1. **项目研究目标、研究内容和拟解决的关键问题**

**2．1 项目研究目标**

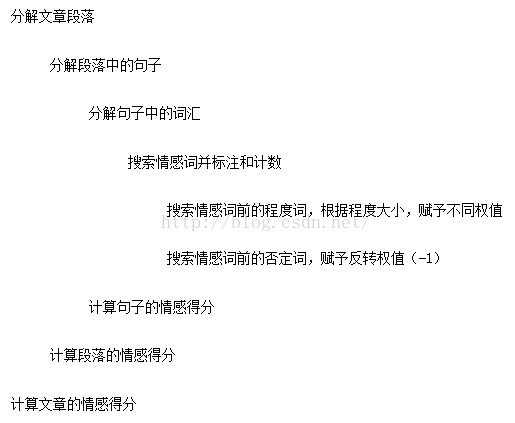
本项目拟实现一个基于python的情感文本分类系统，该系统也能够对当前难以处理的超大文本，短文本，多标签文本进行分类。

**2．2 研究内容**

（1）基于情感字典的情感分析

情感分析对象的粒度最小是词汇，但是表达一个情感的最基本的单位则是句子，词汇虽然能描述情感的基本信息，但是单一的词汇缺少对象，缺少关联程度，并且不同的词汇组合在一起所得到的情感程度不同甚至情感倾向都相反。所以以句子为最基本的情感分析粒度是较为合理的。篇章或者段落的情感可以通过句子的情感来计算。

基于词典的情感分析大致步骤如下：

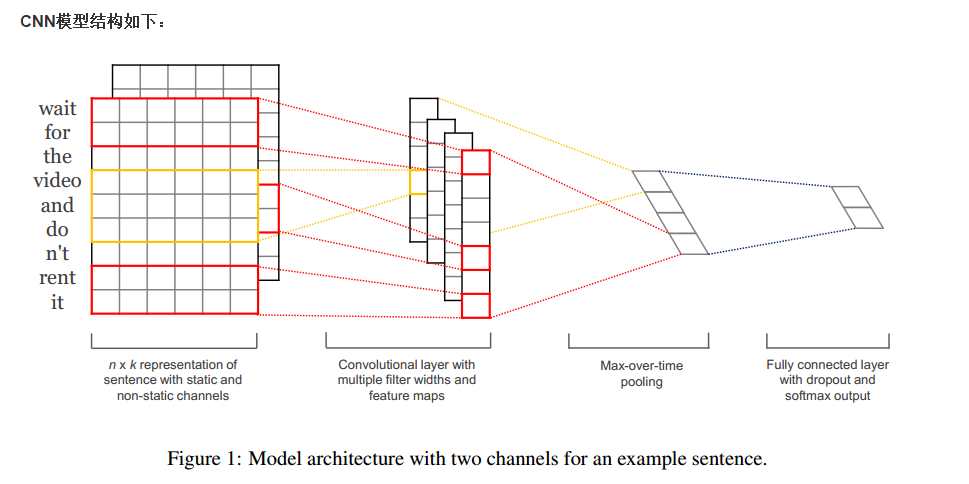


考虑到语句中的褒贬并非稳定分布，以上步骤对于积极和消极的情感词分开执行，最终的到两个分值，分别表示文本的正向情感值和负向情感值。

进过以上的步骤，每篇文章的每个段落的每个句子都会有相应的情感分值，之后针对需求，可以针对句子的分值作统计计算，也可以针对段落的分值作统计计算，得到最后的文本的正向情感值和负向情感值。

（2）基于卷积神经网络的情感分析

CNN的模型结构如下：



一共包括4部分：

1、  输入层：

如图所示，输入层是句子中的词语对应的wordvector依次（从上到下）排列的矩阵，假设句子有 n 个词，vector的维数为  k  ，那么这个矩阵就是  n × k 的(在CNN中可以看作一副高度为n、宽度为k的图像)。

这个矩阵的类型可以是静态的(static)，也可以是动态的(non static)。静态就是word vector是固定不变的，而动态则是在模型训练过程中，word vector也当做是可优化的参数，通常把反向误差传播导致word vector中值发生变化的这一过程称为Fine tune。(这里如果word vector如果是随机初始化的，不仅训练得到了CNN分类模型，还得到了word2vec这个副产品了，如果已经有训练的word vector，那么其实是一个迁移学习的过程)

对于未登录词的vector，可以用0或者随机小的正数来填充。

2、  第一层卷积层：

输入层通过卷积操作得到若干个Feature Map，卷积窗口的大小为 h ×k ，其中 h  表示纵向词语的个数，而  k  表示word vector的维数。通过这样一个大型的卷积窗口，将得到若干个列数为1的Feature Map。(熟悉NLP中N-GRAM模型的读者应该懂得这个意思)。

3、  池化层：

接下来的池化层，文中用了一种称为Max-over-timePooling的方法。这种方法就是简单地从之前一维的Feature Map中提出最大的值，文中解释最大值代表着最重要的信号。可以看出，这种Pooling方式可以解决可变长度的句子输入问题（因为不管Feature Map中有多少个值，只需要提取其中的最大值）。最终池化层的输出为各个Feature Map的最大值们，即一个一维的向量。

4、  全连接+softmax层：

池化层的一维向量的输出通过全连接的方式，连接一个Softmax层，Softmax层可根据任务的需要设置（通常反映着最终类别上的概率分布）。

训练方案：

在倒数第二层的全连接部分上使用Dropout技术，Dropout是指在模型训练时随机让网络某些隐含层节点的权重不工作，不工作的那些节点可以暂时认为不是网络结构的一部分，但是它的权重得保留下来（只是暂时不更新而已），因为下次样本输入时它可能又得工作了，它是防止模型过拟合的一种常用的trikc。同时对全连接层上的权值参数给予L2正则化的限制。这样做的好处是防止隐藏层单元自适应（或者对称），从而减轻过拟合的程度。

在样本处理上使用minibatch方式来降低一次模型拟合计算量，使用shuffle\_batch的方式来降低各批次输入样本之间的相关性(在机器学习中，如果训练数据之间相关性很大，可能会让结果很差、泛化能力得不到训练、这时通常需要将训练数据打散，称之为shuffle\_batch)。

通过使用卷积神经网络，能够学习句子级别或更长粒度的文本，从而可以更好的进行情感分析。

（3）基于spark平台的超大文本分析

对于超大文本我们采用spark平台处理，在文本表示方面,提出了基于流数据的在线分域特征选择算法(OFFS算法)。该算法对向量空间模型进行改进,可以对流数据进行实时的特征提取,快速生成文本向量。解决了传统特征提取算法效率低、耗费内存等问题。在分类器设计方面,设计出基于BP神经网络与OFFS算法相结合的OFFS-BP神经网络文本分类器。该分类器适应了分布式并行计算环境,减少模型训练时间,能够兼顾计算效率和分类准确率。基于Spark平台,实现了 OFFS-BP神经网络分类器。首先利用Spark Streaming子框架实现OFFS算法;然后使用Spark MLlib子框架实现BP神经网络分类器;最后将SparkStreaming和Spark MLlib框架通过Spark编程模型RDD进行无缝连接。

（4）基于卷积神经网络的短文本分类

对于短文本我们采用基于卷积神经网络短文本分类算法，首先通过Word2vec的Skip-gram模型获得短文特征,接着送入CNNs中进一步提取高层次特征,最后通过K-max池化操作后放入Softmax分类器得出分类模型。在实验中,该方法和机器学习方法以及DBN方法相比,结果表明本文方法不仅解决了文本向量的维数灾难和局部最优解问题,而且有效地提高了互联网短文本两级分类准确率,证实了基于CNNs的互联网短文本分类的有效性。

（5）使用改进的最二乘双支持向量机(LSTSVM)多标签分类器对多标签文本进行分类

对于多语义多类别的文本，我们采用将多标签多学习（MIML）的方法应用到中文文本分类，针对中文文本语义丰富的特点使用文本切分粒度，对特征项目进行了独立性假设，数据表示阶段使用多示例句子包的形式进行文本表示，避免基于语义独立性假设带来的语义损失，并进一步优化使其成为主题包，缩短了文本处理的时间在文本分类阶段使用改进的LSTSVM多标签分类器进行分类，对于使用多示例主题包表示的文本，基于退化策略讲多示例 多标签数据通过聚类处成为单示例多标签学习，使用改进的最二乘双支持向量机(LSTSVM)多标签分类器对文本进行分类

2.3.拟解决的问题：

（1）实现对用户上传的文件包括ppt、docx、pdf进行文本提取

对上传的文件中的图片进行提取

（2）将情感词典和卷积神经网络结合来对含有感情色彩的文本进行分类

（3）对超大文本，短文本，多类别文本按照相关方法进行分类

项目研究的实施方案及拟采取的研究方法和技术路线

项目的实施方案与技术路线：

目前常见的文本分类路线如下：

Train（训练）

特征选择

构成矩阵

分词

原中文文本

训练语料词典

具体算法

Test（测试）

得到模型

具体算法

构成矩阵

分词

原中文文本

得到结果

（1）文本提取

项目后台的实现利用的是python语言编写， 利用python的pdfminer解析pdf文件，并对pdf文件里的文本进行提取。同时利用python中的其他包对ppt，docx进行文本提取，利用七牛云进行图片提取处理。

（2）分词

之后利用分词自带的jieba分词对提取的文本进行分词，

（3）特征选择

从分词分到的基本单元中找到能够表示情感的特征词语

（4）具体算法

在通过卷积神经网络方法产生训练集

（5）测试

以构成的图书分类系统来测试分类效果，分类的指标为准确率与召回率。

另外本系统添加了利用djangobb搭建的论坛系统便于使用者们讨论，互相帮助。也添加了笔记功能，便于使用者对使用时的问题，心得进行整理。

项目的可行性分析

本文本分类系统开发利用python语言。Python具有许多特点，使得其在文本分类等科学计算领域 有广泛应用。 良好的可读性，Python语言在设计时即十分重视可读性，在 书写格式统一、尽量采用常用单词而非特殊符号表示关键词、丰富的数据结构、算法和标准库等方面均有体现，从而使得Python 代码易于阅读和维护，有利于涉及多个环节、多人参与、需借鉴 参考他人研究成果等情况的系统开发维护。 内置常用数据结构与算法，不仅有利于提高程序易读性，且 使得开发重心集中在文本系统本身关键算法实现疗面。 丰富的标准库与第三方库，许多辅助环节如字符编码、网络 信息抓取等已有实现可以借鉴，从而町以避免重复开发。 良好的嵌入扩展与“胶水”能力，利用“胶水”特性可将已有 程序“黏合”在一起自动运行；嵌入与扩展可与其它编程语言互 操作．发挥混合语言编程的优势。 相关研究的丰富积累．如同在其它众多科学计算领域一样。 Python在自然语言理解、文本处理和机器学习等领域也有广泛 的研究，有许多优秀的相关模块与研究专著。丰富的研究成果 为进一步研究提供了坚实基础。 综上所述，Python的各种特点使得文本分类系统各个环节 的需求得以顺利实现。

目前已有一些比较成熟的情感词典，如[知网](http://www.keenage.com/html/c_bulletin_2007.htm)，[台湾大学简体中文情感极性词典](http://www.datatang.com/data/44317)。加上卷积神经网络算法（CNN）在不同方面也已经得到了广泛的运用，，所以我们在利用它们来搭建情感文本分类系统时具有相关文献资料可供参考。