这里先把论文中可能写的一些东西作为提纲列出来，避免将来忘记

题目：基于 k-means 思想的词向量的文本分类研究
算法名称：KeyWords Based Document Classification 简称: (KWBDC)

基本思想：分类语料经过清洗、预处理步骤之后，用 google 的 word2vec 工具对待分类语料进行处理得到分类语料

格式方面提醒：
注意学校要求的格式
注意图标的绘制
注意页眉页脚的细节
字体大小、格式

前置部分：

1、封面。

2、扉页。

3、作者声明。

4、中文题目、中文摘要、关键词：摘要应具有独立性和自含性，简短明了；硕士学位论文摘要 300－500 字，博士学位论文摘要 1000 字左右；关键词是用以表示全文主题内容信息款目的单词或术语(3-5 个)。

5、外文题目、外文摘要、关键词：外文题目、摘要、关键词应是中文题目、摘要、关键词的译文。

6、目录。

7、符号、单位、术语等的说明(需要时)。

题目：基于词向量的文本分类探究

摘要：（最后写）

目录

第 1 章　绪论 5
1.1　研究背景及意义 5
1.2　国内外现状 5
1.3　本文主要工作和创新 5
1.4　本文组织结构 5
第 2 章　相关技术与理论 6
2.1　文本分类的概念 6
2.2　文本预处理 6
2.2.1　数据清洗 7
2.2.2　中文分词 7
2.2.3　去停用词 8
2.3　词向量模型 9
2.3.1　One-hot Representation 9
2.3.2　Distributed Representation 9
2.4　文本表示模型 9
2.4.1　布尔模型 9
2.4.2　向量空间模型 10
2.4.3　概率模型 10
2.5　特征选择方法 10
2.5.1　互信息 10
2.5.2　文档频率与逆文档频率 10
2.5.3　信息增益 11
2.6　常用分类方法介绍 12
2.6.1　朴素贝叶斯分类算法 12
2.6.2　决策树分类算法 13
2.6.3　支持向量机分类算法 14
2.6.4　K 最邻近分类算法 14
2.7　本章小结 15
第 3 章　语言模型 16
3.1　统计语言模型 16
3.2　神经网络语言模型 18
第 4 章　Word2vec 工作原理 18
4.1　Word2vec 介绍 18
4.2　Word2vec 训练模型 19

4.3 Word2vec 相关实验 20
第 5 章 基于词向量的分类算法的提出（重点自己叙述） 21
5.1 算法流程图 21
5.2 实验 21
5.3 算法可行性分析 21
第 6 章 基于词向量的算法实验探究（重点在实验结果分析） 22
6.1 实验数据准备（描述语料情况） 22
6.2 实验结果 22
6.3 分析结果 22
第 7 章 总结与展望 23

第 1 章　绪论

1.1　研究背景及意义

1.2　国内外现状

1.3　本文主要工作和创新

I. 提出算法
II. 证明必须是相关语料分类效果才会好
III. 寻找最有配置参数
分析结果

1.4　本文组织结构

简单描述一下论文框架，这个可以等其他章节写完再做总结。

第 2 章　相关技术与理论

本章将介绍文本分类中用到的相关技术，包括文本预处理、文本表达、中文分词和特征选择方法等。通过这些内容，可以更好的帮助理解文本分类常用的技术与方法，也为后续章节的描述打下基础。

2.1　文本分类的概念

文本分类(Text Classification)是指将文本按照内容的不同判别到一个或多个预先确定的类别之中的过程,文本文类是一种有指导的映射过程，也称做监督学习，整个过程中，需要计算机通过已经标注好的数据，学习特征和类别之间的关系模型，然后以此模型来预测新的文本所属的类别。给定一组事先定义好的文本类别集合 C={c1,c2,c3 … cn}和一组文本集合 D = {d1,d2,d3 … dm}，其中 n、m 分别表示文本类别和待分类文本数量。

文本分类的过程就是找到一个文本分类模型 f，f 本质上是从文本集合到文本类别集合的一个映射 f：D → C，如图 2.1 所示：

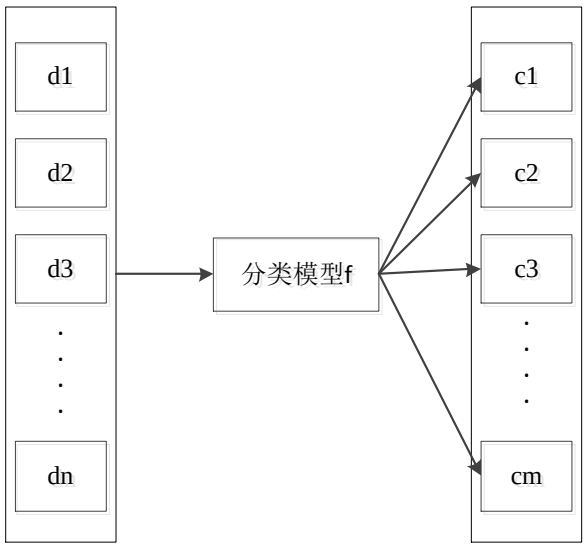


图 2.1 文本集合到文本类别的映射

2.2　文本预处理

文本预处理是指在文本分类中，收集的数据可能是结构化、半结构化数据，我们需要的是文本内容有关的信息,除此之外的其他内容,例如 HTML、XML 标签、

特殊符号等与正文内容无关的信息，是我们不需要的。因此，拿到数据集之后应先对其预处理，提取出真正有用的信息。在本实验中，使用的语料是搜狐语料（要加引用）和中文新闻语料库（要加引用）。

2.2.1 数据清洗

搜狐语料是 XML 格式的结构化数据，需从相应的标签中提取出正文内容，样例数据如表 1；对新闻语料进行分析，从 SQL Server 2008 中读抽取出正文，去除首尾的新闻格式信息，如：“凤凰社实习记者 张三报道”、“更多新闻请点击链接……”等类似无用信息。

<doc>		
<url>	http://news.sohu.com/20120613/n345535702.shtml	</url>
<docno>	e4103f4f49da2142-69713306c0bb3300	</docno>
<contenttitle>	欧洲杯大战在即　荷兰葡萄牙面临淘汰将背水一战	</contenttitle>
<content>	中广网北京 6 月 1 3 日消息（记者王宇）据中国之声《新闻晚高峰》报道，明天凌晨两场欧洲杯的精彩比赛上演，死亡之组 B 组当中两支传统的强队……	</content>
</doc>		

表 1 搜狐语料样例

以上步骤完成后，注意调整文本的编码格式，本实验中文本统一使用 utf-8 编码。

2.2.2 中文分词

分词是指将连续的语句或者段落按照一定的算法拆分成相应的字、词集合的过程。

英文最小的单位是词，词与词之间由空格分隔开，因此英文文本的分词只需按照空格拆分，处理英文标点符号即可。与英文不同，字是汉语的最小单位，字与字之间不是显示的使用空格分隔，但是从语法角度看，词是表示文本的基础，因此我们必须先将文本进行切分得到词的集合。中文分词方法主要包括以下几种：

基于词典的分词方法

基于词典的分词方法，又称机械分词法，其基本原理是将待分词的语句与一个充分大的词典按照一定的规则进行匹配，若在词典中匹配到某个字符串，则认为从语句中识别出一个词。在使用基于词典的分词算法时，一般会设置一个优先

条件，比如语句扫描方向、词语长度等。根据扫描方向不同，匹配字符串方法可以分为：正向最大匹配法、逆向最大匹配法、双向最大匹配法；根据字符串长度的不同，有最长匹配与最短匹配。

基于理解的分词方法

基于理解的分词方法，其思想是试图使计算机像人类一样理解语句，在理解语句的情况下对中文语句进行分词，并且同时可以进行相应的自然语言处理。该算法严重依赖语言学知识，因为汉语语言知识的笼统、复杂，难以将各种语言信息组织成计算机所理解的模式，因此目前还没有成熟的基于理解的分词系统。

基于统计的分词方法

基于统计的分词方法，是目前最流行和最常用的一种分词方法。基本思想是：扫描一段文本，相邻的字共同出现的次数越多，那么它们构成一个词的概率就越大。因此，可以对语料中相邻的、共同出现的字的组合进行统计，计算它们出现的频率，这种频率从某种程度上体现了汉字间组合成词的可能性大小。现在基于统计的分词方法中，常用到的模型有最大熵模型、隐马尔科夫模型、N 元统计模型等。

目前中文分词技术已取得了很多成果，出现了一大批实用的、可靠的中文分词系统。其代表有：基于 lucene 为应用主体开发的 IKAnalyzer 中文分词系统；庖丁中文分词系统；纯 C 语言开发的简易中文分词系统 SCWS；中国科学院计算技术研究所张华平博士推出的汉语词法分析系统 ICTCLAS，现在为 NLPIR 汉语分词系统；哈尔滨工业大学信息检索研究室研制的 IRLAS；另外国内的北大语言研究所、清华大学、北京师范大学等机构也推出了相应的分词系统。

本文实验中中采用的中文分词工具是张华平博士研发的 NLPIR 分词工具包。

2.2.3 去停用词

在文本处理中，我们把那些缺乏实际意义，对文本分析没有利用价值但又频繁出现的词成为停用词，如：“的”、“又”、“了”等。若保留停用词，可能会对正文内容分析产生影响，因此一般会在文本表示中将其删除。在文本处理中，一般将标点符号、单个数字、单个字母、控制符、以及高频单个汉字作为停用词。

本实验中使用哈工大停用词表，并根据对语料特征分析，扩展了此停用词表。

2.3 词向量模型

计算机无法像人类一样直接识别汉字,从本质上来说只能识别 0 和 1。因此,文本分类中一项重要的任务就是,在尽量保证语义信息完整的条件下,将复杂的文本数据抽象为计算机所能处理的数据格式。以下内容介绍常用的文本表示模型:

2.3.1 One-hot Representation

在自然语言处理(Natural language processing,简称 NLP)中,最常见的一步是创建一个词库表并把每个词顺序编号。这实际就是词表示方法中的 One-hot Representation,这种方法把每个词顺序编号,每个词就是一个很长的向量,向量的维度为所选语料对应词库表的大小。其中只有对应位置上的元素值为 1,其余为零。在实际应用中,一般采用稀疏编码存储,采用此的编号。

这种表示方法一个最大的缺点是无法捕捉词与词之间的相似度,就算是近义词也无法从词向量中看出任何关系,如:“西红柿”表示为:[0, 0, 0, 0, 1, …… 0, 0, 0]^{T},“番茄”表示为:[0, 1, 0, 0, …… 0, 0, 0]^{T},即使“西红柿”与“番茄”表示的是同一事物,但是从词向量无法看出两者的联系。此外这种表示方法还容易发生维数灾难[3],尤其是在 Deep Learning 相关的一些应用中。

2.3.2 Distributed Representation

Distributed Representation(分布式表示)最早是 Hinton 于 1986 年提出的[6],可以克服 one-hot Representation 的缺点。其基本思想是通过训练已经分词的语料,将每个词映射成为 K 维空间中的一个点,该点与一个 K 维实数向量对应。

与 one-hot Representation 相比,词的分布式能够使用维度更低的稠密向量表示,K 的值可以人为指定,以几十或者几百为常见,一般远小于词库表的大小。不同的词之间的语义相似度可以通过词对应的实数向量之间的距离来判断,可以使用传统的欧氏距离来衡量。

本文中所使用的词向量即用 Distributed Representation 表示的向量。

2.4 文本表示模型

2.4.1 布尔模型

2.4.2 向量空间模型

2.4.3 概率模型

2.5 特征选择方法

在文本分类任务中，表示文本特征的数目并不是越多越好，如果利用一个特征对分类结果没有较大影响，则称这个特征是没有分类能力的[7]，经验上去掉此特征对分类结果影响不大。因此，在文本分类的任务中需要对文本进行降维，目前主要的方法是进行特征选择，即对当前大量的文本特征按照某种标准进行选择，仅挑选出能够代表文本内容、具有较好类别区分能力的特征。下对目前常用的文本特征选择算法进行简要介绍：

2.5.1 互信息

互信息（Mutual Information）通过计算特征和类别共同出现的概率，来度量特征和类别的相关性[8]。特征 t 和类别 ci 互信息值计算公式如公式（1）所示：

$$\mathrm{MI(t,c)} = log\frac{p(t,c_i)}{p(t)*p(c_i)} = log\frac{p(t|c_i)}{p(t)} \quad (1)$$

公式中，$p(t,c_i)$表示训练集中同时满足包含特征 t 又属于类别 c_i的文本出现的概率。$p(t)$表示训练集中包含特征 t 的文本出现的概率，$p(c_i)$表示训练集中文本属于类别 c_i的概率。当 t 与 c_i完全独立时，$MI(t,c_i)$为 0；当特征 t 在类别 ci 的文本中出现概率高，而在其他类别中出现概率低，即特征 t 和类别 ci 相关性大，将获得较高的互信息值 $MI(t,c_i)$，反之将获得较低的互信息值 $MI(t,c_i)$。使用互信息进行特征选择操作时，常采用最大互信息或平均互信息[9]。

2.5.2 文档频率与逆文档频率

TF-IDF（Term Frequency－Inverse Document Frequency）是一种用于资讯检索与文本挖掘的常用加权技术[12][12]。TF-IDF 是一种统计方法，用以评估一个字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的

频率成反比下降。

词频(Term Frequency，简称 TF)指的是某个给定的词在文档中出现的频率。对于一个特定的词t_i来说，它在第 j 个文档d_j中的 tf 值计算方法为公式(2)：

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (2)$$

其中$n_{i,j}$是该词在文件d_j中的出现次数，而分母则是在文件d_j中所有字词的出现次数之和。为防止 tf 值偏向长文，通常使用的是对文档总词数归一化后的结果。

语料库中包含某个词 W 的文档的数量称为该词的文档频率（Document Frequency，简称 DF）[11]。如果统计词语在整个语料库的文档频率，通过过滤文档频率低与一定阈值的词语可以在一定程度上去除无效词语，但是对于一些常用词，比如“我们”、“但是”等使用频率非常高的但实际包含信息量不大的词，仍会被保留下来。通常是由另一个方法来衡量——逆文档频率（Inverse Document Frequency，简称 IDF)。IDF 的计算方法如公式(3)：

$$\mathrm{idf(w)} = log\frac{|D|}{|\{j:w\in d_j\}|+1} \quad (3)$$

其中$|D|$代表文档总数，$\left|\{j:w \in d_j\}\right|$表示包含 w 的文档总数，分母加 1 是为了防止分母为 0 的情况。IDF 用来衡量词语在整个语料中的重要性，IDF 与 TF 结合起来的 TF-IDF 则可以用来衡量某个词语对某篇文档的重要性，具体计算方法如公式(4)：

$$tfidf_{i,j} = tf_{i,j} * idf_i \quad (4)$$

2.5.3 信息增益

信息增益（Information Gain，简称 IG)，一种基于信息熵的方法[7]。为便于说明，先给出熵与条件熵的定义。

在信息论与概率统计中，熵是表示随机变量不确定性的度量，设 X 是一个有限取值的变量，其概率分布为：

$$\mathrm{P(X} = x_i) = p_i \ i = 1,2,3,\dots,n \quad (5)$$

则随机变量 X 的熵定义为：

$$\mathrm{H(X)} = -\sum_{i=1}^{n} p_i \, log \, p_i \;\; i = 1,2,3,\ldots,n \tag{6}$$

熵越大，随机变量的不确定性就越大。

设有随机变量(X,Y)，联合概率分布为：

$$\mathrm{P(X} = x_i, \mathrm{Y} = y_j) = p_{ij} \;\; i = 1,2,3,\ldots,n;\; j = 1,2,3,\ldots,n \tag{7}$$

条件熵 H(Y|X)表示在已知随机变量 X 的条件下随机变量 Y 的不确定性。随机变量 X 给定的条件下随机变量 Y 的条件熵(Conditional Entropy) H(Y|X)，定义为 X 给定条件下 Y 的条件概率分布的熵对 X 的数学期望[7]：

$$\mathrm{H(Y|X)} = \sum_{i=1}^{n} p_i \; H(Y|X = X_I) \tag{8}$$

特征 t 对数据集 D 的信息增益 g(D,t)，定义为集合 D 的熵 H(D)与特征 t 给定条件下的条件熵 H(D|t)之差，即：

$$\mathrm{g(D|t)} = H(D) - H(D|t) \tag{9}$$

根据信息增益进行特征选择的方法是：计算每个特征的信息增益，比较它们的大小，选择信息增益最大的特征。

2.6 常用分类方法介绍

2.6.1 朴素贝叶斯分类算法

朴素贝叶斯（Naive Bayes classifier）[15]在贝叶斯学习方法中最具实用性的一种，是基于贝叶斯定理与特征条件独立假设的分类方法[7]。

设输入空间为 n 维向量的集合，输出空间为类标记集合{c1,c2,c3,…,ck}。X 是定义在输入空间上的随机变量，Y 是定义在输出空间上的随机变量，P(X,Y)是 X 和 Y 的联合概率分布。训练集为：

$$\mathrm{T} = \{(x_1),(x_2),\ldots,(x_N,y_N)\}$$

通过训练数据集学习联合概率分布 P(X,Y)。具体地，学习以下先验概率分布和条件概率分布。先验概率分布：

$$\mathrm{P(Y} = y_i), \mathrm{i} = 1,2,\ldots,\mathrm{k}$$

条件概率分布：

$$\mathrm{P(X = x|Y} = y_i) = \mathrm{P}\left(X^{(1)} = x^{(1)},\ldots,X^{(n)} = x^{(n)} \middle| \mathrm{Y} = y_i\right), \mathrm{i} = 1,2,\ldots,\mathrm{k}$$

朴素贝叶斯法对条件概率分布作了条件独立性假设：每个特征词之间都是相

互独立互不影响的，因此**公式等价于公式**：

$$\mathrm{P}(\mathrm{X}=\mathrm{x}|\mathrm{Y}=y_i)=\mathrm{P}\left(X^{(1)}=x^{(1)},\ldots,X^{(n)}=x^{(n)}\middle|\mathrm{Y}=y_i\right)$$

$$=\prod_{j=1}^{\mathrm{n}}P(X^{(j)}=x^{(j)}|Y=\ y_i)\,,\mathrm{i}=1,2,\ldots,\mathrm{k}$$

根据贝叶斯定理可得，文本类别$\mathrm{y_i}$的后验概率$P(\mathrm{y_i}|\mathrm{x})$如公式所示：

$$P(\mathrm{y_i}|\mathrm{x})=\frac{P(\mathrm{x}|\mathrm{y_i})P(\mathrm{y_i})}{P(\mathrm{x})}$$

由于公式中分母表示输入 x 在整个训练集中出现的概率值，这个值对于各个类别都是常数，因此可以舍去，与公式结合可得：

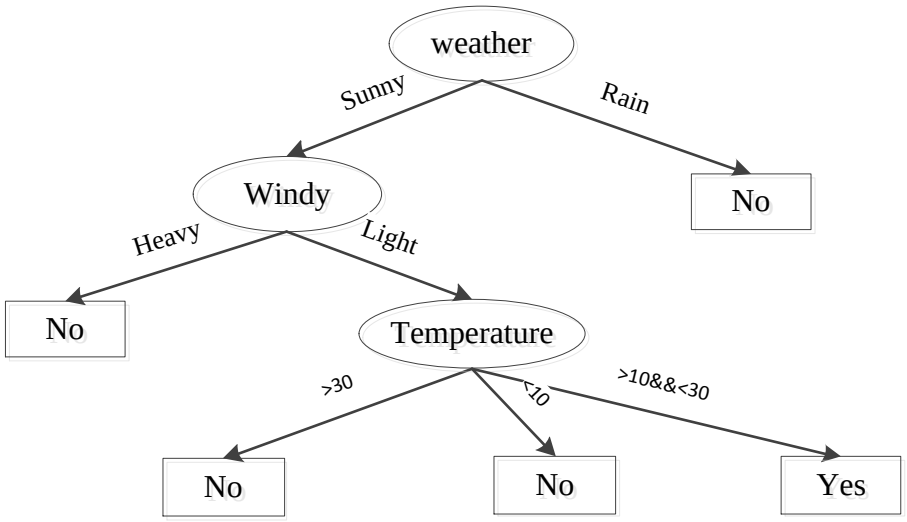
$$P(\mathrm{y_i}|\mathrm{x})=P(\mathrm{y_i})\prod_{j=1}^{\mathrm{n}}P(\mathrm{x}_j|y_i)$$

得到每个类别对于文档 x 的概率，然后找出使$P(\mathrm{y_i}|\mathrm{x})$最高的类别$\mathrm{y_i}$，将文档 x 分类给该类别。

朴素贝叶斯分类算法实验开销相对较小，分类效果也比较好。但是其缺点也是很明显的，算法对各个特征项作相互独立假设，这是不合理的，尤其会使文档语义信息的保留受损。

2.6.2 决策树分类算法

决策树（Decision tree）[13]是一种描述对实例进行分类的树形结构。决策树包括决策节点、分支和叶节点三部分。其中，决策节点代表分类样本的某个属性，在该属性上的不同预测结果代表一个分支，分支表示某个决策节点的不同取值，叶节点存放某个分类标签，代表一种可能的分类结果[14][7]。以决定是否打篮球为例构建决策树，如下图所示：



对训练数据集使用决策树算法进行训练，训练之后保存如上图所示的树形结构。在进行分类时，加载该模型，自上而下进行搜索，直达叶节点，叶节点的类别标签就是该未知样本的类别。

2.6.3 支持向量机分类算法

2.6.4 K 最邻近分类算法

K 最近邻法（k-nearest neighbor，K-NN）分类算法，是一种发展成熟且易于理解的机器学习算法。由 Cover 和 Hart 于 1968 年提出。这是一种基于实例的算法，其基本思想是：在一个样本空间中，如果计算出一个样本与 K 个样本最相似，并且 K 个样本中的大多数属于某一类别，K 最邻近算法则将该文本判别为这一类别中。在这个算法中，得到的最相似的文本都已正确分类。

K 最近邻算法没有训练阶段，它的分类要借助整个训练集，而且训练文档越多，分类计算所需的时间也将线性增加。其中 K 值的选择、距离的度量以及分类规则是影响分类效果的三个重要因素：

（1）K 值的选择：K 值的选择会对算法产生较大的影响。如果 K 值过小，则容易发生过拟合；如果 K 值过大，会增加计算成本，而且学习的误差也会增大。在实际应用中，K 值一般不宜过大，可采用交叉验证选取最有效的 K 值。

（2）距离度量：距离度量指的是待分类样本与其他样本的相似度的一种方法，需要注意的是度量之前需要将样本各个属性值进行规范化处理，防止具有较大值的属性对具有较小值的属性影响过大。计算文本之间相似度的方法有很多，

最常采用的是两个向量余弦值的方法，对于给定两个文档的向量、，它们之间的余弦值公式如下：

$$\mathrm{sim}(d_i, d_j) = \frac{\sum_{k=1}^{n}(w_{ik} * w_{jk})}{\sqrt{(\sum_{k=1}^{n} {w_{ik}}^2 * \sum_{k=1}^{n} {w_{jk}}^2)}}$$

上式中，w_{ik}表示第 i 个文档的第 k 个属性，$\mathrm{sim}(d_i, d_j)$的值越大，则表示两文本的相似度越高。

（3）分类规则：分类规则往往使用多数表决法，即将距离待分类样本最近的 K 个样本中的多数样本所在的类作为待分类样本的类别。

K 最邻近算法的主要缺点是，需要计算待分类文本与训练集中每一个样本的相似度，计算时间的开销比较大；K 值需要明确指定，K 值选择不当同样会影响分类精度。

2.7 本章小结

本章主要对文本分类中的相关技术与理论做了简要介绍。首先描述文本分类的概念，给出文本分类的定义；接着总结了文本预处理中用到的方法，包括数据清洗、中文分词和去停用词等操作；然后对文本表示模型进行了介绍，常见的布尔模型、向量空间模型和概率模型；最后对分类问题中常用的特征选择方法做了基本的说明，包括互信息、文档频率与逆文档频率和信息增益等。

第 3 章　语言模型

3.1　统计语言模型

统计语言模型是用来刻画一个句子出现概率的模型[3]。给定一个由 n 个词顺序组成的一个句子，$S = (w_1, w_2, \ldots, w_n)$，则该句子出现的概率$p(S)$即为统计语言模型。通过贝叶斯公式，可将 $p(S)$ 分解为 $p(S) = p(w_1)p(w_2|w_1)p(w_3|w_1^2)\ldots p(w_n|w_1^{n-1})$。由此可见，要计算一个句子出现的概率，只需要在给定上下文语境的情况下，计算出下一个词为某一个词的概率即可，即 $p(w_i|context(w_i))$，其中 context 为上下文。当一个句子中所有词的 $p(w_i|context(w_i))$计算出来之后，连乘即可计算出$p(S)$。所以，统计语言模型的关键在于找到计算条件概率$p(w_i|context(w_i))$的模型。

其中 Context 即为上下文，根据对 Context 不同的划分方法，可以分为两类：

（1） 上下文无关模型（Context = NULL）

该模型仅仅考虑当前词本身的概率，不考虑该词所对应的上下文环境。这是一种最简单，易于实现，但没有多大实际应用价值的统计语言模型。

$$p(wt|\text{Context}) = p(wt) = Nwt/N$$

这个模型不考虑任何上下文信息，仅仅依赖于训练文本中的词频统计。它是 n-gram 模型中当 n = 1 的特殊情形，所以有时也称作 Unigra Model（一元文法统计模型）。实际应用中，常被应用到一些商用语音识别系统中。

（2）n-gram 模型（Context = wt−n+1, wt−n+2, …, wt−1）

n = 1 时，就是上面所说的上下文无关模型，这里 n-gram 一般认为是 N> = 2 是的上下文相关模型。当 n = 2 时，也称为 Bigram 语言模型，直观的想，在自然语言中“白色汽车”的概率比“白色飞翔”的概率要大很多，也就是 p(汽车|白色)>p(飞翔|白色)。n>2 也类似，只是往前看 n-1 个词而不是一个词。一般 n-gram 模型优化的目标是最大 log 似然，即：

$$\prod Tt = 1pt(wt|wt\text{−n+1}, wt\text{−n+2}, \cdots, wt\text{−1})\log pm(wt|wt\text{−n+1}, wt\text{−n+2}, \cdots, wt\text{−1})$$

n-gram 模型的优点包含了前 N-1 个词所能提供的全部信息，这些信息对当前词出现具有很强的约束力。同时因为只看 N-1 个词而不是所有词也使得模型的效率

较高。

在计算复杂度方面，表 1 给出了 n-gram 模型中模型参数数量随 n 的逐渐增大而变化的情况，其中假设词典大小 N = 200000（汉语的词汇量大致是这个量级）

表 1 模型参数与 n 的关系

n	模型参数数量
1(unigram)	2*105
2(bigram)	4*1010
3(trigram)	8*1015
4(4-gram)	16*1020

在效果方面，理论上是 n 越大越好，今天互联网海量数据以及机器性能的提升使得计算更高阶的语言模型（n>10）成为可能，但是当 n 达到一定程度时，模型效果提升会越来越小。

但是 N-gram 存在一个问题，若训练语料里面有些 n 元组没有出现过，其对应的条件概率就是 0，这会导致计算一整句话的概率为 0。解决这个问题有两种常用方法：

一种是平滑法，最简单的是将出现 k 次的某个 n 元组看做出现了 k+1 次，这样出现 0 次的 n 元组就变成了出现 1 次。

另一种是回退法，即如果 n 元的概率不到，有点像决策树中的后剪枝方法，那就用 n-1 元的概率乘上一个权重来模拟。

除了上面说的问题之外，n-gram 还存在其他问题：

1.n-gram 语言模型无法建模更远的关系，语料的不足使得无法训练更高阶的语言模型。（这篇文章发表时，基本都是 trigram，还没有高阶的模型，不过几年后，互联网的海量数据使得可以训练 10 几阶的语言模型）

2.这种模型无法建模出词之间的相似度，有时候两个具有某种相似性的词，如果一个词经常出现在某段词之后，那么也许另一个词出现在这段词后面的概率也比较大,比如

The cat is walking in the bedroom

A dog was running in a room

如果第一句话里的元组在语料中出现的很多，训练的很充分，第二句话中的元组在语料中出现的少，训练的不充分，那么使用语言模型计算第一句话的概率就比较高，而第二句话的概率就低。

如果有一种方法，能知道 The 和 a 相似，cat 和 dog 相似等等，并且会给相似的词类似的语言模型概率，那么第二句话也可以得到高概率。

3.2 神经网络语言模型

Bengio 等[16]于 2003 年提出一个基于神经网络的语言模型，即神经网络语言模型（Neural Network Language Model，NNLM），如下图所示：

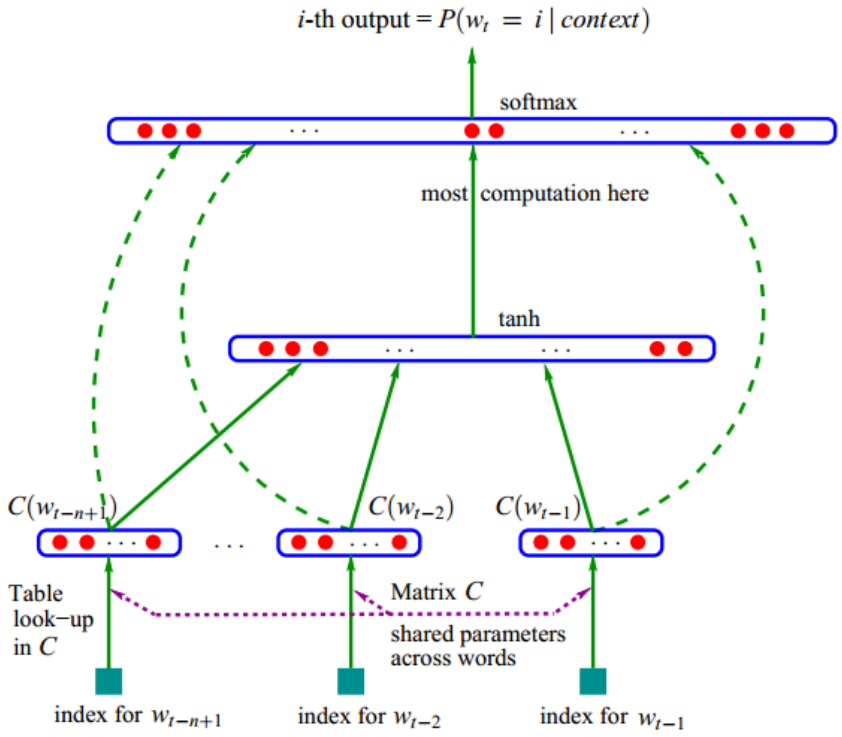


图 3.1

NNLM 可以计算一段上下文的下一个词为w_i的概率，即$p(w_i|context(w_i))$，NNLM 采用 Distributed Representation 来表示词向量，即每个词被表示为一个浮点向量，并且词向量是 NNLM 训练得到的副产物。

第 4 章 Word2vec 工作原理

4.1 Word2vec 介绍

Word2vec[21]是一个用于处理文本的双层神经网络，是 2013 年 Tomas

Mikolov 在 Google 带领研究团队创造。它的输入是文本语料，输出是一个词汇表，其中每个词都有一个对应的向量，即本文 2.3.2 节描述的 Distributed Representation 分布式词向量。

Word2vec 的目的和功用是在向量空间内将词的向量按相似性进行分组。它能够识别出数学上的相似性。Word2vec 能生成向量，以分布式的数值形式来表示词的上下文等特征。而这一过程无需人工干预

本文换个思路，将经过特征选择取出的词作为特征，通过使用 Word2vec 对特征词的处理、计算，可以应用到文本分类算法中去，具体算法将在第 5 章描述。

4.2 Word2vec 训练模型

Word2vec 是 Mikolov 等[17][18]所提出的神经网络语言模型的一个实现，可以用来快速训练词向量。Word2vec 内含两种训练模型，分别是连续词袋模型（Continuous Bag-of-Words Model，简称 CBOW）和 Skip-gram[19]模型，如下图所示：

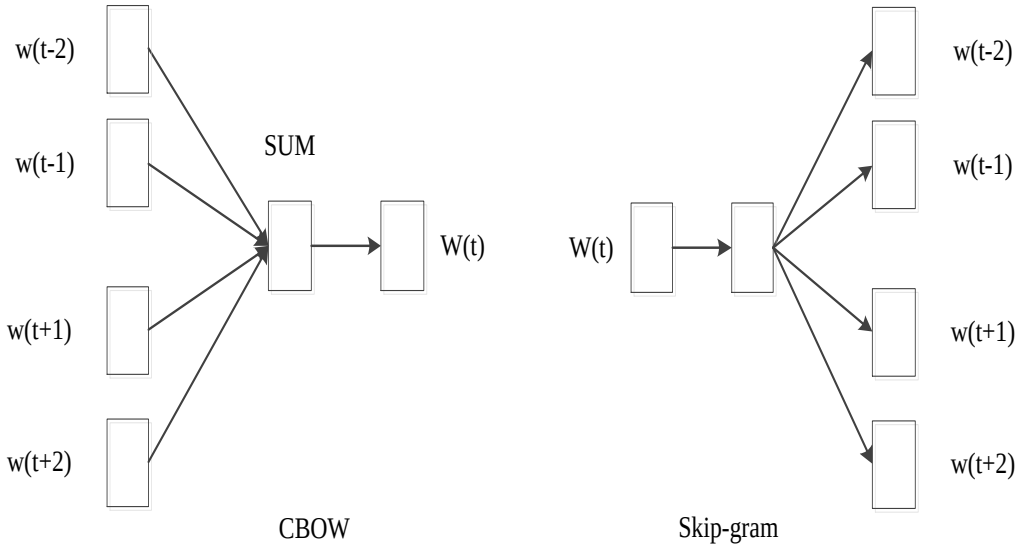


图 3.2 CBOW 和 Skip-gram 模型

从上图中可以看出，CBOW 和 Skip-gram 模型都包含输入层、隐含层和输出层。其中，CBOW 模式是通过上下文来预测当前词，而 Skip-gram 是使用当前词来预测其上下文。

Word2vec 中提供两种优化方法来提高训练效率，分别是 HierachySoftmax 和 Negative Sampling。将训练模型和优化方法进行组合可得到 4 种训练词向量的框架，如表 3.1 所示

表 3.1　Word2vec 词向量训练框架

模型	CBOW	Skip-gram
Hierachy Softmax	CBOW+HS	Skip_gram+HS
Negative Sampling	CBOW+NS	Skip_gram+NS

4.3　Word2vec 相关实验

第 5 章　基于词向量的分类算法的提出（重点自己叙述）

描述算法，给出算法框架图，必要时加上流程图（这里需要安装 visio，正在安装）

5.1　算法流程图

5.2　实验

5.3　算法可行性分析

第 6 章　基于词向量的算法实验探究（重点在实验结果分析）

6.1　实验数据准备（描述语料情况）

使用语言环境和机器配置等信息

6.2　实验结果

配图

准确率、召回率、F 值

6.3　分析结果

第 7 章　总结与展望

总结：本文研究了基于词向量的文本分类，提出了一种基于 k-means 聚类思想，以 tfidf 和词之间的距离为权**错误!未找到引用源。错误!未找到引用源。**重，关键字投票决定分类的方法——KWBDC。该算法的优点是一次训练词向量，以后分类直接加载调用即可，缺点是随着语料的增加，训练词向量所需时间也会相应增加，主题类别词汇需要人工归纳得出。通过实验证明该算法对中短文本分类的可行性，以及不同词向量输入、关键词个数对分类效果的影响，根据实验结果分析实验数据，得到相应的结论。

展望：实验所用数据集大小仅为 58M，所选分类类别数为 2 类、3 类，如果进一步丰富训练词向量的语料，则可以得到更高质量的词向量，并且对有望扩充至更多类别的分类。应用场景为快速新闻文本分类。

参考文献

[1] 搜狗全网新闻数据：https://www.sogou.com/labs/resource/ca.php

[2] 面向文本分类研究的中英文新闻分类语料：http://more.datatang.com/data/13484

[3] 周练. Word2vec 的工作原理及应用探究[J]. 科技情报开发与经济，2015（25）02：145-148

[4] Word2vec 官网：https://code.google.com/p/Word2vec/

[5] Word2vec 代码： http://Word2vec.googlecode.com/svn/trunk/

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.

[7] 《统计学习方法》李航

[8] 文本分类中互信息特征选择方法的研究

[9] 基于词向量的文本分类算法研究与改进_王明亚

[10] 基于词向量的短文本分类方法研究_江大鹏

[11] Robertson S. Understanding inverse document frequency: on theoretical arguments for IDF[J]. Journal of documentation, 2004, 60(5): 503-520.

[12] https://zh.wikipedia.org/wiki/TF-IDF

[13] Damage prediction for regular reinforced concrete buildings using the decision tree algorithm

[14] Weka 书籍

[15] Troussas C, Virvou M, Espinosa K J, et al. Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning[C]// Fourth International Conference on Information, Intelligence, Systems and Applications. IEEE, 2013:1-6.

[16] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. Journal of machine learning research, 2003, 3(Feb): 1137-1155.

[17] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[18] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in neural information processing systems. 2013: 3111-3119.

[19] Guthrie D, Allison B, Liu W, et al. A closer look at skip-gram modelling[C]//Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006). 2006: 1-4.

[20] 维基百科. 语言模型 [OL]. [2013-3-12]. http://zh.wikipedia.org/zh-cn/%E8%AA%9E%E8%A8%80%E6%A8%A1%E5%9E%8B

[21] Tomas Mikolov.Word2vec project [EB/OL].[2014－09－18].

https//code.google.com/p/Word2vec/.
[22]

10、致谢：

致谢（4 号宋体加粗居中），上下各空 1 行

致谢内容（小 4 号宋体）