

硕士专业学位论文

作者：

导师：

北京交通大学

2020年3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**

作者姓名： 学 号：

导师姓名： 职 称：

工程硕士专业领域： 学位级别：硕士

北京交通大学

2020年3月

致谢

放置在摘要页前，对象包括：1）国家科学基金，资助研究工作的奖学金基金，合同单位，资助或支持的企业、组织或个人。2）协助完成研究工作和提供便利条件的组织或个人。3）在研究工作中提出建议和提供帮助的人。4）给予转载和引用权的资料、图片、文献、研究思想和设想的所有者。5）其他应感谢的组织和个人。

摘要

中文摘要应将学位论文的内容要点简短明了地表达出来，硕士学位论文一般为500~1000字，博士学位论文一般为1000~2000字。留学生英文版学位论文不少于3000字中文摘要，留学生英文版博士学位论文不少于5000字中文摘要。字体为宋体小四号。内容应包括工作目的、研究方法、成果和结论。要突出本论文的创新点，语言力求精炼。为了便于文献检索，应在本页下方另起一行注明论文的关键词（3-8个），如有可能，尽量采用《汉语主题词表》等词表提供的规范词。图X幅，表X个，参考文献X篇。

**关键词：**

ABSTRACT

一般为1000个左右实词。

**KEYWORDS：**

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

引言（或绪论）简要说明研究工作的目的、范围、相关领域的前人工作和知识空白、理论基础和分析、研究设想、研究方法和实验设计、预期结果和意义等。应言简意赅，不要与摘要雷同，不要成为摘要的注释。一般教科书中有的知识，在引言中不必赘述。

## 4646

标题

学位论文为了需要反映出作者确已掌握了坚实的基础理论和系统的专门知识，具有开阔的科学视野，对研究方案作了充分论证，因此，有关历史回顾和前人工作的综合评述，以及理论分析等，可以单独成章，用足够的文字叙述。正文是学位论文的核心部分，占主要篇幅，可以包括：调查对象、实验和观测方法、仪器设备、材料原料、实验和观测结果、计算方法和编程原理、数据资料、经过加工整理的图表、形成的论点和导出的结论等。

由于研究工作涉及的学科、选题、研究方法、工作进程、结果表达方式等有很大的差异，对正文内容不能作统一的规定。但是，必须实事求是，客观真切，准确完备，合乎逻辑，层次分明，简练可读。

**图**：见《北京交通大学学位论文撰写规范》3.10.4

图应有编号。图的编号由“图”和从“1”开始的阿拉伯数字组成，图较多时，可分章依序编号。

图宜有图题，图题即图的名称，置于图的编号之后。图的编号和图题应置于图下方。图题采用中英文对照，英文（Times New Roman）字体五号，中文宋体五号。居中书写，中文在上。

照片图要求主题和主要显示部分的轮廓鲜明，便于制版。如用放大缩小的复制品，必须清晰，反差适中。照片上应有表示目的物尺寸的标度。

**表**：见《北京交通大学学位论文撰写规范》3.10.5

表应有编号。表的编号由“表”和从“1”开始的阿拉伯数字组成，表较多时，可分章依序编号。

表宜有表题，表题即表的名称，置于表的编号之后。表的编号和表题应置于表上方。表题采用中英文对照，居中，中文在上。英文（Times New Roman）字体五号，中文宋体五号。

表的编排，一般是内容和测试项目由左至右横读，数据依序竖读。

表的编排建议采用国际通行的三线表。

如某个表需要转页接排，在随后的各页上应重复表的编号。编号后跟表题（可省略）和“（续）”，置于表上方。

续表均应重复表头。

**公式**：见《北京交通大学学位论文撰写规范》3.10.12

论文中的公式应另行起，并居中书写，与周围文字留足够的空间区分开。

如有两个以上的公式，应用从“1”开始的阿拉伯数字进行编号，并将编号置于括号内。公式的编号右端对齐，公式与编号之间可用“…”连接。公式较多时，可分章编号。

示例：

 (5)

较长的公式需要转行时，应尽可能在“＝”处回行，或者在“+”、“－”“×”、“/”等记号处回行。公式中分数线的横线，其长度应等于或略大于分子和分母中较长的一方。

如正文中书写分数，应尽量将其高度降低为一行。如将分数线书写为“/”，将根号改为负指数。

示例：

将  写成 1/ 或 

**引用**：见《北京交通大学学位论文撰写规范》3.10.7

论文中引用的文献的标注方法遵照GB/T 7714－2005，采用顺序编码制，并以参考文献形式统一编号。正文中引用文献的标示应置于所引内容最后一个字的右上角，所引文献编号用阿拉伯数字置于方括号“[ ]”中，用小4号字体的上角标。

**注释**：见《北京交通大学学位论文撰写规范》3.10.6

当论文中的字、词或短语，需要进一步加以说明，而又没有具体的文献来源时，用注释。应控制论文中的注释数量，不宜过多。采用文中编号加“脚注”的方式。

标题

标题

序号与题名之间空两格。

正文中的标号按如下编号格式顺序排列：先(1) (2) (3), 再1) 2) 3)，后①②③，标号后避免出现标点符号、英文字母、 PPT中的项目符号。

# 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，该部分见第X章。

## 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 clearfix">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 clearfix">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 clearfix" id="infozj 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 |
| [PUA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 |
| [PUA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MySQL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。

MySQL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MySQL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：rawPoetry

Typed Poetry： 存储已标注好类型的诗词数据；创建六张表：

War、Landscape、Farewell、History、Love、Nostalgia。

SiKuQuanShu：存储四库全书所有的文本数据。创建一张表：SiKuText

其中Raw Poetry和SiKuQuanShu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SiKuText：

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NLP区别于英文NLP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Jieba、SnowNLP、PkuSeg、THULAC和HanLP。经过对比观察，笔者决定使用Jieba作为分词工具。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。

Gensim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Word2Vec等。笔者使用Gensim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256；

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Word2Vec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

## 文档向量化方法研究

## 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, TP: 预测为正，实际为正，真阳；

True Negative, TN: 预测为负，实际为负，真阴；

False Positive, FP: 预测为正，实际为负，假阳；

False Negative, FN: 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F1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F1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={f1, f2, f3, ... ,fm}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, x2, x3, …,xn｝。样本标记包含k种类别，即Y=｛y1, ,y2, y3, …,yn｝。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(yk|x)中最大的那个，即求后验概率最大的输出P(yk|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MultinomialNB，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。

### k最近邻算法分类及其效果评估

K最近邻算法（k-Nearest Neighbors, kNN）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。kNN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（Minkowski 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。

### 支持向量机分类及其效果评估

支持向量机（Support Vector Machine, SVM）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。SVM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。SVM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。SVM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

## 使用字向量集合构建主题域进行文本分类的研究

### 主题域的构想

### 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

# 搭建神经网络实现文本分类

文本

## 标题

文本

### 标题

文本

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

参考文献是文中引用的有具体文字来源的文献集合。按照GB 7714《文后参考文献著录规则》的规定执行。

参考文献以文献在整个论文中出现的次序用[1]、[2]、[3]……形式统一排序、依次列出。

参考文献的表示格式为:

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)

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

包括教育经历、工作经历、攻读学位期间发表的论文和完成的工作等。行距16磅，段前后各为0磅。

一、作者简历

二、发表论文

[1]

[2]

[3]

.

.

.

三、参与科研项目

[1]

[2]

[3]

.

.

.

四、专利

[1]

[2]

[3]

.

.

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。 | | | | | |