](#_Toc35973453)

[独创性声明 32](#_Toc35973454)

[学位论文数据集 33](#_Toc35973455)

# 

# 引言

**古典诗词是中国传统文学王冠上的明珠，因其亘古不衰的艺术魅力代代相传，千百年来在无数人心中唤起跨越时空的共鸣。即使到了现代，仍然是最广为流传的文学形式，被无数人传颂吟诵。它形象生动的字词，富有韵律的节奏，饱含深情的意境，隽永悠长的情感，深受从文人到布衣的喜爱，引得无数学者对它深入研究。**

**然而，中国古典诗词承载着千年的传承，作品数量浩如烟海，对于普通研究者来说其工作量十分繁重，靠个人的学识与记忆往往难顾周全，对于诗词的意境思想的理解也是千人千面。而传统的诗词计算机辅助研究主要工作于诗词的收集整理分门别类，以及数据库的构建和知识图谱的生成等，对于深入到语义理解层次的研究则相当空白。随着计算机科学的不断发展，自然语言处理作为一门新兴的学科在各个语言文本领域取得了相当大的成功，因此笔者将自然语言处理方法应用于古典诗词的研究中，使用当前主流的机器学习和深度学习方法，对古典诗词的主题和情感，进行分类分析，以期望达到对数以万计的诗词做出有效的分类和标注。这对我们弘扬传统文化，发扬文化自信具有一定的积极意义。**

## 课题背景及研究意义

最早的自动文本分类可追溯到上世纪60年代，Maron等人提出基于概率索引的分类模型，并使用贝叶斯公式实现了一定程度的自动分类。随后各种文本分类的研究相继面世。文本分类问题大致可分为文本表示，分类模型训练，分类效果评估三个步骤。文本的表示有词袋模型（Bag of Words）、n - gram模型，共现矩阵 (Cocurrence matrix)、One - Hot编码等，在基于词嵌入方法上还有词向量等各种分布式表示。在经典机器学习领域分类模型常见的有朴素贝叶斯，支持向量机，k最近邻，回归模型，主题模型，决策树与随机森林等。在分类效果评估上采用准确率（Precision）、召回率（Recall）、F-1评分等指标。

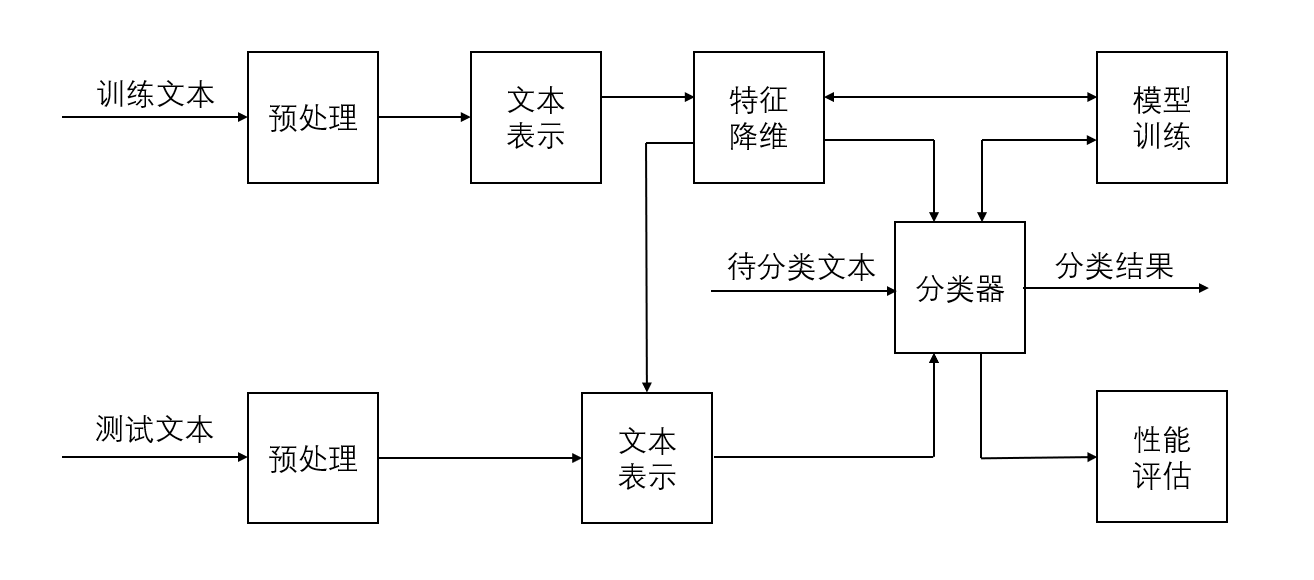
经典文本分类策略的流程如下：

图1-1 经典文本分类模型流程图

除了传统的机器学习算法外，基于神经网络的深度学习分类算法也在蓬勃发展。Bengio等人在2003年提出NNML[1]是最早的自然语言处理神经网络模型，近几年随着计算机算力的飞速发展，各种NLP神经网络模型如雨后春笋般出现，语料预处理有Word2Vec，ELMo等模型；文本分类有LDA主题模型，Fast Text，以及CNN和RNN的各种衍生模型；预训练模型有GPT，BERT等。文本分类技术的日渐成熟使得本研究成为了可能。

在庆祝中国共产党成立95周年大会上习近平总书记提出“四个自信”，在原有的“三个自信”基础上增加一条“文化自信”。并指文化是一个国家、一个民族的灵魂，文化自信是一个国家、一个民族发展中更基本、更深沉、更持久的力量[2]。古典诗词作为中国传统文化的瑰宝，深深地铭刻在我们的民族记忆之中，流淌在我们文化的血脉里，是中华民族精神内核中不可或缺的部分。利用自然语言处理技术对古典诗词的意境进行分析研究，对于在当代弘扬“文化自信”有着积极的意义。

## 国内外研究现状

在古典诗词的自然语言处理领域，国内早期研究中比较有代表性的是北京大学计算语言研究所与台湾元智大学古文献研究所合作研究的“古代诗词研究的计算机支持环境”[1]，此项研究建立了由古典诗词构成的语料库，建立了全唐诗检索系统，在对诗词文本的注音，词汇语义的辨析，作者信息的分辨上做了一定的研究。厦门大学的周乐昌教授在他的“心脑计算举要”[5]中首次提出“计算诗学”的概念，将人工智能与古典诗词分词相结合，并对未来研究做了一定的展望。在胡俊峰的博士论文“基于词汇语义分析的唐宋诗计算机辅助深层研究”[6]中，把一些当前自然语言处理和语言学计算技术按照古典诗词的特点付诸应用，获得了一定的效果。此项研究有体系地提取了汉语古典诗词语料组成知识库，为后继者的研究带来了一定的启发。在重庆大学的易勇的博士论文“计算机辅助诗词创作中的风格辨析及联语应对研究”[7]中，提出了使用机器学习算法对唐诗宋词的风格按豪放和婉约进行分类，并使用层次聚类算法和自组织特征映射网络对分类器进行优化，达到了可观的效果；同时使用了隐形马尔科夫模型序列学习法和基于转换的错误驱动序列学习法，实现了对联中按上联自动生成下联的功能。梁健楠等在“基于神经网络的集句诗自动生成”[8]中，提出了利用循环神经网络(RNN)自动生成集句诗的模型，所谓集句诗是指从前人的诗词作品中挑选诗句，将其组合成新诗的艺术再创造形式。它不仅要求合辙押韵，而且要求内容完整、上下文连贯，对文本生成模型的语义表示和理解能力有很高的要求。申资卓等在“基于主题模型的古典乐器诗词文本挖掘”[9]中聚焦于古典诗词中乐器的研究，按古人把乐器依照制作材料分为八种，即“八音”的规则，深度挖掘了《全唐诗》《全宋词》中“八音”有关的诗句，研究了LDA主题模型的挖掘，作者相似度等的计算，深入地研究了唐诗宋词中的古典乐器。吴斌等在“基于迁移学习的唐诗宋词情感分析”[10]中，对诗词按照“正面，中性，负面”三大类进行情感分类，使用迁移学习建立了三个分类器，以投票的方式得出情感分析结果，使用了CATL- PCO模型解决古文因文本短而造成的特征矩阵稀疏的问题，同时解决了缺乏现代文译文而导致的古典诗词情感难以分析的问题。李奇在“古典诗词意境分类方法的研究与实现”[11]中，提出了将古典诗词分为“军旅边塞，咏史怀古，山水田园”三类。使用特征项聚类的方法将这三类诗词的常见字列举出来，在机器学习中加大其权重，再用常见的机器学习算法进行分类，取得了一定的效果。李炎良等在“基于词联接的诗词风格评价技术”[12]中，提出基于词联接技术对诗词风格按照“豪放”和“婉约”进行分类，将所有词汇建立两两关系，每个词与上下文构成三元组判断修饰词与中心词的正负趋向关系，在此基础上构建句子最优树搜索判定风格的正负。易勇等在“基于机器学习的古典诗词作者的判别研究”[13]中，基于机器学习中的朴素贝叶斯算法，对唐诗中李白和杜甫的作品依靠文本判定作者，进行二分类研究，使用爬山算法和最佳信息增益法对分类器进行改进，将准确率提高到了可观的程度。

近年来，随着深度学习的迅猛发展，学术界也开始尝试将各种神经网络应用于文本分类的研究当中。Guoyin Wang[14]等人提出将文本分类看作一个标签-单词的联合嵌入问题:每个标签与单词向量嵌入在同一个空间，构建了一个attention框架，它度量文本序列和标签嵌入之间的兼容性。用词嵌入矩阵表示文本序列，通过神经网络等方法将词嵌入矩阵变为定长的向量表示，将定长向量z传入分类器，获得分类概率分布向量。（）

## 论文主要工作

## 论文篇章结构

本文由以下几个部分构成：

第一章为引言部分，主要介绍课题背景和研究意义，当前国内外的研究现状，本文的工作内容和篇章结构。

第二章

第三章主要讲述数据准备的过程。本章确定了研究所需要的语料，介绍了常用爬虫工具和使用爬虫获取语料的过程，介绍了数据库的搭建，数据的清洗和入库。本章详细说明了词向量原理，描述了将已有语料构建成词向量的方法。

第四章主要讲述了使用经典机器学习方法对古典诗词进行文本分类的研究。本章提出了几种不同的文档向量化方案，使用朴素贝叶斯，k最近邻，支持向量机三种机器学习分类器进行文本主题分类实验，比较其准确率。本章研究了各种方案的优劣，提出了相应的改进策略，为后续进一步研究提供了有效信息。

第五章

第六章是全文总结，归纳了研究的成果和不足，并提出了仍需进一步完善的内容和研究方向。

# 古典诗词的意境浅析

## 标题

### 标题

# 基于词向量的古文本分析

通过词嵌入（Word Embedding）将自然语料转化成词向量（Word2Vec），是当前自然语言处理领域的热门方法。相比于使用原始语料的传统学习方法，使用词向量作为机器学习以及神经网络的初始输入和训练材料，具有更准确，更强大，更具有拓展性和可解释性的诸多优点，同时也能减少特征提取的难度，避免特征工程中的诸多困难。因此本文主要采用词向量用于语料的预处理。

## Word2Vec简介

2001年， Bengio 等人正式提出神经网络语言模型（ Neural Network Language Model ，NNLM），该模型在学习语言模型的同时，作为副产品得到了词向量。本节将对词向量作一个梳理介绍。当前主流的词嵌入实现是Word2Vec和Glove，本文使用Word2Vec作为具体实施方法，Glove方法在此不再赘述。

### 从词嵌入到词向量

在自然语言处理中对词语表示的最常见方法是独热码表示（One-Hot Representation），也就是将每个不同的单词表示成一个某一位置是1，其他位置全是0的一维向量，向量长度即等于词典中单词的总数。

（图）

独热码有两个显著的缺陷：首先是独热码的向量长度随着词典长度增加而增加，导致维度灾难（Curse of Dimensionality）；更重要的是独热码无法表示词与词之间的相似关系，任意两个词都是孤立的，这对文本的深入分析造成了阻碍。为了解决这个问题，词的分布式表示方法（Dristributed Representation）被引入到NLP的研究中。它的基本思路是通过神经网络的训练，把每个词映射成一个维度较短的向量，由词典中所有词向量所组成的空间即向量空间，向量的维度由实际情况自定义，一般在100维到300维之间。

（图）

词嵌入（Word Embedding）即是这种映射的训练方法，它的核心思想是分布假说（Distributional Hypothesis）：词的语义由其上下文决定。通过神经网络输入一个句子的One-Hot编码，预测每个词依赖于上下文表现的分布概率，不断迭代训练调整网络参数，使得预测概率的准确率达到最大。

自然语言预计模型NNLM(Neural Network Language Model)的网络结构

词向量最大的特点就在于它能表示单词之间的关系，越是相似的词其向量夹角越小。通过计算向量的余弦相似度，即能得到两个词的相似度。

词向量的另一大优势在于它能进行单词类比，两个向量相减的向量距离在向量空间中具有实际意义，相似向量可以通过向量加减来进行类比。

2013年，Google团队正式发布了他们的词嵌入工具并将其命名为Word2Vec，该工具封装了两个训练方法： CBOW，即连续词袋模型（Continuous bag of words）；和Skip-gram，即跳字模型。

### CBOW模型和Skip-gram模型

词嵌入的基本思想是词的语义由上下文决定。在实践中可以用NNLM式的多层前向反馈神经网络，通过上下文来拟合在一个词序列中某个特定的词的条件概率，即

p(wt|wt-n,...,wt−1,Wt+1,…Wt+n) = f(wt-n,...,wt-1,Wt+1,…Wt+n)

在这个模型中对输入和输出的定义方式分为两种，CBOW和Skip-gram。其网络结构见图3.x

CBOW模型（左）和Skip-gram模型（右）

CBOW模型源自于词袋模型（Bag of Words），该模型的训练输入是某一个选定词的上下文对应的词向量，输出该词所在位置所有可能的词向量的softmax概率，训练的目标是使得训练样本中原本的词所对应的softmax概率达到最大。一般人文上下文每个词都是平等的，也就是不考虑每个词和目标词之间的距离。

Skip-gram模型与CBOW模型正好相反，输入某个选定词的词向量输出该词所在位置上下文所有可能的词向量的softmax概率，训练网络使得上下文中原本词汇对应的概率达到最大。值得一提的是Skip-gram模型考虑上下文每个词的位置和加权距离。

### 句向量和文档向量

作为词向量的一种扩展，句向量（Sentence Vector）被提出来研究分布式表示方法的更多应用。其典型方法有Ryan Kiros等人在2015年提出的Skip-Thought Vectors。利用前后语句之间的关联性来学句子的Embedding. 其句子的预测是用RNN来完成，该模型先编码某个选定的句子，然后用这个RNN的输出依次来解码预测前一个和后一个句子，是Word2Vec的Skip-gram模式的直接扩展应用。文档向量（Document Vector）同样是词向量的一种扩展，其中比较常用于实际工程中的是Quoc Le 和Tomas Mikolov给出的Doc2Vec的方法。

笔者认为直接使用文档向量用作文本分类的依据将会是一种可行的方案，然而当前主流研究中文档向量基于TF-IDF算法实现，其结果不能很好地适用于关键词难以提取的古汉语文本。笔者认为有必要自行设计一种适用于诗词的，从词向量出发映射到文档向量计算方法，该部分见下一章节。

## 古文词向量与现代文词向量的差异比较

作为一种流行的NLP方法，词向量广泛地应用于研究中。海量的开源项目提供了数量众多的已经训练好的词向量模型。然而笔者却发现，直接使用这些现成的词向量用于古典诗词文本分析的效果差强人意。笔者认为，当前研究中的词向量都是基于大量的现代汉语文本进行训练所得，而现代汉语与古汉语有着显著的差异，这些差异也体现在了词向量上。

古汉语在语音规律，构词规律，词汇，词性活用，语法，句法等方面，都与现代汉语存在巨大的差异。其中比较重要的一点是，现代汉语是词本位的，词语是文本的基本构成元素，在文本中单个的字如果不是单字成词，则往往是没有意义的。而古汉语无法严格区分字本位与词本位，一般认为古汉语一字一义，同时又包含大量的特定词汇与组合词汇，语义不能由其组成的单字语义简单相加而成。

因此笔者认为有必要抛弃已有的现代汉语词向量，完全以古典文本为语料，训练专用的古汉语词向量用于后续研究。

## 字本位与词本位

中文NLP研究中的另一大问题在于分词，不同于英文不存在词语切分的问题，中文语料分析必须先先进行分词，当涉及到古汉语时，情况将变得更加复杂。这是因为对于现代汉语而言，只需按词语进行划分即可；但对于古汉语，还需要考虑采用字本位还是词本位的问题。

在语言学中，义原（Sememe）指的是最小的不可再分的语义单位，在现代汉语中我们可以粗暴地认为词语就是义原，虽然不严谨，但在大量的实践中证明了这种划分切实可行。而古汉语的义原究竟是字还是词是一个模糊的问题，在NLP研究实践中也没有先例可循。笔者决定以字本位和词本位为基础，各自独立地训练出两套互不相关的词向量模型，在后续研究中同时使用，并比较其在不同算法中的表现。

## 汉字计算机编码问题研究

当前使用的简化汉字是20世纪新中国为方便群众使用，快速降低文盲率而推广的文字，在简化的过程中省略了许多古文中的细节，在某些情况下把多个繁体字合并成同一个简体字使用。比如——

“發”“髮”都简化为了“发”；

“複”“復”都简化为了“复”；

“乾”“幹”“干”都简化为了“干”；

“蘇”“蘓”“穌”“甦”“囌”“苏”都简化为了“苏”，

因此使用简体字并不能完整地反映古文的原始风貌。学界在对于古汉语进行研究时，一律都是使用繁体字，即传统中文进行的。

对于普通的汉字相关问题研究来说，只要统一使用Unicode（或GBK等其他字符集），以UTF-8进行编码存储即可，不存在编码问题。但对于计算机中的繁体中文来说，这是一个必须详细研究的问题。

### 字符，字符集与字符编码

字符（Character），指人类各种语言文字和符号的集合，包括各个国家和民族的文字，标点符号，图形符号，以及数字等。

字符集（Character set），指将字符映射成二进制的一种系统，也就是给每个字符分配一个唯一ID。比如拉丁语系字符集ASCII。需要我们研究的字符集有Unicode、GB2313、GBK、GB18030、BIG5。

字符编码（Character encoding），指将字符集转化成特定的二进制字节序列，并在计算机中存储的规则。各个字符集都有自己的编码模式，但Unicode除外。常见的编码方式还有UTF-8以及在此基础上扩展的UTF-16、UTF-32。通常Unicode使用UTF-8进行编码存储和传输。

### 各字符集及其对繁体中文的支持

ASCII是基于拉丁字母的最早的通行字符集。由于英语等拉丁语系字符表达的便利性，这套编码规则只需要定义128个字符即可满足绝大多数情况下的需求。

GB2313是最早的中文字符集，兼容ASCII，由中国国家标准总局在1981 年 5 月 1 日发布，是中国国家标准简体中文字符集。在此字符集中收录了简体汉字6763个，其他字符682个。

Big5是由台湾地区在1983年五大软件所设计的繁体中文字符集，因此得名“大五码”。共收录13060个繁体字。由于Big5是针对当代港台地区所使用的现代繁体中文所设计，并不能很好地编码古文里的异体字和罕见字。

GBK是对GB2313的升级，加入了对繁体字的支持，其中的K即“扩展”。由全国信息技术标准化技术委员会在1995年12月15日发布，共收录汉字21003个。该标准因广泛地应用于微软的各个Windows版本中而广为人知。对罕见字的收录仍不尽如人意。

GB 18030由信息产业部在2000年 3月17日发布，共收录汉字70244个。该标准向下兼容GB 2312和GBK的全部内容，新增了大量此前不被收录的异体字和罕见字，除此之外还收录了彝、藏、蒙、朝、维等少数民族文字，以及其他几种字符集一直没有收录的特殊字符。

Unicode又称万国码，是为了解决各个国家不同文字使用不同编码标准这个问题而产生的统一标准，于1994年正式实行。Unicode经过多版本的迭代，如今已包含71226个汉字。

## 数据获取

基于上文叙述的相关技术，为满足古文词向量训练的要求，需要搜集到足够的古典文本作为语料。需要的数据主要分为三类：1，已经标注好的诗词语料；2，未经标注的诗词语料；3，其他尽可能多的古文语料。

### 全唐诗，全宋词和其他古典诗词

在我国浩如烟海的古典诗词中，唐诗宋词是最具有代表性，最为人所熟知的，它们是中国古代文学发展史上的两座高峰，为人世世代代口耳传诵，流传千年至今。其文学价值，人格境界，反应的社会风貌，记载的历史进程在当今具有无可比拟的价值。

《全唐诗》是清康熙四十四年（1705年），彭定求、沈三曾、杨中讷、汪士鋐、汪绎、俞梅、徐树本、车鼎晋、潘从律、查嗣瑮等十人奉敕编校，“得诗四万八千九百余首，凡二千二百余人”，共计900卷，目录13卷。《全宋词》是当代新编的宋词总集编，由中华书局出资编纂。

除我们常常吟咏的唐诗宋词外，古典诗词的类型还有诗经，楚辞，汉魏晋诗歌，南北朝诗，唐词宋诗，元曲，明清诗词，凡此种种，不一而足。每一种都是中华文化的瑰宝，也是我们词向量训练的重要材料。得益于当代互联网分享精神的广泛流传，我们可以不必再从繁复的卷帙中寻章摘句，使用爬虫技术即可获取全部需要的数据。

### 四库全书

经笔者统计，使用爬虫从互联网上爬取到接近全部的可用诗词语料后，其全部数据量为诗词累计约35万首，字数约1360万字。作为用于词向量训练的语料来说，其数据量略显不足。为解决这个问题，笔者考虑将其他非诗词的古文本也加入到词向量的训练中来。

《四库全书》是清乾隆年间，由清廷主持编纂的一部包罗万象的中国古典书目大百科全书。是现今中国规模最大的一套书目集合。其编纂历时9年，共收书3503种，79337卷，36304册，近230万页，约8亿字。清廷为之特设“四库全书馆”负责编纂总务，参与人员包括内阁大学士，六部尚书，侍郎，各界学者等，参与编纂者最多时达7000余人。

所谓“四库”，指的是经、史、子、集4个大的类目，其下收录的书目有四书五经，二十四史，兵法政论，农林工商，医学术数，佛道释杂，地方县志，诗集文选，演义小说，不一而足。当代能寻到的所有古文本，几乎都被收录在四库全书之中。因此它可以在我们的研究中，近似地等于所有古文本语料。

### 使用爬虫获取文本数据

对于古汉语文本，当前并没有一个通用的开源语料库可供使用，因此在选择好训练所需的语料后，我们需要使用爬虫从互联网上爬取语料数据。常见的爬虫框架有Scrapy，PySpider，Beautiful Soup，Crawley，Portia，Nutch等。

Scrapy是一个高层次的、快速的网页抓取框架，它能自定义请求并从响应中提取数据，做结构化解析后输出。内置了cookies and session 处理，HTTP 压缩，HTTP 认证，HTTP 缓存，user-agent模拟等功能，对于动态网页爬取有较强的适应性。它也可以通过访问API来提取数据。

PySpider是一个中国开发者发布的爬虫框架，自带强大的UI界面。它使用分布式架构，与多种数据库后端可进行直接通信。它包含项目管理器，任务监视器，脚本编辑器以及结果查看器。

Beautiful Soup是一个网页编码解析器，它可以从XML和HTML中提取数据，做结构化解析输出，方便数据和文档的辨认，转化和提取。Beautiful Soup需要和其他的HTTP通信工具一并使用。

由于需要的数据都来自静态网页，不需要做用户认证，不涉及监听、抓包、JavaScript文件分析等问题，因此不必使用动态网页抓取的相关工具；另一方面笔者观察发现目标网页结构复杂，冗余信息繁多，数据提取和数据清洗的需求大，因此需要对HTML解析能力强大的工具帮助。综合以上两点需求，笔者决定使用Beautiful Soup作为爬取工具。

Beautiful Soup自身并不包含Web通信的部分，因此需要一个单独的工具用来发送HTTP请求。一般的小项目中通常和Requests配合使用，但这样构建出来的爬虫是同步爬虫，也就是每个请求都必须等到得到响应后，才能发送下一个请求。在大量数据需求的场景下，其效率十分低下。为了提高数据获取的效率，可以引入异步IO模块。Asyncio是Python内置的单线程并发IO操作模块，通过消息循环实现了对异步IO的支持，并实现了SSL、TCP、UDP等通信协议。Aiohttp则是基于Asyncio实现的HTTP框架。使用Asyncio + Beautiful Soup便获得了我们需要的异步IO请求爬虫框架。每类语料都要对目标网页进行预分析，这里以四库全书为例说明。

选定目标站点“国学大师”http://www.guoxuedashi.com/SiKuQuanShu/作为爬取目标。在四库全书的栏目中，首先是1级目录，按“经史子集”四个大类依次罗列了所有的书目名称和链接。打开书目链接到达2级目录，这个目录里罗列的是该书的各个章节，打开章节链接到达正文页面。依据此结构设置爬虫规则。

首先爬取1级目录，查看网页HTML源码可知，需要的信息存放在<div class="info content zj clear fix">标签下的表格中，有四张大表标签分别是<table class="table1">到<table class="table4">，对应“经史子集”。每个书目存于href链接中，比如<td><a href="/a/54h/" target="\_blank">《资治通鉴》</a>。依次上述顺序获取信息，将书名和对应的链接存入“四库全书\_一级目录.txt”中。

接着爬取2级目录，读取一级目录txt中的每条记录，请求对应页面，解析响应的HTML源码。目录存放在<div class="info cate clear fix">中的href标签中，依次读取，存入“书名\_目录.txt”。

最后遍历每个2级目录，发起请求抓取正文。正文存放在<div class="info txt clear fix" id="info zj txt">标签下，读取正文后调用负责数据清洗的函数，存入数据库。其他语料的抓取不再累述。

### 数据清洗

各个网站上的诗词和四库全书数据中包含了大量的冗余信息，不属于有效语料数据，在存入数据库之前需要进行数据清洗。

经统计得到的文本中包含了不同形式的括号，其中包含了各种内容，有的是文本段落的编号，有的是编者加的注释，还有人名地名典故的解释等等，但都不是我们需要的古文本信息，需要连带括号一并去除。文本中包含的括号类型有：[]、{}、()、〔〕、【】等。通过正则表达式匹配括号，连带括号内的内容一并替换成空字符串。表达式如下：

t = re.sub(u'\\(.\*?\\)|\\〔.\*?\\〕|\\（.\*?\\）|\\【.\*?\\】|\\[.\*?\\]|\\{.\*?\\}|\\｛.\*?\\｝','',text)

据笔者观察，由于各个语料来源不一，文献的汇编者水平参差不齐，网页排版方式各不相同，标记、引用、编号、注释的书写方式也是千差万别，更有甚者在文档中加入各种稀奇古怪的字符，给笔者的数据归一化造成了相当大的困难。

如前文所述语料主要分为三大部分：标注诗词，未标注诗词，四库全书。经仔细观察，标注诗词来自于同一个网站，其排版书写规则相对规范，没有非法字符。可以优先处理。处理步骤如下：

在爬虫程序的HTTP json字符串中，诗词名和作者如《静夜思·（唐）李白》，去除书名号和括号内容，分割后各存入Title和Author变量中；包含词牌名的如《念奴娇·赤壁怀古——（宋）苏轼》，将词牌名和题目合并，中间加空格区分，其他的和上面一致。某些文本中包含“编者按：”的前置介绍语句，通过字符匹配“编者按：”这四个字符将这一段话一并删除。有括号的删除括号及内部文本。有些诗词包含字词解释、全篇翻译或赏析，通过匹配“注：”、“注释：”、“译文：”、“赏析：”等字符串删除。到此为止清洗完毕。

对于未标注诗词和四库全书情况则要复杂得多。考虑这部分语料内容主要是来做词向量训练，笔者查阅了互联网上许多词向量训练的项目，发现在大规模和超大规模语料的处理中，特别是当语料来自各种不同信息源时，是无法做到百分之百准确清洗的。但在词向量训练的任务中，这个问题的影响非常微小，这是因为词向量是基于分布假说的统计模型，只要非法字符占比足够低，就能忽略它们在统计模型中的作用。因此笔者认为不需要做到如同上文中标注诗词那样，完全没有一点杂质的清洗程度。

对于2，3类语料首先按1类语料的方法进行处理。然后设计更多的字符串匹配规则去除包含现代文的部分，在此不再累述。随后笔者再次仔细观察了这些语料，认为里面包含极少量的现代文信息是可以接受的。剩下要做的就是去除非法字符。上文详细叙述的汉字编码在这里将得到应用。考虑到各个文档书写规则差异，笔者决定采用的方法是：将Unicode中文编码的范围和常见标点符号编码视为“合法范围”，检查每个字符是否在这个合法范围内，只保留合法的字符。

全体汉字 Unicode 编码范围

|  |  |  |
| --- | --- | --- |
| 字符集 | 字数 | Unicode 编码 |
| [基本汉字](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=jbhz) | 20902字 | 4E00 - 9FA5 |
| [基本汉字补充](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=jbhzbc) | 74字 | 9FA6 - 9FEF |
| [扩展A](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=kza) | 6582字 | 3400 - 4DB5 |
| [扩展B](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=kzb) | 42711字 | 20000 - 2A6D6 |
| [扩展C](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=kzc) | 4149字 | 2A700 - 2B734 |
| [扩展D](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=kzd) | 222字 | 2B740 - 2B81D |
| [扩展E](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=kze) | 5762字 | 2B820 - 2CEA1 |
| [扩展F](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=kzf) | 7473字 | 2CEB0 - 2EBE0 |
| 康熙部首 | 214字 | 2F00 - 2FD5 |
| [部首扩展](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=bskz) | 115字 | 2E80 - 2EF3 |
| [兼容汉字](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=jrhz) | 477字 | F900 - FAD9 |
| [兼容扩展](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=jrkz) | 542字 | 2F800 - 2FA1D |
| [P U A部件](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=puabj) | 81字 | E815 - E86F |
| [部件扩展](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=bjkz) | 452字 | E400 - E5E8 |
| [P U A增补](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=puazb) | 207字 | E600 - E6CF |
| [汉字笔画](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=hzbh) | 36字 | 31C0 - 31E3 |
| [汉字结构](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=hzjg) | 12字 | 2FF0 - 2FFB |
| [汉语注音](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=hyzy) | 43字 | 3105 - 312F |
| [注音扩展](https://www.qqxiuzi.cn/zh/hanzi-unicode-bianma.php?zfj=zykz) | 22字 | 31A0 - 31BA |
| 〇 | 1字 | 3007 |

另外，在上述清洗过程中，所有涉及标点符号的字符都同时包含全角标点和半角标点，如在删除“编者按”语句的时候，实际上是匹配了“编者按：”和“编者按:”这两个不同的字符串，其他有标点符号的情况与此同理。

### 使用My SQL建立语料存储数据库

考虑到数据库的易用性笔者决定使用MySQL存储已经处理好的语料。

My SQL有个“历史悠久”的Bug：它从2003年的4.1版本开始支持UTF - 8编码，早于我们今天使用的UTF-8标准（即RFC 3629）的制定时间。现在标准的UTF-8中每个字符最多有4个字节，但My SQL的UTF-8只有3个字节。这导致在MySQL中使用“utf8”实际上不表示任何标准字符集。由于MySQL已经在实际环境中运行很长时间，直接修改会导致所有用户不得不重构数据库，MySQL开发者在2010年发布了“utf8 mb4”来表示标准的UTF-8字符集。因此在MySQL中“utf8 mb4”才是真正的UTF-8，它的“utf8”实际上是个非法字符集。这个问题对于只存储常见字的项目来说不是问题，但在本项研究中必须考虑到。

为满足训练和后续研究的需要，本项目需要构建三个数据库：

Raw Poetry：存储未经分类的诗词数据；创建一张表：raw Poetry

Typed Poetry： 存储已标注好类型的诗词数据；创建六张表：

War、Land scape、Fare well、History、Love、Nostalgia。

Si Ku Quan Shu：存储四库全书所有的文本数据。创建一张表：Si Ku Text

其中Raw Poetry和Si Ku Quan Shu中的数据一并用于词向量的训练；Raw Poetry用于后续工作中无监督学习；Typed Poetry用于有监督学习。

数据表的字段设置如下：

表raw Poetry：

id int primary key not null auto \_ increment,

title var char (300),

author var char (10),

theme var char (10),

style var char (2),

scene var char (20),

emotion var char (10),

content text.

Typed Poetry中的各表：

id int primary key not null auto \_ increment,

title var char (300),

author var char (10),

style var char (2),

scene var char (20),

emotion var char (10),

content text.

表Si Ku Text：

id int primary key not null auto \_ increment,

book var char (50),

title var char (50),

author var char (10),

content text.

数据库构建好后就可以启动爬虫程序，数据库与爬虫的部分直接相连。PyMySQL是Python 3版本中用来连接MySQL数据库服务器的工具，取代Python 2版本中的MySQLdb。使用PyMySQL建立爬虫程序与MySQL服务器的连接，每当获取到一条数据，完成数据清洗，就将其存储到对应的数据表中，对于没有分类标注的诗词数据，其中的对应字段留空。

## 向量训练

本节阐述获取语料后的处理流程。现有的语料经过清洗后，只需要再经过切分就可以输入神经网络中训练词向量。得益于当前深度学习在自然语言处理领域的广泛研究，我们可调用现有模型，无需自行构建该神经网络。

### 古汉语分词规则研究

分词是中文N L P区别于英文N L P的一个独有情况。如前文所述，我们将会对于古文本采用按字切分和按词切分两种处理方式，按字切分不需要算法，按词切分需要寻找适配于古汉语的算法。

分词算法大致分为三类：基于词表的分词算法、基于统计模型的分词算法、基于序列标注的分析算法。

基于词表的分词算法需要一个强大的词典。正向最大匹配法（FMM）对于一段输入的文本，将会从左往右，以贪心的方式切分出当前符合词典的最大的词。负向最大匹配法（BMM）即与正向匹配相反。

基于统计模型的分词方法中比较典型的是N-gram。对于一段固定的句子，对其进行切分的路径是有限条，利用概率学方法可以从统计信息中求解出一条概率条件最大化的路径，该路径即是最佳分词路径。

基于序列标注的分词算法的核心思想是由字构词，把句子中的每个字分类标注为词首、词中、词尾或单字成词，让句子分词问题转变成字分类问题。需要分好词的语料进行有监督训练。具体实现包括基于HMM的分词方法和基于CRF的分词方法等。

上述的分词算法都有出色的实践案例，在实习生产和研究中五款中文分词包脱颖而出，分别是Jie ba、Snow NLP、Pku Seg、THULAC和Han LP。经过对比观察，笔者决定使用Jie ba作为分词工具，在不同条件下视情况换用其他分词工具，比较其优劣。在进行向量训练时，按照需要的数据，从对应的数据库中一条一条地取出语料，进行分词，即可输入到训练器中去。

### 使用自然语言处理工具Gensim进行向量训练

语料预处理的最后一步就是通过训练得到我们需要的数据——词向量。

Gen sim是一款开源的第三方Python自然语言处理工具包。它支持多种主题模型算法，包括LSA，LDA，TF - IDF，和Word 2 Vec等。笔者使用Gen sim中的Word2Vec进行向量训练。训练中主要的参数设置如下：

Line Sentence：语料的输入端，一行就代表一句话，行中的单词已经切分好，以空格分开。无用的字符已提前去除。对于大语料库，应当从磁盘迭代地读取句子。

size：训练好后词向量的维度，设置为100；

window：训练过程中扫描上下文的窗口的宽度，设置为3，也就是考虑目标词的前3个词和后3个词；

min - count：每个词语在所有文档中的最小出现次数，如果小于该值则被丢弃。默认值为5；

workers：训练模型时的线程数，设置为3；

sg ({0, 1}, optional)：模型的训练算法，0使用CBOW，1使用Skip - gram。设置为1；

alpha：训练的初始学习率，设置为1；

min \_ alpha：随着训练的进行，学习率线性递减的最小值，默认为0.0001；

iter：训练中的迭代次数，默认为5，设置为10。

使用语料Ancient \_ Chinese \_ by \_ Zi.txt和Ancient \_ Chinese \_ by \_ Ci.txt分别输入，经过一段训练后得到各自的model，将其命名为Ancient \_ Chinese \_ Zi \_ Vec.bin和Ancient \_ Chinese \_ Ci\_ Vec.bin存储至硬盘中。

## 本章小结

本章详细阐述了数据准备工作中的细节。

首先选用词向量作为语料处理的根本方法，选择Google的Word 2 Vec作为词向量的训练算法，选择其中的Skip - gram模型。考察了现有的文档向量算法，综合其在古文本中的表现后，笔者决定自行定义从词向量出发映射到文档向量的算法。

本章分析了古汉语和现代汉语的不同，决定抛弃现有的词向量模型，从零出发单独训练古汉语词向量；分析了古汉语义原模糊的问题，决定同时采用字本位和词本位训练向量；阐明了使用繁体中文作为语料的原因，并因此讨论了各种中文编码标准对繁体中文的支持情况。

本章叙述了语料选择的过程，阐明了诗词语料不足而引入四库全书增加语料的考虑；讲述了通过网络爬虫获取全部语料的过程，对语料做数据清洗的过程，数据库构建和语料入库的过程。

本章讲述古汉语分词的规则，语料分词的过程。全部准备工作得当后，使用Gensim工具训练得到最终成果：繁体中文古汉语字向量模型、繁体中文古汉语词向量模型。

后续研究将在此基础上进行。

# 文本分类算法的使用和结果分析

本章将使用当前实践中高效的机器学习文本分类算法对语料进行文本分类。研究以词向量构建文本向量的不同方法，将其代入分类器实践中；使用不同的机器学习算法分类器进行实验和改进，观察效果并总结优劣。

## 文档向量化方法研究

在2014年谷歌团队的Quoc Le和Tomas Miko lov提出了Doc 2 Vec方法，也叫Paragraph 2 Vec，Sentence Embeddings，它与Word 2 Vec一样，也是通过上下文嵌入的方法将文本转化成向量，不同的是Doc 2 Vec不限制文本的长短，通过滑动窗口阅读并记忆文本产生训练集，用上下文预测文本的分布概率。和Word 2 Vec一样它同样有两种训练模式：PV - DM或者PV - DBOW，对于大多数任务，PV-DM的方法表现更好。此方法集成在了开源包Gensim中。

笔者注意到，Quoc Le等的Doc 2 Vec方法是建立在上下文嵌入后概率分布计算之上的，这种算法对于具有较强的模式化路径的现代语言来说足够可靠，但对于行文风格普遍天马行空的古诗文并不见得十分有效。因此需要引入其他的文本向量化方法来与之比较。

通过词向量构建文本向量的最常用方法是将文本中所有词向量相加求平均，这种方式一般被称为“横向扩展”，依据是词向量中的每一个维度往往代表了不同语义层次的表达程度，相加的结果通常体现了文本在某些语义层次上的强表达，以及在另一些语义层次上的弱表达。依据此可以对文本整体语义层次的表达有一个大致的把握，有利于直接进行分类。

考虑到不同字词在文本整体含义上的分量不同，在“横向扩展”的基础上可以增减某些字词的权重，来调整文档向量的准确度。首先用统计方法分类出的高频词，中频词和低频词，高频词往往是一些常用字和各种类型诗词中都会出现的常见意象，对提高区分度无益，低频词在词向量训练中表现不足，语义表达可能存在偏差，因此中频词是最适合提高计算权重的。这种方式被称为“横向加权扩展”。

在深度学习中有一种文本向量预处理方法，对于长短不一的文本构成的二维矩阵，选一个归一化长度，超长的部分截断，不足的部分用0补齐。这里也可以借用此方法，把所有词向量按顺序连接成一个一维向量，超长截断不足补0。考虑到大多数诗词的长短，这里的长度选择为40，也就是选诗词前40个词的向量拼接一个4000维向量，对于字向量的情况，选择前70个字拼接成7000维向量，不足的部分一律补0。这种方式被称为“纵向扩展”。

由于古典诗词文本的特殊性，学界尚未有较成熟的文档向量化方法，笔者将上述方法逐一代入后续的机器学习算法实验中以验证它们的效果。

## 各机器学习分类算法及其效果评估

在开始实验之前需要定义好评估分类效果的指标，以便对算法进行评价和改进。这里采用信息检索领域常用的几个指标。首先机器学习的测试过程中总共有4种情况：

True Positive, T P: 预测为正，实际为正，真阳；

True Negative, T N: 预测为负，实际为负，真阴；

False Positive, F P: 预测为正，实际为负，假阳；

False Negative, F N: 预测为负，实际为正，假阴。

通常基于这四种情况可以设计出一下若干指标来评价算法的好坏：

精度Accuracy，指所有被正确分类的样本占总样本数的比例。

准确率Precision，又称查准率，指被预测为正例的样本中，确实是正例所占的比例。下文简写为P。

召回率Recall，又称查全率，指所有实际是正例的样本中，被成功预测为正例所占的比例。下文简写为R。

一个合格的机器学习模型需要同时具备良好的准确率和召回率，但这两大指标在很大程度上是互斥的，如果强行调整模型和参数使得一个指标接近100 %，那另一个指标就会变得非常差，实际工程中需要的是使两个指标能够同时处于较高的水平。为了使两个指标达到均衡性，这里引入了F 1得分指标，它是准确率和召回率的调和均值（Harmonic mean）

这三项指标能够很好地帮助我们评估分类结果的准确性，但在具体到分类细节上，特别是某类样本中有多少被错误地预测成了其他样本，有多少其他样本被错误地归类为该样本这种程度的细节，则无法呈现给我们。

混淆矩阵（Confusion matrix）是一种用于帮助理解分类模型性能的数据表，它有助于我们理解每类数据都被或正确或错误地预测成了哪些类别，我们可以从中发现某些分类的效果比其他分类更差，这有助于我们有的放矢地调整参数，更有目的性地优化改进算法。

在后续的实验中，笔者将对每一种算法采用准确率P、召回率R和F 1得分指标作为评判好坏的依据。对于需要改进的算法，使用混淆矩阵展示具体的分类细节。

### 朴素贝叶斯分类及其效果评估

朴素贝叶斯（Naïve Bayes）是一种基于贝叶斯定理和特征条件独立假设的机器学习算法。它来源于古典数学概率学，有着扎实的理论基础和丰富的实践案例。特征条件独立假设即认为样本空间中的事件发生概率各自独立，不存在互相影响。基于此假设的条件概率即为

以此出发可得推论

贝叶斯定理是对独立事件A和B的条件概率的描述，其公式为

上式的分母P(A)可由全概率公式改写为

假设 x={f 1, f 2, f 3, ... ,f m}是一个待分类项，每个f是该样本的一个特征，所有的样本构成样本集合X=｛x1, x 2, x 3, …,x n｝。样本标记包含k种类别，即Y=｛y1, ,y2, y3, …,y n｝。如果现在来了一个新样本x，我们要怎么判断它的类别？从概率的角度来看，这个问题就是给定x，它属于哪个类别的概率最大。那么问题就转化为求解P(y1 | x),P(y2 | x),...,P(y k | x)中最大的那个，即求后验概率最大的输出P(y k | x)。代入贝叶斯公式（），即可求解出最大输出概率，完成对新样本的分类。

在机器学习的实践中，主要包含三种朴素贝叶斯的实现类型：高斯朴素贝叶斯（Gaussian Naive Bayes）、伯努利朴素贝叶斯（Bernoulli Naive Bayes）和多项式朴素贝叶斯（Multinomial Naive Bayes）。

高斯分布即正态分布，对于单变量x∈(−∞,+∞)符合连续的正态概率密度曲线。由于事前假设所有变量符合独立事件概率分布，那么多元高斯分布就可以表达成单元高斯分布概率密度函数的连乘积。高斯朴素贝叶斯是假设条件概率P(x)是多元高斯分布，同时假设各个维度的特征值都是连续变量。

伯努利分布即（0，1）分布，或称两点分布。伯努利朴素贝叶斯需要定义一个二值化的方法，该方法会接受一个阈值并将输入的特征二值化（1，0）。当然也可以直接采用Multinomial NB，但需要预先将输入的特征二值化。伯努利多项式将文档的生成模型P(x) 假设服从为多元伯努利分布，由贝叶斯公式求解。

将伯努利分布的单变量扩展到d维向量，则将得到多项式分布。假设给定文档类型的基础上文档生成模型是一个多项式分布。由与伯努利一致的方法用贝叶斯公式可以求解最大概率。

笔者通过简单的实验确认了高斯朴素贝叶斯是这三者中较为适合的分类器，并以此设计了朴素贝叶斯的实验。将数据集中已分类标注的诗词作为数据输入，训练集和测试集按3：1划分，使用Python的机器学习标准库Sk learn调用分类器，将不同的向量化后的数据依次进行实验，得到以下数据。

高斯朴素贝叶斯分类器对不同向量化古诗词的分类效果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 向量化方法 | | 准确率P | 召回率R | F - 1评分 |
| 字本位 | 横向扩展 | 70. 32 | 68. 74 | 69. 52 |
| 横向加权扩展 | 74. 72 | 71. 15 | 72. 89 |
| 纵向扩展 | 61. 74 | 62. 72 | 62. 23 |
| Doc 2 Vec | 64. 04 | 66. 53 | 65. 26 |
| 词本位 | 横向扩展 | 66. 13 | 59. 98 | 62. 91 |
| 横向加权扩展 | 68. 61 | 67. 90 | 68. 25 |
| 纵向扩展 | 54. 05 | 48. 63 | 51. 20 |
| Doc 2 Vec | 61. 74 | 60. 68 | 61. 21 |

### k最近邻算法分类及其效果评估

K最近邻算法（k - Nearest Neighbors, k NN）是一种既可以用于回归又可用于分类的常见算法。其基本思路是，每一个样本都可以由与它距离最近的k个样本来表示。k NN是一种懒惰学习（Lazy learning），也就是说它没有实质的训练过程，所有的训练样本被输入样本空间记录其特征和分类标注，因此它的训练时间为零。当有新样本输入时，再计算与新样本特征最接近的k个样本，以此预测新样本的信息。在分类任务中通常采用“投票法”，即由计算出的k个样本中出现次数最多的样本标记作为最终的预测结构。因此k的取值一般是奇数。在实际运用中，由于不同距离的样本的参考价值不尽相同，为了使得距离越近的样本具有更大的参考价值，可以使用基于距离大小的加权平均值或加权投票进行最终的预测，这种改进被称为k加权最近邻算法。

首先需要定义样本空间的距离，对于二维空间，可采用欧氏距离公式

对于高维空间，可由欧式距离引申出闵可夫斯基距离（Min kowski distance）,设空间维度为p，则有

利用空间距离对新样本在样本空间中的k最近邻距离进行计算，就可以实现样本的分类。在本研究的情景中考虑到不同距离的文本其权重理应有所差异，这里选择加权最近邻算法。

不同k值的表现形式

另待解决的一个问题就是k值的选择，过小的k值会导致异常点的影响被放大，造成方差过高，换句话说，k值较小就意味着整体模型变得复杂，容易发生过拟合；而过大的k值会导致远距离上低参考价值的样本被过多的考虑，偏差增大，换句话说，k值较大就意味着整体模型变得简单，容易发生欠拟合，同时也增加计算开销。因此需要通过实验确定一个合适的k值。

使用Sk learn中的k NN方法，将训练集和测试集按3：1划分，得到结果如下所示。

k加权最近邻分类器对不同向量化古诗词的分类效果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 向量化方法 | | 准确率P | 召回率R | F - 1评分 |
| 字本位 | 横向扩展 | 71. 66 | 65. 03 | 68. 18 |
| 横向加权扩展 | 75. 64 | 68. 06 | 71. 65 |
| 纵向扩展 | 54. 51 | 49. 32 | 51. 79 |
| Doc 2 Vec | 66. 59 | 57. 28 | 61. 59 |
| 词本位 | 横向扩展 | 62. 19 | 53. 36 | 57. 44 |
| 横向加权扩展 | 69. 37 | 66. 38 | 67. 84 |
| 纵向扩展 | 44. 18 | 46. 86 | 45. 48 |
| Doc 2 Vec | 52. 26 | 62. 94 | 57. 10 |

### 支持向量机分类及其效果评估

支持向量机（Support Vector Machine, S VM）是一种常见的二分类模型算法，属于有监督学习分类算法，主要用于各种经典的二分类问题研究中。它的根本方法是在特征空间中寻找最大间隔的超平面，此时它是一种线性分类器。S V M还包含核函数技巧，可以使得它的分割超平面变成超曲面，成为非线性的分类器。S V M的训练核心就是特征空间间隔最大化，在数学上转化成一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最小化问题。S V M的学习算法就是求解凸二次规划的最优化算法。多分类问题可以转化为多个二分类问题依次解决。



对于给定的样本集空间

其中是n维的向量，是样本分类标注。寻找一个空间间隔最大，泛化能力最强的超平面，将不同样本分割开来。

对于给定的数据集T和超平面, 定义样本点的关于超平面几何距离为

超平面关于所有样本点的几何间隔的最小值为

如果存在这么一个超平面，使得样本按上述规则分类成功，则

使等号成立的几个样本点称为“支持向量”，两个异类支持向量到超平面的距离之和为γ，我们要找到具有“最大间隔”的超平面，即

求解式X即可得到分类器模型

上述推导说明了样本集在线性可分的情况下，如何选择超平面对样本集进行分类。实际情况中可能会遇到线性不可分的问题，这时候就需要核函数将样本空间映射到更高的维度来实现线性可分。

将线性不可分空间映射为高维线性可分空间

核函数（Kernel Function）是一种特征空间映射到高维度的映射关系的内积。常见的核函数有线性核（Linear Kernel）、多项式核（Polynomial Kernel）、高斯核（Gaussian Kernel）、拉普拉斯核（Laplacian Kernel）、西格玛核（Sigmoid Kernel）等，有了上面两次实验的经验后，笔者采用字本位横向加权扩展的文本向量做了一个简单实验比较各个核函数的优劣，结果显示这里采用高斯核效果较好：

与上文相同，这里使用Sk learn中的S VM分类器，将训练集和测试集按3：1划分，得到结果如下所示。

支持向量机分类器对不同向量化古诗词的分类效果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 向量化方法 | | 准确率P | 召回率R | F - 1评分 |
| 字本位 | 横向扩展 | 72. 45 | 70. 95 | 71. 69 |
| 横向加权扩展 | 79. 47 | 76. 89 | 78. 16 |
| 纵向扩展 | 61. 96 | 64. 35 | 63. 13 |
| Doc 2 Vec | 71. 37 | 70. 93 | 71. 15 |
| 词本位 | 横向扩展 | 66. 25 | 62. 71 | 64. 43 |
| 横向加权扩展 | 72. 94 | 71. 64 | 72. 28 |
| 纵向扩展 | 53. 22 | 49. 51 | 51. 30 |
| Doc 2 Vec | 62. 76 | 63. 99 | 63. 37 |

## 使用字向量集合构建主题域进行文本分类的研究

上述研究的过程中显现出了古诗词文本分类的两大问题，其一是关键词难以提取，一首古诗往往惜墨如金，字字珠玑，往往没有哪个字词相较于其他字词有着特别的权重，另一方面诗词文本极短，每个词汇几乎都不会重复出现，这导致词频统计类算法如TF - IDF无用武之地，结果就是我们很难从中提取出关键字词；其二是文本向量难以构建，无论是词矩阵还是词向量映射构成的文本向量，其作为特征值用于机器学习算法的准确率都不尽如人意，笔者认为这是因为构建文本向量的过程中损失了过多的文本细节信息所导致的。因此笔者构想了一种直接使用复数词向量相似度比较的分类方法。

### 主题域的构想

古诗词的创作极端讲究“炼字”，正所谓“吟安一个字，拈断数茎须”，为了准确地表达心中意境，诗人们往往费尽心思。我们所熟知的“推敲”一词，便是诗人贾岛和韩愈讨论诗句“僧推月下门”更好还是“僧敲月下门”更好而留下的典故。这说明诗词中的每一个字都应当得到足够的关注，被认证对待。笔者认为，古诗词的主题并不由诗文中一两个字词体现，而是相对均匀地分布在每个字词之间。

### 近似关键字搜索算法

### 字向量加权相似计算主题相似度

### 结果评估

## 关于各算法优劣的比较

从整体上来看，在古诗文分类问题上机器学习算法的表现差强人意。F - 1评分最高的也只有78.16%，不到百分之八十，其他各类组合平均效果在55%到70%之间。这意味着古文本的复杂性和特殊性使得经典学习学习算法的表现欠佳，我们需要更加现代化的方案来处理古典诗词的分类问题。但从上述研究中仍然可以提取出许多有价值的信息，值得对每一项数据详细分析。

朴素贝叶斯分类器和k最近邻分离器的表现差距不大，在支持向量机中使用高斯核函数的分类器比线性分离器效果高大约5个百分点。使用非线性核函数的支持向量机是最好的分类器。

横向加权扩展是最好的文档向量化方法，纵向扩展效果最差，Doc 2 Vec方法差强人意。Doc 2 Vec是谷歌团队的成熟产品，这再一次说明了在现代文中久经考验的工具和算法并不一定适用于古文。

在几种分类器中，字本位的表现明显优于词本位。在各个算法内部比较，按字分词的准确率比按词汇分词高出5到10个百分点。造成这种现象的原因有二，首先古典诗词极端讲究“炼字”，正所谓“吟安一个字，拈断数茎须”，为了准确地表达心中意境，诗人们往往费尽心思。我们所熟知的“推敲”一词，便是诗人贾岛和韩愈讨论诗句“僧推月下门”更好还是“僧敲月下门”更好而留下的典故。诗人们往往通过凝练的单字组合表达情景，多数词也可以拆分成复数单字理解而不损失含义，只有极少数的特有词汇无法拆分。另一方面现有的分词算法和工具在古文中表现不佳。笔者观察了大量诗词的分词效果，发现断字错误，分词颗粒过大等诸多问题，比如李白《将进酒》中的一段分词效果如下：

“君 不见 黄河 之 水 天上 来 ， 奔流 到 海不复 回 。 君 不见 高堂 明镜 悲 白发 ， 朝如 青丝 暮成 雪 。 人生 得意 须 尽欢 ， 莫使 金樽空 对 月 。 天生我材必有用 ， 千金 散尽 还 复来 。”

其中“海不复”、“金樽空”是明显的分词错误，“朝如”应分为状语“朝”和动词“如”，“暮成”应分为状语“暮”和动词“成”，是典型的颗粒过大，“天生我材必有用”更是直接被当成了一个词，这可能是因为在预训练分词工具的大量语料中，这句诗被广泛地作为成语使用，结合词典后被当作了一个词。为了验证这个猜测，笔者又拿白居易的《琵琶行》做了实验，果然“犹抱琵琶半遮面”，“同是天涯沦落人”都被当成了一个词。尽管这些问题都可以通过微调参数在表面上解决，但这足以反映当前分词策略在诗词中的局限性。

此结果已经显著地表明字本位更适合于古典诗词的文本分析，后续部分将不再采用按词汇分词的向量构建方法。

## 本章小结

本章设计和实现了经典机器学习算法对古典诗词的分类实验。

本章提出了构建文本向量的四种方式，除了使用成熟的文档向量算法实现工具Doc 2 Vec外，另外构建了三种从字向量到文档向量的方法：横向扩展，横向加权扩展和纵向扩展，将这四种方法，结合字本位和词本位两种基础向量构建，共八种语料向量化的思路代入到各机器学习算法的实现中，比较其准确率。

本章选用了朴素贝叶斯，k加权最近邻，支持向量机三种经典的机器学习算法作为分类器，研究了它们各自的原理和实现。在它们内部不同的子项中选择了最适合的来使用，将第三章中准备好的数据作为训练集和测试集代入实验。比较分析了各项实验数据后得出了若干结论：

经典机器学习算法在古诗词分类上表现欠佳；

使用高斯核函数的支持向量机分类准确率最高；

字本位的效果显著优于词本位；

一些在现代文中成熟的算法和工具不见得适用于古文。

为了进一步提高准确率，我们需要更加强大的理论和工具来克服经典机器学习算法中遇到的困难，后续研究将在此基础上进行。

# 搭建神经网络实现主题与情感识别

本章主要介绍当前实用的自然语言处理中的深度学习方法，以及成熟的神经网络预训练模型，将其使用于古典诗词文本分类的实验中，调整模型和参数并进行横向比较。本章的语料特征构建将不局限于Gensim Word2Vec方法。

## 神经网络原理

人工智能自诞生之初便希望能复制人类大脑的成功，人脑是一种及其复杂的生物器官以至于脑科学家们至今仍未能解析出大脑的详细结构。人工神经网络便是计算机学家们结合了机器学习方法和对人脑的结构的推测想象而创造出来的。区别于机器学习，基于神经网络的AI研究领域被称为深度学习，如今已在计算机视觉和自然语言处理等领域取得了令人瞩目的成果。

### 神经元

神经元是神经网络的基本构成单位，一个神经元接受来自外部的初始输入或上一次神经网络的多个输出，作为本单元的输入并进行一些计算得到输出值，连接到下一次的一个或多个神经元上去。单个神经元的结构如图5-1所示：

图5-1 人工神经元结构

对于每个输入在网络中都有一个相应权重与之相乘，该权重表示这个输入值在网络中的重要性，神经网络迭代训练就是反复优化所有这些网络参数的值。神经元得到输入后将所有值相加并传入激活函数中，得到的函数值就是神经元的输出值。整个处理流程的表达式是：

### 激励函数

在神经网络的层次结构之间，往往会加入激励函数。

激励函数本质上是非线性方程，之所以加入激励函数是因为一层的神经元经过加权求和后得到输出，如果直接与下一层相连，本质上是线性运算，这样无论无论神经网络有多少层，输出都是输入的线性组合。激活函数给神经元引入了非线性因素，使得神经网络可以任意逼近任何非线性函数，这样神经网络就可以应用到众多的非线性模型中。它会将数据压缩到一定的范围区间内，得到的数据的大小将决定该神经元是否处于活跃状态，即神经元是否被激活，因此它又称激活函数。常见的激励函数有Sigmoid函数，Re Lu函数，tanh函数，Soft Max函数，Dropout函数等等。

在实践中Sigmoid函数常被用于二分类的输出而Soft Max函数常被用于多分类的输出。Sigmoid函数将一个连续的概率转化成（0，1）分布，可表示分类问题中的正负类，其表达式为

Soft Max函数通常作为多分类问题的最后一层输出，其表达式为

它将上一层输入映射到下一层各节点的0到1之间，下一层每个节点即表示一个类别，其输出值表示被预测为该类别的概率，取其中的最大值输出的节点就是最终预测的类别。各节点之和归一化为1，对应多分类概率之和为1。在本项研究中，Soft Max正适合用来做最后的分类预测输出。

Drop out函数在2014年由Geof frey[14]等提出用于降低过拟合的影响。当神经网络中参数过多，而训练样本又不足的时候，训练出来的模型就容易出现过拟合的问题，即模型在训练数据上准确率很高，损失函数值很低；但在测试集上准确率不足，损失函数值很高，模型过于迎合训练数据而导致泛化能力不足。Drop out函数因此被引入缓解这种情况，在前向传播当中，随机让某些神经元以一定的概率断开连接，这样可以使得模型不会过度依赖某些局部特征，以增强模型的泛化能力，如图5-2所示。



图5-2 全连接神经网络（左）加入Dropout的神经网络（右）

### 前馈神经网络

前馈神经网络（Feed for ward Neural Net work, FNN）是一种最简单的神经网络，各神经元分层排列。每个神经元只与前一层的神经元相连。接收前一层的输出，并输出给下一层．各层间没有反馈。是目前应用最广泛、发展最迅速的人工神经网络之一。研究从20世纪60年代开始，目前理论研究和实际应用达到了很高的水平[15]。前馈神经网络的结构如图5-3所示

图5-3 前馈神经网络

一般的前馈神经网络输入层（In put Layer）、隐层（Hidden Layer）和输出层（Output Layer）组成，每一层都由复数个人工神经元组成，每个神经元完成各自的计算并传递信息。其中输入层接收外界的样本输入，一般是特征向量的形式。隐层是中间信息传递和计算过程，可以有多个隐层，也可以没有隐层，前馈神经网络的非线性拟合能力随隐层的增加而增强，但同时计算量也将大大增加。输出层将隐层的结果输出，在分类问题中，输出层输出的就是每个类别的预测概率。

### 卷积神经网络

卷积神经网络（Convolutional Neural Net work, CNN）起源于生物学家对猫大脑的视觉系统的研究[16]，在20世纪60年代，Hubel等生物学家发现猫的视觉皮层存在一种叫感知视野的特征，它扫描接受到的图像的每一个子区域，将其分割成不同的小块传递给大脑视觉细胞处理，这成了日后“卷积核”这一概念的来源。

早期的卷积神经网络主要集中于计算机视觉领域的研究。1980年日本科学家福岛邦彦创造性地提出了neocognitron[17]，从人类视觉系统中引入了许多新概念，将卷积层，池化层等概念加入到神经网络中，试图构建一个像人脑一样运行的图像识别系统。第一个真正意义上的卷积神经网络是Le Cun在1989年提出的Le Net[18]，随后在1998年的提出的改进版本Le Net - 5[19]是第一个成功达到商用级别的RNN。

卷积神经网络的区别于其他网络模型的特点是卷积和池化。

卷积层（Convolutional Layer）的提出源于对全连接网络的优化，在全连接的网络结构中，每个神经元都与相邻层的所有神经元相连，也就是全面地考虑输入输出的全部特征。但在很多情况下，我们往往只需要注重样本的局部特性就可以了，比如在图片识别中，关注于体现图片特点的局部图像上，或者在文本处理中，着重考虑关键字段或文本。卷积核（Convolutional Kernel）可以看做一个微型神经网络，它逐次扫过二维矩阵，对于每个局部都采用相同的权重矩阵进行操作，生产一个新的特征矩阵，因此它本质上就是一种特征提取器，同时由于相同的卷积核权重设置，大大地减少了模型中参数的数量，降低了计算规模。

池化层（Pooling Layer）往往接在卷积层之后，用于压缩数据和参数的规模，降低过拟合。池化采样常用的有最大采样和平均采样，最大采用能体现出局部突出的特征值，是使用最多的采样。池化采样识别了局部的突出信息，舍弃了相对次要的部分，因此有减小特征矩阵降低计算规模，却不损失效果的好处。

近年来卷积神经网络也被成功地运用于自然语言处理领域，将待处理的文本中每个词转化成词向量，将所有词向量按顺序排列起来就得到了与图像处理时相似的二维矩阵。不同的是卷积核不再以二维扫描的方式对整个二维矩阵进行卷积，而是顺着句子方向以一维进行扫描，也就是卷积核的宽度与词向量的长度相等。为了保证所有文本构造的二维矩阵的一致性，通常采用文本超长时超出的部分截断，文本长度不足时补0填充的办法。

### 循环神经网络

循环神经网络（Recurrent Neural Net work, RNN）一般被用于处理序列数据相关的问题。

在一般的神经网络中，神经元与相邻层次连接，同一层中的神经元互相不存在连接。循环神经网络引入了定向循环，在相同层次的神经元中使用次序数据传递，使得每个神经元之间存在状态影响，体现在文本序列中，就是当前词汇的表达受上文的影响。典型的循环神经网络结构如图5-4所示：



在处理流程中，每一时刻的输出都是此刻输入和之前输出值加权之后的结果，用公式表达t时刻的输出为：

表示到第t步时的隐藏状态，它的输出是当前时刻的输出，表示序列中的第t个输入，比如文本序列中的一个词汇，U代表这一步的输入值的加权计算，是上一步的输出，W是上一步输出的在这一步的权重计算。通过这样的方式计算，我们可以认为，当前的结果包含之前的结果，或者说受到之前结果的影响。

循环神经网络也可以分为输入层，输出层和隐藏层，在隐藏层内部节点之间不再是无连接而是次序连接的，这样的结构赋予了循环神经网络近似于“记忆”的特性，也就是说循环神经网络具有分辨前因后果的能力。这样的特性对于强调上下文分析的自然语言处理任务具有很好的适配性。因此循环神经网络在语音识别，机器翻译，音乐生成，文本生成，情感分析，命名实体识别等领域有着广泛的应用。

### 反向传播算法

神经网络与机器学习算法的一个显著差异是，神经网络中的参数规模十分庞大，训练开销惊人。反向传播（Back Propagation, B P）是一种与梯度下降等最优化方法结合使用的用来训练神经网络的基本方法。

## NLP领域成熟的神经网络模型

神经网络的一大特性就是网络结构可以自由组合构建，不像经典机器学习，各种算法的具体实现方法都相对固定。这样的好处是可以有无数可能的网络模型用以适配多种问题，且能根据情况做各种微调，但代价则是各种网络模型性能各异，逐一尝试开销过大。因此我们可以借鉴前人在自然语言处理领域成熟的神经网络模型来搭建我们需要的模型。

### Fast Text

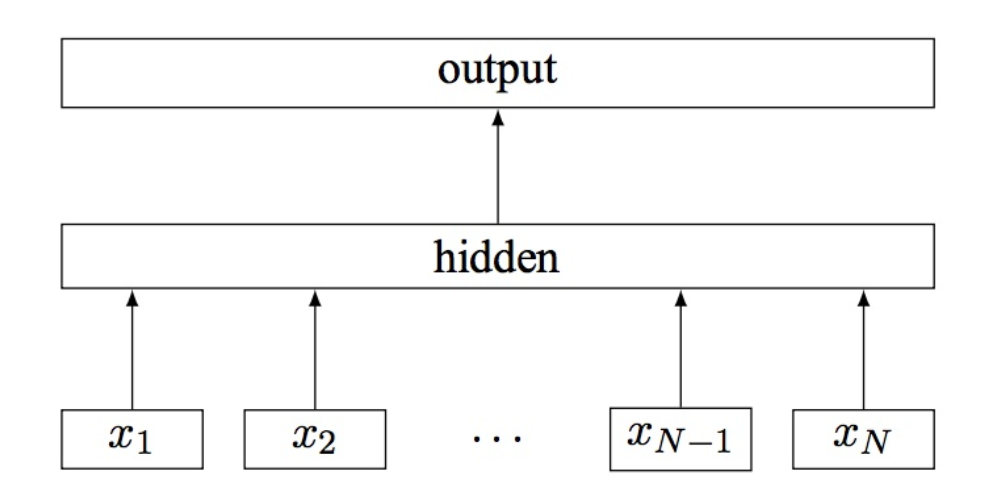
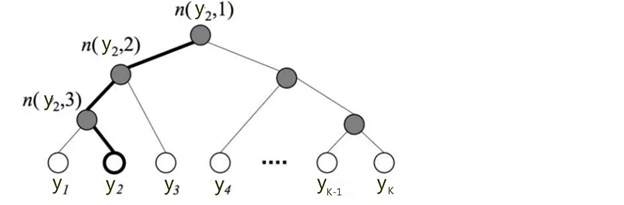
Fast Text是一种快速的文本分类模型，利用N - gram特征构建自己的词表征，不需要事先准备好词向量输入。其结构如下：

图 Fast Text结构图

从图中可以看出Fast Text包含输入层，一层隐藏层和输出层。在输入层输入多个单词的N - gram特征，隐藏层是对多个特征的叠加平均，输出的是Soft Max分类概率。

Fast Text的一大特征是分层次Soft Max，为了降低目标数量过大时线性分类器的高昂开销，可以引入基于霍夫曼树的分层Soft Max。

其中k个叶子节点输出最终预测概率，k-1个内层节点作为参数，从根节点到叶子节点i形成一条路径，长度为，则输出概率可以表示为：

Fast Text由于其快速训练部署的特性和较好的准确率，在短文本主题分类和情感分析上应用广泛。

### Text CNN

卷积神经网络原本是为解决计算机视觉问题而诞生的，2014年Yoon Kim将卷积神经网络应用于文本分类问题，提出了Text CNN[20]。该网络模型的结构简图如下：

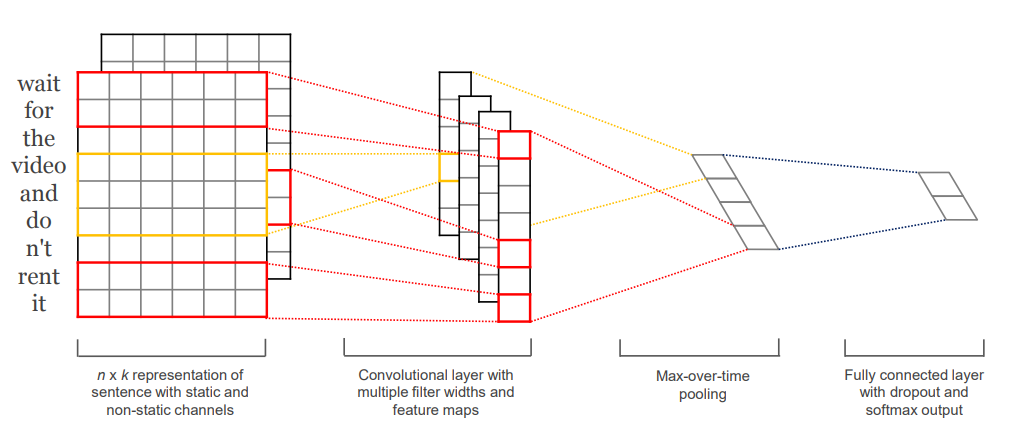


图 Text CNN网络结构（多通道模型）

把文本中每一个单词的词向量堆叠起来，将文本转化成一个n×k的二维矩阵，使用多个不同长度的卷积核来进行卷积，每个卷积核的宽度等于文本矩阵的宽度，也就是所谓的一维卷积（Conv1D）。以此提取出文本中的关键信息，捕捉文本局部相关性。经过激活函数后是一个最大池化层，在这里不同长度的文本会变成定长的特征向量表示。最后经过一个全连接的Soft Max输出每个类别的预测概率。

Text CNN的一大优势在于结构简单，参数数目少，训练速度快，利用已经训练好的词向量基础上再做微调可以取得很不错的结果，在多项数据集上表现良好。

Text CNN模型有具体实现上有4个模型：

CNN - rand: 输入的单词随机初始化，设计好embedding在训练过程中优化参数作为词向量输入；

CNN - static: 直接输入由词嵌入工具训练好的词向量，比如使用C O W模型下的Word 2 Vec工具，并且词向量在后续步骤中固定不变；

CNN - non - static: 同CNN - static一样输入Word 2 Vec等工具训练好的词向量，但在后续训练过程中词向量会被微调；

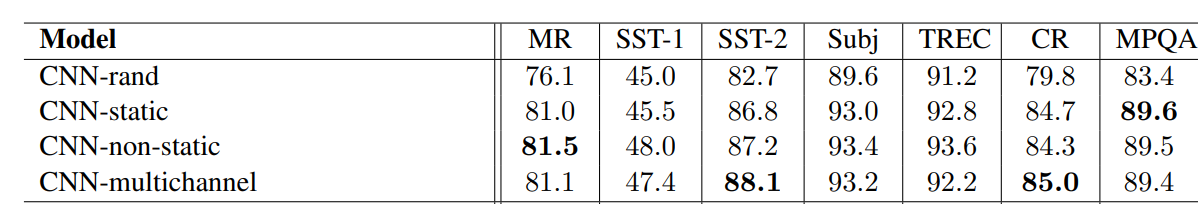
CNN - multichannel: 将CNN - static和CNN - non - static分别作为不同的通道输入模型，类似于计算机视觉中的RGB通道（Channel），属于多通道模型。

图 四种Text CNN模型在七大公开数据集上准确率

Ye Zhang等人在Yoon Kim的Text CNN模型上做了大量的实验后得出一些结论[21]：多通道的Text CNN并没有提升文本分类的能力，在七个数据集中有五个单通道Text CNN表现都要略好于多通道Text CNN；预训练的词向量能较为有效地提升准确率，使得CNN - static表现好于CNN - rand；训练过程中适当的词向量微调是有益的，它可以让词向量更加接近当前的具体任务，这使得CNN-non-static多数时候表现好于CNN-static。

DCNN

### LSTM和BiLSTM

自然语言处理中的一大难题便是长期记忆问题。文本的解析常常依赖于上下文的关联，特别是有些需要预测的信息和与之关联的信息可能在文中相聚很远，比如一段文本中有“我出生在俄罗斯”这样一句话，那么下文中“我能流利地说\_\_\_\_”就应当提高预测为“俄语”的概率。这对我们的网络能“记住”上下文的能力提出了要求。另一方面，文本中总会有大量的无关信息，这些信息如果全部被记忆下来，将会扰乱模型对文本的解析，因此在拥有“记住”上下文的能力的同时，模型还应当有“选择性遗忘”的能力。

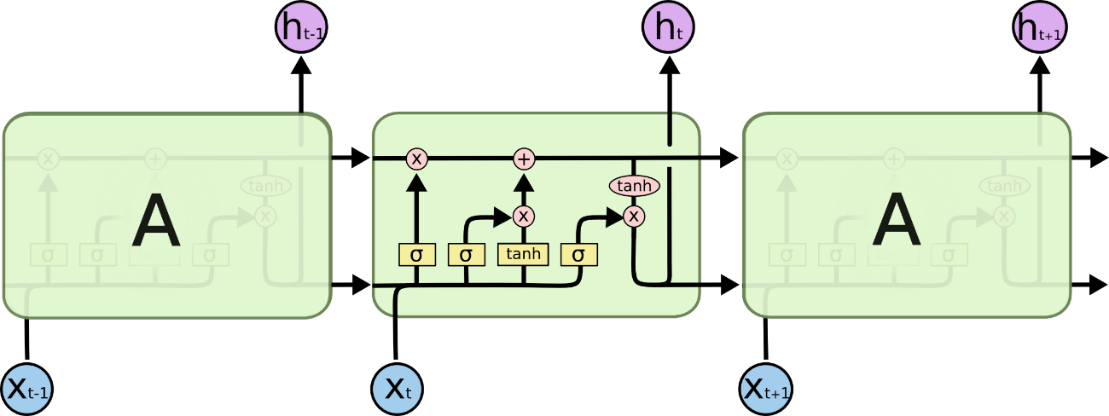
长短时记忆网络（LSTM）就是这样一种特殊的模型。它在循环神经网络的基础之上发展而来，不同的是，在每个神经元里将单一的激活函数换成了复杂的门结构组成的单元，这种单元被称为记忆块（Memory Block），在循环神经网络的的一层中重复出现[22]。记忆块的结构如图所示：

图 LSTM记忆块结构

在LSTM记忆块中，主要有三个门结构组成，它们分别是遗忘门（Forget Gate），决定上一刻的信息中哪些被排除掉；输入门（Input Gate），将筛选后的信息经过sigmoid函数来更新并传递；输出门（Output Gate），经过sigmoid函数并和tanh函数缩放后的Ct值相乘后输出。图中Xt是当前的输入数据，它会与同一层级的上一个记忆块的输出ht-1相结合，进入筛选和激活部分，图中上面一条水平的线被称为单元状态，它负责作为控制信息的传递部分，将控制信息送到控制单元，对当前输入和上一个单元的输出相结合的信息做处理，传递和激活一部分，遗忘另一部分。本单元的输出ht同时被送往下一个单元，重复记忆与遗忘的过程。

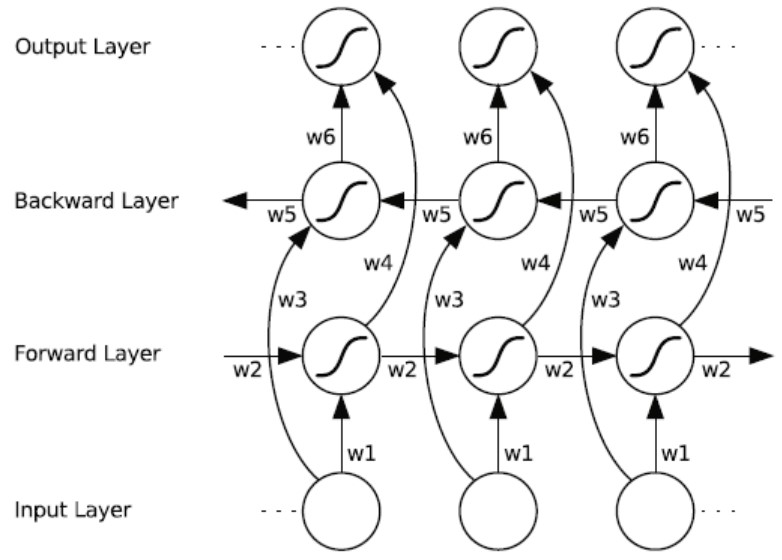
由于在处理长序列问题的优势，LSTM被广泛地应用于文本分类和情感分析，取得了出色的成绩。但单层的LSTM结构实际上只考虑了上文对当前信息的影响，这与同时分析上下文的需求不符，因此双向长短时记忆网络，即Bi LSTM应运而生。Bi LSTM由两个传递方向相反的LSTM隐层堆叠而成，数据同时输入到这两层的记忆单元中，这两个单元同时连接到输出层，其结构如下：

图 Bi LSTM结构

在Bi LSTM中前向和后向隐藏层之间是没有信息传递的，这保证了两层之间是非循环的。图中 Wx表达的是时序处理关系。双向的信息传递和遗忘流程确保了Bi LSTM在文本处理问题上对上下文的充分利用。综合来看，Bi LSTM是进行古典诗词主题分类和情感分析的最佳选择。

## NLP神经网络预训练模型

ELMo

GPT

BERT

## 实验设计与评估

结论

论文的结论是最终的、总体的结论，不是正文中各段的小结的简单重复。结论应该准确、完整、明确、精练。如果不可能导出应有的结论，也可以没有结论而进行必要的讨论。可以在结论或讨论中提出建议、研究设想、仪器设备改进意见以及尚待解决的问题等。

参考文献

著作：[序号]作者.译者.书名[M].版本(第一版不著录).出版地:出版社,出版时间:引用部分起止页.

期刊：[序号]作者.译者.文章题目[J].期刊名,年份,卷号(期数):引用部分起止页.

会议论文集：[序号]作者.译者.文章名[C]. //编者.论文集名,会议地址，会议时间.出版地:出版者，出版年.引用部分起止页.

学位论文：[序号]作者.题名[D].保存地点:保存单位,年份.引用部分起止页.

专利：[序号]专利申请者.专利文献题名[P].国别,专利文献种类,专利号.发布日期:引用部分起止页.

技术标准：[序号]起草责任者.标准代号.标准顺序号——发布年.标准名称.出版地.出版者.出版年份:引用部分起止页.

报纸: [序号]作者.题名[N].报纸名，出版日期(版次)

1. Maron, M.E. and Kuhns, J.L. (1960) On Relevance, Probabilistic Indexing and Information Retrieval[J]. Journal of the ACM (JACM), 7, 216-244.
2. Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Jauvin.A Neural Probabilistic Language Model[J]. Journal of Machine Learning Research 3 (2003) 1137–1155
3. 习近平.在庆祝中国共产党成立95周年大会上的讲话[Z].北京:人民出版社.2016
4. 北京大学计算语言研究所.古代诗词计算机辅助研究系统.http://icl.pku.edu.cn
5. 北京大学中文系全唐诗电子检索系统.http://chinese.pku.edu.cn/tang/
6. 台湾地区元智大学.全唐诗检索系统.http://cls.admin.yzu.edu.tw/QTS/HOME.HTM
7. 周昌乐.心脑计算举要[M].北京:清华大学出版社,2003.
8. 胡俊峰.基于词汇语义分析的唐宋诗计算机辅助深层研究.[D].北京:北京大学,2001.
9. 易勇.计算机辅助诗词创作中的风格辨析及联语应对研究[D].重庆:重庆大学.2005.
10. 梁健楠,孙茂松,矣晓沅,杨成,陈慧敏,刘正皓.基于神经网络的集句诗自动生成[J].中文信息学报,2019,3.
11. 申资卓,杨莹,邵艳秋.基于主题模型的古典乐器诗词文本挖掘[J].中文信息学报.2019.3
12. 吴斌,吉佳,孟琳,石川,赵惠东,李仪清.基于迁移学习的唐诗宋词情感分析[J].电子学报.2016.11.
13. 李奇.古典诗词意境分类方法的研究与实现[D].成都:东华大学.2014.
14. 李良炎,何中市,易勇.基于词联接的诗词风格评价技术[J].中文信息学报. 2005.6.
15. 易勇,郑艳,何中市,李良炎.基于机器学习的古典诗词作者的判别研究[J].心智与计算.2007.3
16. Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, Lawrence Carin. Joint Embedding of Words and Labels for Text Classification [J] .Duke University.2018.
17. Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors[J]. Computer Science.2012.6
18. 张立毅.神经网络盲均衡理论、算法与应用[M]:清华大学出版社,2013.12.
19. David Hubel, Torsten Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex.The Journal of Physiology, Vol. 160, No. 1. 1961.
20. Fukushima K, Miyake S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition[M].Competition and cooperation in neural nets. Springer, Berlin, Heidelberg, 1982: 267-285.
21. Y. LeCun, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation, 1989, 1(4): 541-551.
22. Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 1998, 86(11):2278-2324.
23. Yoon Kim. Convolutional Neural Networks for Sentence Classification[J].arXiv preprint arXiv:1408.5882, 2014.(EMNLP 2014)
24. Ye Zhang, Byron Wallace. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification[J]. Neural and Evolutionary Computing.2016.13.
25. Sak, Haim，Senior, Andrew，Beaufays, Franoise. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition[J]. Computer Science.2014.5.

附录 A

附录是作为论文主体的补充项目，并不是必须的。

论文的附录依序用大写正体英文字母A、B、C……编序号，如：附录A。

索引

按照需要编排分类索引、著者索引、关键词索引等。

作者简历及攻读硕士/博士学位期间取得的研究成果

一、作者简历

蒋俊成，男1994年4月生，四川简阳人，硕士研究生，毕业于北京交通大学。

教育经历：

2012年9月 – 2017年6月就读于北京交通大学电气工程及其自动化专业

2017年9月 – 2020年6月就读于北京交通大学软件工程专业

实习经历：

2018年6月 – 2018年9月 北京新浪科技有限公司 后端开发实习生

交流经历：

2019年2月 – 2019年8月 美国佛罗里达大西洋大学 交换生

二、发表论文

无

三、参与科研项目

无

四、专利

无

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名： 签字日期： 年 月 日

学位论文数据集

表1.1： 数据集页

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 关键词\* | 密级\* | 中图分类号 | | UDC | 论文资助 |
|  |  |  | |  |  |
| 学位授予单位名称\* | | 学位授予单位代码\* | | 学位类别\* | 学位级别\* |
| 北京交通大学 | | 10004 | |  |  |
| 论文题名\* | | 并列题名 | | | 论文语种\* |
|  | |  | | |  |
| 作者姓名\* |  | | | 学号\* |  |
| 培养单位名称\* | | | 培养单位代码\* | 培养单位地址 | 邮编 |
| 北京交通大学 | | | 10004 | 北京市海淀区西直门外上园村3号 | 100044 |
| 工程领域\* | | | 研究方向\* | 学制\* | 学位授予年\* |
|  | | |  |  |  |
| 论文提交日期\* |  | | | | |
| 导师姓名\* |  | | | 职称\* |  |
| 评阅人 | 答辩委员会主席\* | | | 答辩委员会成员 | |
|  |  | | |  | |
| 电子版论文提交格式 文本（ ） 图像（ ） 视频（ ） 音频（ ） 多媒体（ ） 其他（ ） 推荐格式：application/msword；application/pdf | | | | | |
| 电子版论文出版（发布）者 | | | 电子版论文出版（发布）地 | | 权限声明 |
|  | | |  | |  |
| 论文总页数\* |  | | | | |
| 共33项，其中带\*为必填数据，为21项。 | | | | | |