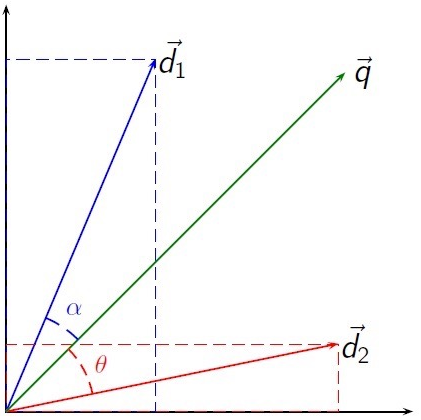
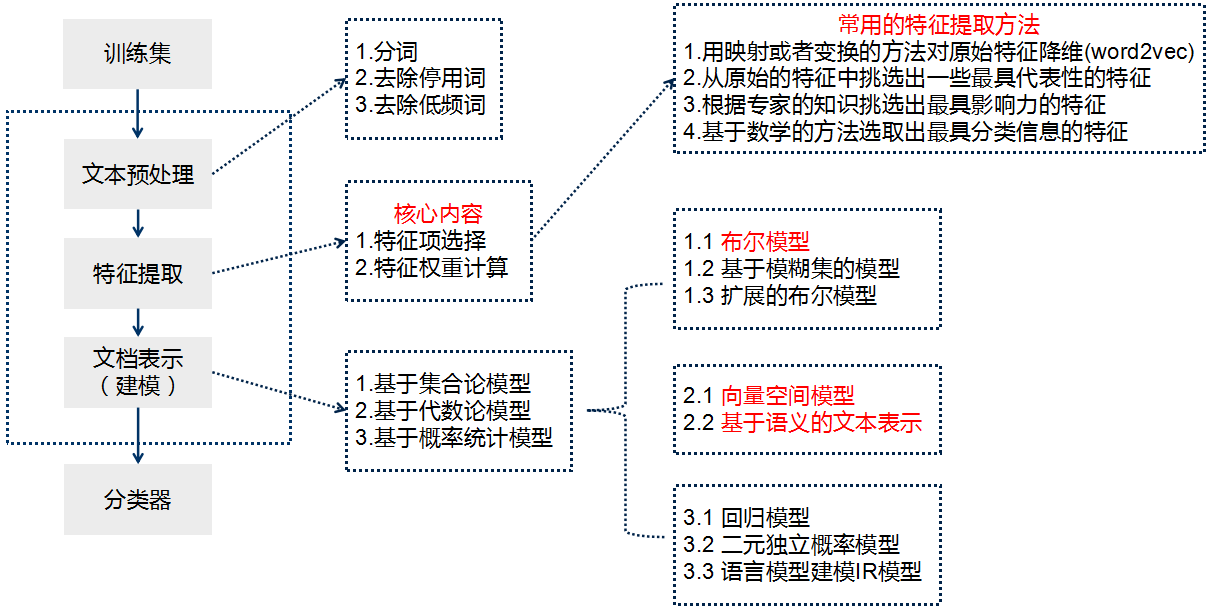
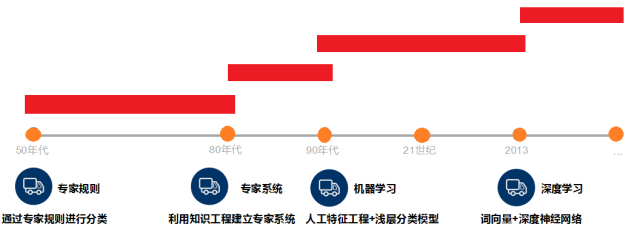
**文本分类特征提取之Word2Vec**

**相似度计算：**两个文档的相似程度可以用两向量的余弦夹角来进行度量，夹角越小证明相似度越高。文本分类就是根据文本内容将其分到合适的类别，它是自然语言处理的一个十分重要的问题。文本分类主要应用于信息检索，机器翻译，自动文摘，信息过滤，邮件分类等任务。

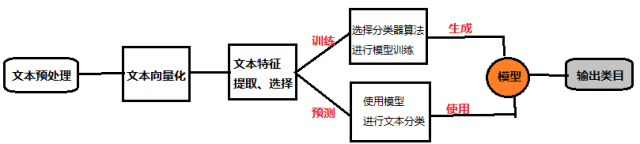


**文本分类技术发展历史**

640?wx_fmt=png&wxfrom=5&wx_lazy=1

* 1960-1970：那时主要通过**人工+规则**（关键词或者正则表达式）的方式，制定规则的人需要对某类目领域有足够的认知和了解。举个栗子：类目词是厨师，涉及到的规则可能是关于美食的、烹饪技巧的（刀工/炒锅/烘焙）或者厨师工种的（水台/砧板/打荷），需要涉及到方方面面，繁琐复杂。
* 1970-1990：出现了利用知识工程建立的**专家系统**，即，**信息检索概率模型VSM**（如向量空间模型Vector Space Model），以及相关评价指标如准确率、召回率的引入。
* 1990-2000：研究主要集中在了**文本特征（即词语）的提取、选择以及分类器模型的设计方面。**
* 2000年至今：多采用**词向量以及深度神经网络**来进行文本分类。

**有监督分类的主流程**

0?wx_fmt=png

* 文本预处理：分词，取出停用词，过滤低频词，编码归一化等；
* 文本向量化：如使用向量空间模型VSM或者概率统计模型对文本进行表示，使计算机能够理解计算，用的方法基于集合论模型、基于代数轮模型、基于频率统计模型等等；
* 文本特征提取和选择：特征提取对应着特征项的选择和特征权重的计算。是文本分类的核心内容，常用的特征提取方法：

1)用映射或者变换的方法对原始特征降维（word2vec）；

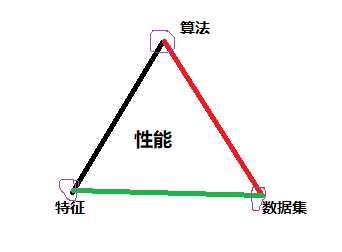
2)从原始的特征中挑选出一些最具代表性的特征；

3)根据专家的知识挑选出最具影响力的特征；

4)基于数学的方法选取出最具分类信息的特征。

* 分类器选择：回归模型，二元独立概率模型，语言模型建模IR模型。

在小编看来，不同的特征提取方法，会有自己的特点，用不同的分类的方法，效果也不一样，不能一概而论（遇到过数据集特征提取后，性能反而下降了，应该是数据集本身比较小的原因）。



分类的算法、文本的特征、所使用的数据集，我认为就像上图的三角关系一样，并不是每一个越长越好，正如可以表征文本的特征有很多，但并不是在构建特征越多，其分类效果越好。

**传统的向量空间模型（VSM）假设特征项之间相互独立，这与实际情况是不相符的。本文主要分享基于词向量的文本分类，什么是Word2Vec？如何有效的表征文本的？**

**Word2Vec**

0?wx_fmt=png

2013年，Google开源了一个用于生成词向量的工具，因其简单实用高效而引起广泛关注。

若有兴趣的读者，可阅读作者的原论文[8]。

Word2Vector本质上有两个学习任务，还有两套模型分别是：

**CBOW（Continuous Bag-Of-Words，即连续的词袋模型）**：对于每个词，用其周围的词，来预测该词生成的概率。

**Skip-Gram**：对于每个词，用其自身去预测周围其他词生成的概率。

这两套模型，**将词语映像到同一坐标系，输出为数值型向量的方法。**简而言之，就是将人类才可以看懂的文字，转换为机器也可以识别、操作、处理的数值，将一串文字转化为一个数值型向量的过程。

Word2vec的产生是一个必然的过程，随着人类对非结构化数据（文字、语音、图像等）分析的需求，尤其是大量文本类数据的分析，必然需要一些让计算机“理解”文字的方法，最直接有效的自然就是将文字转化为数字；换言之，**将全部的文本映射到数值空间**中就是word2vec做的事情。这里需要引入一个概念——**语言模型**，我们不会在此深入的去探索语言模型，只是简单向读者介绍这个概念。

语言最大的特征就是上下文的联系，比如中文当中，两个特定的字会组成一个词语，若干词语和字组成句子，多个句子组成文章，如果文章中的全部文字都随机的打乱顺序排列，那么我想哪怕正在读文章的你是中文系的高材生，要完全看懂这篇文章也要费九牛二虎之力，请看下面这两段文字：

“**自然语言处理是计算机科学领域与人工智能领域中的一个重要方向。它研究能实现人与计算机之间用自然语言进行有效通信的各种理论和方法。自然语言处理是一门融语言学、计算机科学、数学于一体的科学。因此，这一领域的研究将涉及自然语言，即人们日常使用的语言，所以它与语言学的研究有着密切的联系，但又有重要的区别。”**

**“与言将学处处方联及数言理语言信理各、语区工计系别这之和论切领所因人使中间究密的它有然的的与涉有自的要一言算语要效现体进的机融，语学究科科的计然行的学、但实又研是然。与重门能，究然常语域们语科是。自通人种日。言用领域机领域一智着机一人有即言用以言于能研方学的学它自重算向。算学自，计法此研个理一，语”**

好吧，看到这里你可能要骂人了，第二段是人话么？答案是，第二段话就是第一段话，只是对第一段话随机打乱了顺序，这个不倒150字的段落，也许你用30分钟一个小时可以将它还原成“人话”，如果是长篇大论的呢？这里就看出了语言的逻辑性，词语前后关系的重要性。语言模型就是在做这样的事情，考察一个句子出现的可能性（也就是概率）。如果一个句子S由n个词w1~wn，那么S出现的概率就应该等于P(w1,w2,…,wn)，用条件概率的公式即得到共识①如下： 

P(S)=P(w1,w2,…,wn)=P(w1)P(w2│w1)…P(wn|w1,w2,…,w(n−1))

不懂这个公式丝毫不影响后面的学习，这个公式翻译成白话就是：词语wn出现的概率依赖于它前面n−1个词。当n很大时，P(wn│w1,w2,…,w(n−1))的计算是非常麻烦甚至无法估算，于是产生了一个叫做**马尔科夫假设的概念**并由此得到“**二元模型**”。马尔科夫假设的意思是“**任意一个词w\_k只与它前面的词即w(k−1)有关**”。那么这样，公式①就可以写作下面的公式②的形式： 

P(S)=P(w1)P(w2│w1)…P(wn|w(n−1))

语言模型正是为了考察句子，或者说由词a到词b存在在句子中的概率而存在的。如果将我们的语言模型即为f，那么word2vec就是去训练这个f，不关注这个模型是否完美，而是要获取模型产生的参数，对于word2vec，就是神经网络的权重参数，将这些参数作为句子S的替代，由这些参数组成的向量，就是我们所谓的“词向量”。这也就是word2vec的输出。

**word2vec从大量文本语料中以无监督的方式学习语义信息，即通过一个嵌入空间使得语义上相似的单词在该空间内距离很近。**比如“机器”和“机械”意思很相近，而“机器”和“猴子”的意思相差就很远了，那么由word2vec构建的这个数值空间中，“机器”和“机械”的距离较“机器”和“猴子”的距离而言是要近很多的。

**附：Skip-gram 和 CBOW**

0?wx_fmt=png

不要被这两个看起来高大上的词吓到，其实很简单。由word2vec思想，即考察前后文的关联性产生了这两个模型：Skip-gram是通过一个词a去预测它周围的上下文；而CBOW相反，是通过上下文来预测其间的词。CBOW一般用于数据，而Skip-Gram通常用在数据量较大的情况。

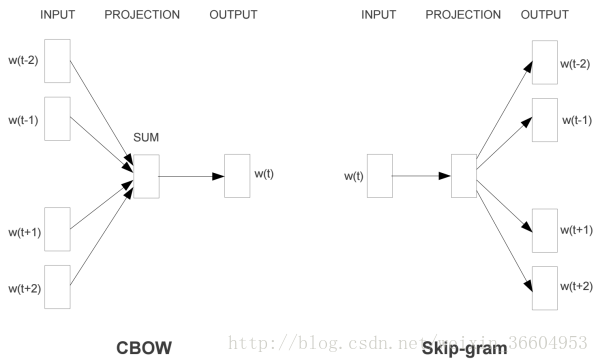


图1

图1比较直接的展示了比较常规的CBOW和Skip-gram，图1左侧的CBOW给出了文本中一个词语

前文说过，word2vec输出是神经网络的权重值，用这些值组成词向量，它的输入就是我们文本中的词（将文本按一定颗粒度切分为词语和字），但是输入进计算机的文字依然是无法被计算机识别啊？该怎么进入这个过程呢？答案就是——**one hot encoder**。one hot encoder就是一个只含有1个1，其他都为0的值构成的向量。如果全世界的词一共有N个，为了简单起见，假设N是3，这3个词分别是“我”、“是”、“帅哥”（别骂我不要脸~~~），那么很显然，“我”就可以被编码为（1，0，0），“是”编码为（0，1，0），“帅哥”编码为（0，0，1）。这样的数值向量是完全可以被计算机识别和处理的。

言归正传，我们先来看一看Skip-gram模型。我们的输入已经明确了，再明确一下我们的输出：Skip-gram是一个单隐藏层的神经网络结构，那么我们要找的词向量其实就是**输入层到第一个隐藏层的权重**（也就是图1中Project层的权重值）。

这里，引入“窗口”和“skip-num”的概念，举个例子，比如有这么一句话：“中秋月亮比往常圆”，“中秋”、“月亮”、“比”、“往常”、“圆”作为6个独立的词，如果我们的输入是“月亮”，窗口设为2，skip-num也是2，那么我们会得到两组输入-输出：（月亮，中秋）和（月亮，比），假如先使用（月亮，中秋）作为神经网络的输入和输出进行训练，那么神经网络会输出整个文本中每个词作为“月亮”的输出的概率，也就是给定这组输入输出数据时，输入是“月亮”，输出结果是“中秋”的概率，即文本中每个词与输入值“月亮”相邻出现的概率（当然，这个例子没用对文本进行one hot encoder，是为了方便解释和理解，读着可以自行把它们假想为0-1向量，不想也没关系，小编觉得这样看得见摸得着不抽象的例子才能让大家都看懂）。

现在假设我们的Skip-gram是一个最简单的情况，用一个输入词去预测它相邻的一个词，如图2和图3所示（图3是图2的展开、细化模式）。那么请接着看图4，如果我们的词表中有10000个词，那么每个词进行one-hot编码后，就会是一个10000维的向量，其中9999个都是0，只有一个是1。从输入层到隐藏层没有使用激活函数，但在输出层中使用了softmax（不用怕，这个东西很简单，后面会介绍的，目前你只要把它简单理解成为类似逻辑回归中的logstic变换即可）。

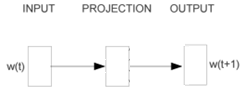


图2

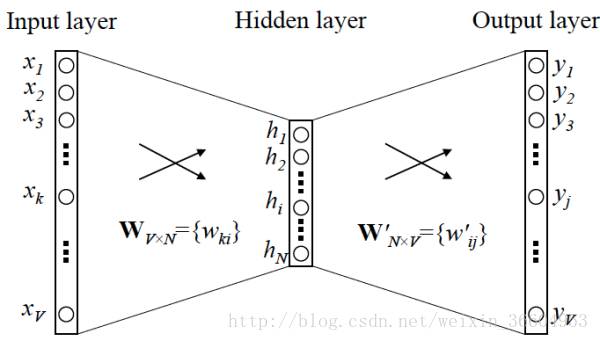
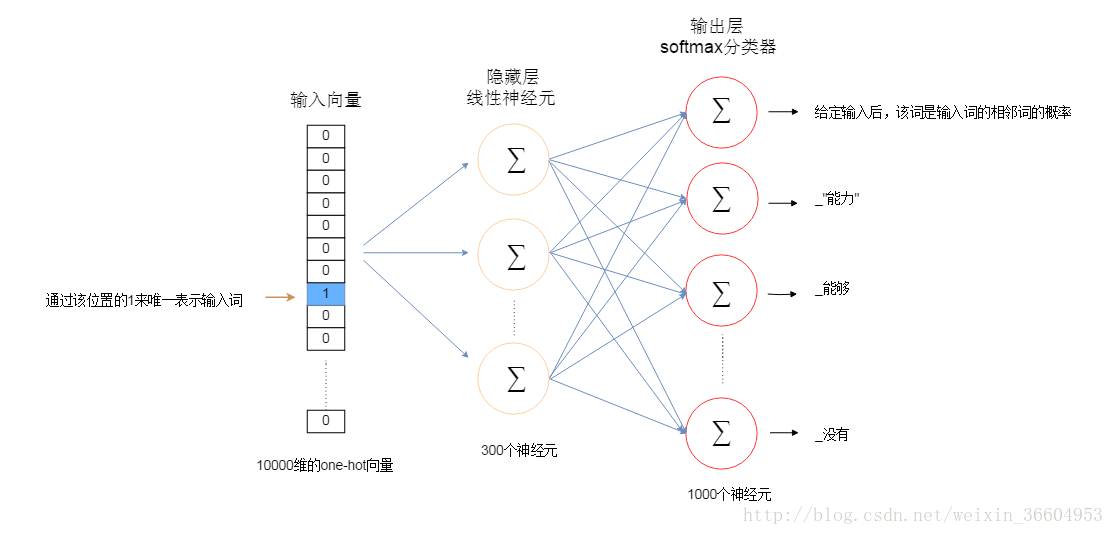
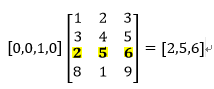


图3

图4

输出层理所当然也是10000维的向量，每个值就是按照输入的这个词，可以得到自己本身的概率，这个10000维的向量其实就是一个概率分布。

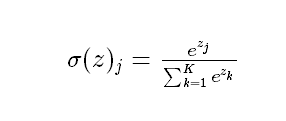
我们关注的点在隐藏层，如果我们想用将每个词表示为一个维度为100的向量，即每个词有100个特征，那么隐藏层就是一个10000行×100列的矩阵（也就是隐藏层有100个结点），这个10000×100的矩阵就是我们最后想要的结果，它会将10000个词中的每一个表示为一个100维的向量，有木有发现，10000维的词瞬间被降维到了100维，当然这个权重矩阵中的值就不是0-1了。输入是一个1×10000的向量，隐藏层是一个10000×100的矩阵，这两个矩阵做乘法运算看似需要耗费很多的计算资源，其实并没有，计算机根本没有在运算（也没有在偷懒哈），它只是做了一个“查表”的工作，也就是对权重矩阵做了一个“提取”的动作，只提识别1×10000的向量中那个不为0的值对应的索引，并且从10000×100的矩阵中提取该索引所对应的行。简单展示如下：



最终得到这个1×100的向量。在输出层通过softmax进行变换，使输出层每个结点将会输出一个0-1之间的值（概率），这些所有输出层结点的概率之和为1。

CBOW是同样的道理（如果按照最简单的情况，Skip-gram用一个输入词预测它后面的词，CBOW用一个输出词去预测它前面的一个词，其实是一模一样的套路）。

一点补充：理论部分的最后一点就是对前文的softmax进行一点简单的说明：softmax函数如下所示，



**总结**

文本分类的一般步骤，基本上都是在大量有数据集上做的有监督学习，需要大量人工标注数据（对于有些任务，标注数据获取成本极高）。从上面的分析看出，Word2Vec模型不需要标注数据，其能自发的学习出词与词之间的相似性和某些概念之间的内在联系。

## 导语

传统的向量空间模型（VSM）假设特征项之间相互独立，这与实际情况是不相符的，为了解决这个问题，可以采用文本的分布式表示方式(例如 word embedding形式)，通过文本的分布式表示，把文本表示成类似图像和语音的连续、稠密的数据。

这样我们就可以把深度学习方法迁移到文本分类领域了。基于词向量和卷积神经网络的文本分类方法不仅考虑了词语之间的相关性，而且还考虑了词语在文本中的相对位置，这无疑会提升在分类任务中的准确率。 经过实验，该方法在验证数据集上的F1-score值达到了0.9372，相对于原来业务中所采用的分类方法，有20%的提升。

## 1.业务背景描述

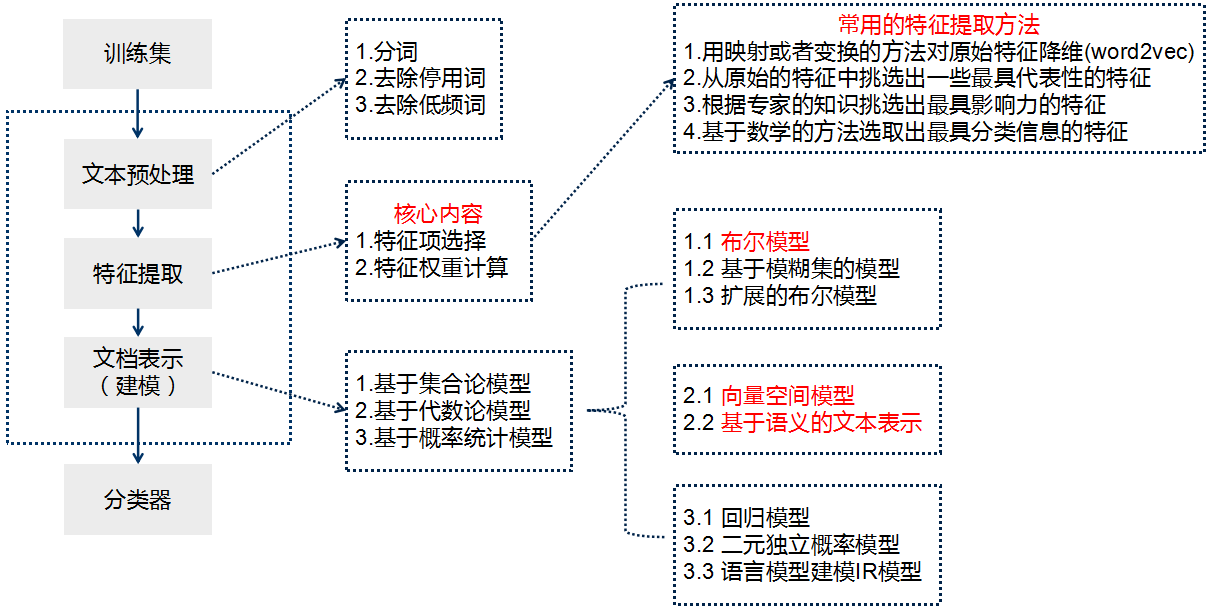
* 分类问题是人类所面临的一个非常重要且具有普遍意义的问题，我们生活中的很多问题归根到底都是分类问题。
* 文本分类就是根据文本内容将其分到合适的类别，它是自然语言处理的一个十分重要的问题。文本分类主要应用于信息检索，机器翻译，自动文摘，信息过滤，邮件分类等任务。

## 2.文本分类综述

### 2.1 文本分类的发展历史

* 文本分类最早可以追溯到上世纪50年代，那时主要通过专家定义规则来进行文本分类
* 80年代出现了利用知识工程建立的专家系统
* 90年代开始借助于机器学习方法，通过人工特征工程和浅层分类模型来进行文本分类。
* 现在多采用词向量以及深度神经网络来进行文本分类。

### 2.2 文本分类的流程



### 2.3 文档表示

如何把文档表示为算法能够处理的**结构化数据**无疑是文本分类非常重要的环节。

根据文本表示过程所使用的数学方法不同，可以分为以下几类：

1.基于集合论模型 a 布尔模型 b. 基于模糊集的模型 c.扩展的布尔模型

2.基于代数论模型 a 向量空间模型(VSM) b 基于语义的文本表示

3.基于概率统计模型 a 回归模型 b.二元独立概率模型 c. 语言模型建模IR模型

接下来会详细介绍一下布尔模型、向量空间模型(VSM)、基于语义的文本表示。

**文档相似度计算：**查询布尔表达式和所有文档的布尔表达式进行匹配，匹配成功得分为1，否则为0.

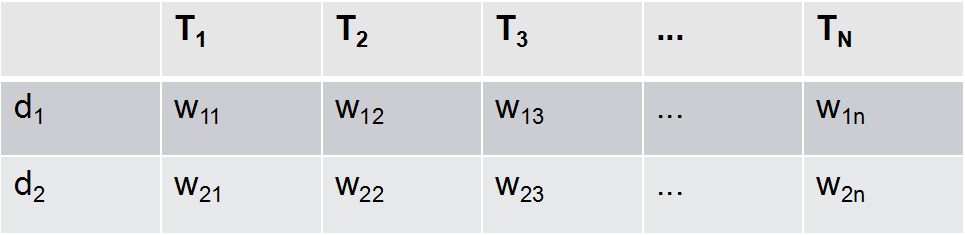
**布尔模型的优缺点：**

优点：简单、现代搜索引擎中依然包含了布尔模型的理念，例如谷歌、百度的高级搜索功能。

缺点：只能严格匹配，另外对于普通用户而言构建查询并不容易。

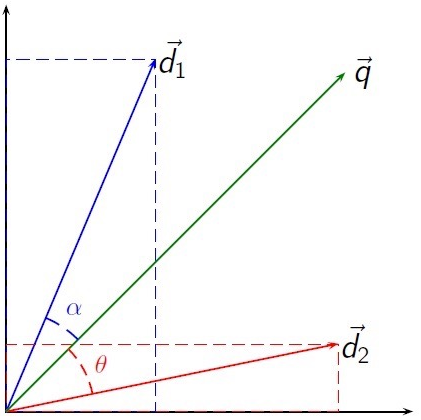
### 2.3.2 向量空间模型

**向量空间模型：**把对文本内容的处理简化为向量空间的向量计算。并且以空间上的相似度表达文档的相似度。



每篇文档由T1、T2、...、Tn一共N个特征项来表示，并且对应着Wi1、Wi2、... 、Win个权重。通过以上方式，每篇文章都表示成了一个N维的向量。

**相似度计算：**两个文档的相似程度可以用两向量的余弦夹角来进行度量，夹角越小证明相似度越高。



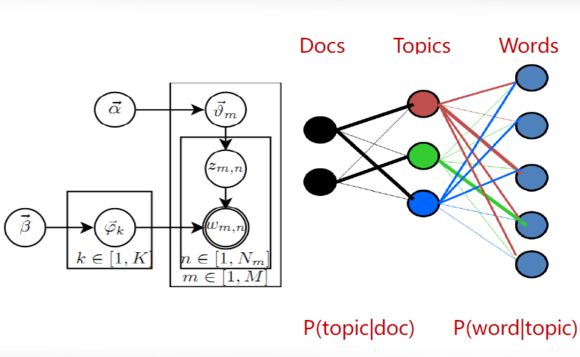
**优缺点：**

优点：1.简洁直观，可以应用到很多领域（文本分类、生物信息学等）2.支持部分匹配和近似匹配，结果可以排序 3. 检索效果不错

缺点：1.理论上支持不够，基于直觉的经验性公式。 2. 特征项之间相互独立的假设与实际不符。例如，VSM会假设小马哥和腾讯两个词语之间是相互独立的，这显然与实际不符。

### 2.3.3 基于语义的文本表示

**基于语义的文本表示方法：**为了解决VSM特征相互独立这一不符合实际的假设，有人提出了基于语义的文本表示方法，比如LDA主题模型，LSI/PLSI概率潜在语义索引等方法，一般认为这些方法得到的文本表示是文档的深层表示。而word embedding文**本分布式表示方法**则是深度学习方法的重要基础。

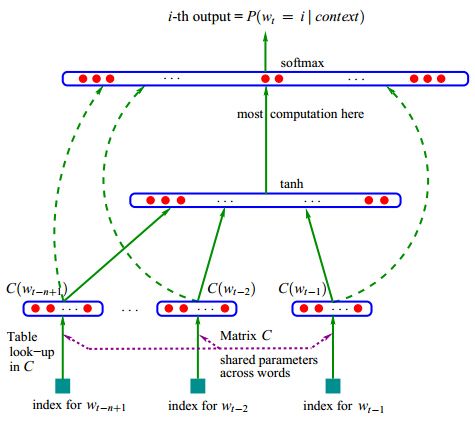


### 三2.3.4 文本的分布式表示：词向量(word embedding)

**文本的分布式表示**(Distributed Representation)的基本思想是将每个词表示为n维稠密，连续的实数向量。

分布式表示的最大**优点**在于它具有非常**强大的表征能力**，比如n维向量每维k个值，可以表征k的n次方个概念。

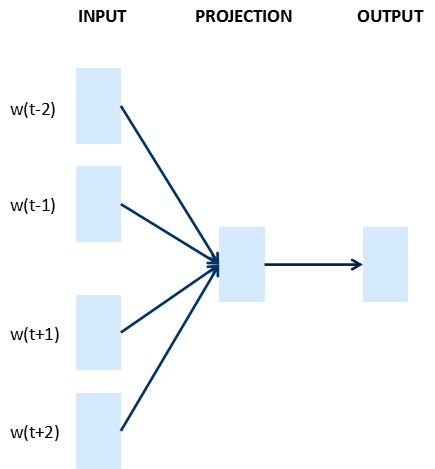
事实上，不管是神经网络的影层，还是多个潜在变量的概率主题模型，都是在应用分布式表示。下图的神经网络语言模型（NNLM）采用的就是文本分布式表示。而词向量(word embedding)是训练该语言模型的一个附加产物，即图中的Matrix C。



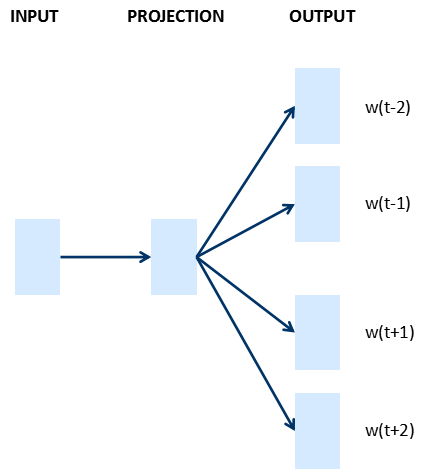
神经网络语言模型（NNLM）

尽管词的分布式表示在86年就提出来了，但真正火起来是13年google发表的两篇word2vec的paper,并随之发布了简单的word2vec工具包，并在语义维度上得到了很好的验证，极大的推动了文本分析的进程。

文本的表示通过词向量的表示方法，把文本数据从高纬度稀疏的神经网络难处理的方式，变成了类似图像、语言的连续稠密数据，这样我们就可以把深度学习的算法迁移到文本领域了。下图是google的词向量文章中涉及的两个模型CBOW和Skip-gram。



CBOW：上下文来预测当前词



Skip-gram：当前词预测上下文

## 2.4 特征提取

**特征提取**对应着**特征项的选择**和**特征权重的计算**。

**特征项的选择**就是指根据某个**评价指标**独立的对原始特征项(词语)进行**评分排序**，从中选取得分最高的一些特征项，**过滤**掉其余的特征项。

特征权重的计算：主要思路是依据一个词的重要程度与类别内的词频成正比(代表性)，与所有类别中出现的次数成反比(区分度)。

当选用数学方法进行特征提取时，决定文本特征提取效果的最主要因素是评估函数的质量。常见的评估函数主要有如下方法：

**2.4.1 TF-IDF**

TF：词频，计算该词描述文档内容的能力 IDF：逆向文档频率，用于计算该词区分文档的的能力

* 思想：一个词的重要程度与在类别内的词频成正比，与所有类别出现的次数成反比。
* 评价：a.TF-IDF的精度并不是特别高。b.TF-IDF并没有体现出单词的位置信息。

**2.4.2 词频(TF)**

词频是一个词在文档中出现的次数。通过词频进行特征选择就是将词频小于某一阈值的词删除。

* 思想：出现频次低的词对过滤的影响也比较小。
* 评价：有时频次低的词汇含有更多有效的信息，因此不宜大幅删减词汇。

**2.4.3 文档频次法(DF)**

它指的是在整个数据集中，有多少个文本包含这个单词。

* 思想：计算每个特征的文档频次，并根据阈值去除文档频次特别低(没有代表性)和特别高的特征(没有区分度)
* 评价：简单、计算量小、速度快、时间复杂度和文本数量成线性关系，非常适合超大规模文本数据集的特征选择。

**2.4.4 互信息方法(Mutual information)**

互信息用于衡量某个词与类别之间的统计独立关系，在过滤问题中用于度量特征对于主题的区分度。

* 思想：在某个特定类别出现频率高，在其他类别出现频率低的词汇与该类的互信息较大。
* 评价：优点-不需要对特征词和类别之间关系的性质做任何假设。缺点-得分非常容易受词边缘概率的影响。实验结果表明互信息分类效果通常比较差。

**2.4.5 期望交叉熵**

交叉熵反映了文本类别的概率分布和在出现了某个特定词的条件下文本类别的概率分布之间的距离 思想：特征词t 的交叉熵越大， 对文本类别分布的影响也越大。 评价：熵的特征选择不考虑单词未发生的情况，效果要优于信息增益。

**2.4.6 信息增益**

信息增益是信息论中的一个重要概念， 它表示了某一个特征项的存在与否对类别预测的影响。

* 思想：某个特征项的信息增益值越大， 贡献越大， 对分类也越重要。
* 评价：信息增益表现出的分类性能偏低，因为信息增益考虑了文本特征未发生的情况。 2.4.7 卡方校验

它指的是在整个数据集中，有多少个文本包含这个单词。

* 思想：在指定类别文本中出现频率高的词条与在其他类别文本中出现频率比较高的词条，对判定文档是否属于该类别都是很有帮助的.
* 评价：卡方校验特征选择算法的准确率、分类效果受训练集影响较小，结果稳定。对存在类别交叉现象的文本进行分类时，性能优于其他类别的分类方法。

**2.4.8 其他评估函数**

* 二次信息熵(QEMI)
* 文本证据权(The weight of Evidence for Text)
* 优势率(Odds Ratio)
* 遗传算法(Genetic Algorithm)
* 主成分分析(PCA)
* 模拟退火算法(Simulating Anneal)
* N-Gram算法

### 2.5 传统特征提取方法总结

传统的特征选择方法大多采用以上特征评估函数进行特征权重的计算。

但由于这些评估函数都是基于统计学原理的，因此一个缺点就是需要一个庞大的训练集，才能获得对分类起关键作用的特征，这需要消耗大量的人力和物力。

另外基于评估函数的特征提取方法建立在特征独立的假设基础上，但在实际中这个假设很难成立。

### 2.6 通过映射和变化来进行特征提取

特征选择也可以通过用映射或变换的方法把原始特征变换为较少的新特征 传统的特征提取降维方法，会损失部分文档信息，以DF为例，它会剔除低频词汇，而很多情况下这部分词汇可能包含较多信息，对于分类的重要性比较大。 如何解决传统特征提取方法的缺点：找到频率低词汇的相似高频词，例如：在介绍月亮的古诗中，玉兔和婵娟是低频词，我们可以用高频词月亮来代替，这无疑会提升分类系统对文本的理解深度。**词向量能够有效的表示词语之间的相似度。**

### 2.7 传统的文本分类方法。

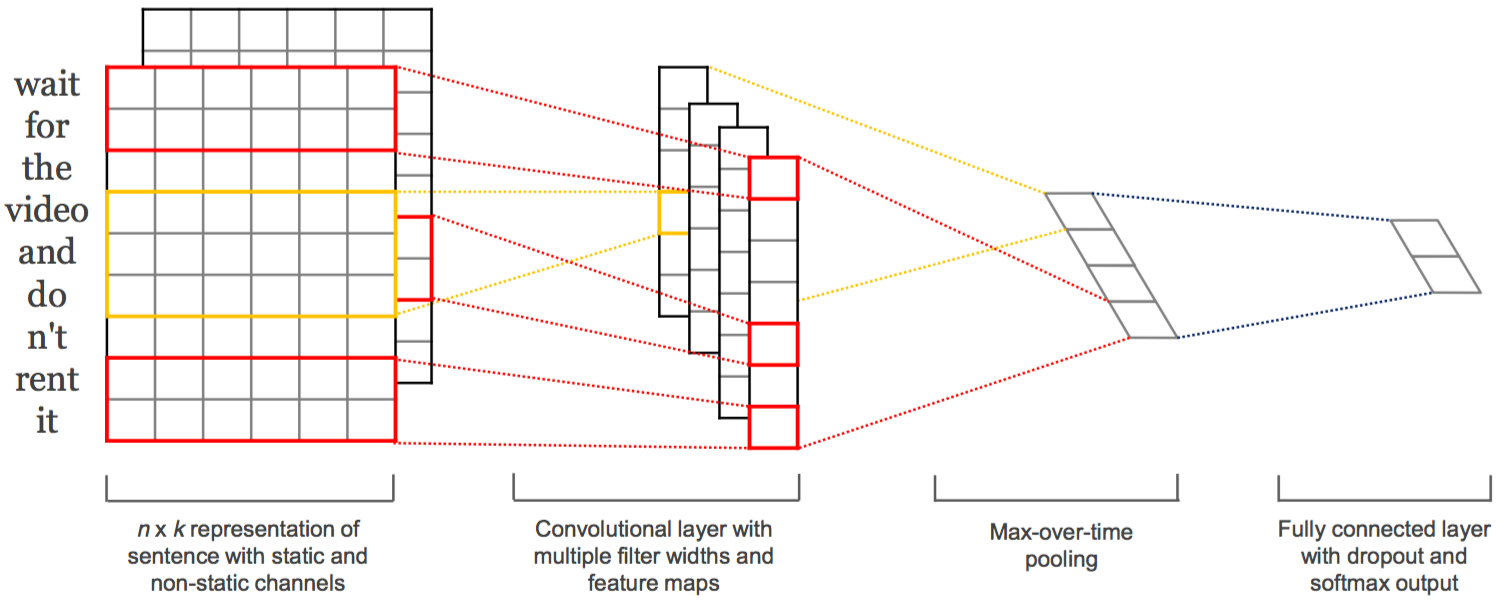
* 基本上大部分机器学习方法都在文本分类领域有所应用。
* 例如：Naive Bayes，KNN，SVM，集合类方法，最大熵，神经网络等等。

### 2.8 深度学习文本分类方法

* 卷积神经网络(TextCNN)
* 循环神经网络(TextRNN)
* TextRNN+Attention
* TextRCNN(TextRNN+CNN)

本文采用的是卷积神经网络(TextCNN)

## 3.实践及结果

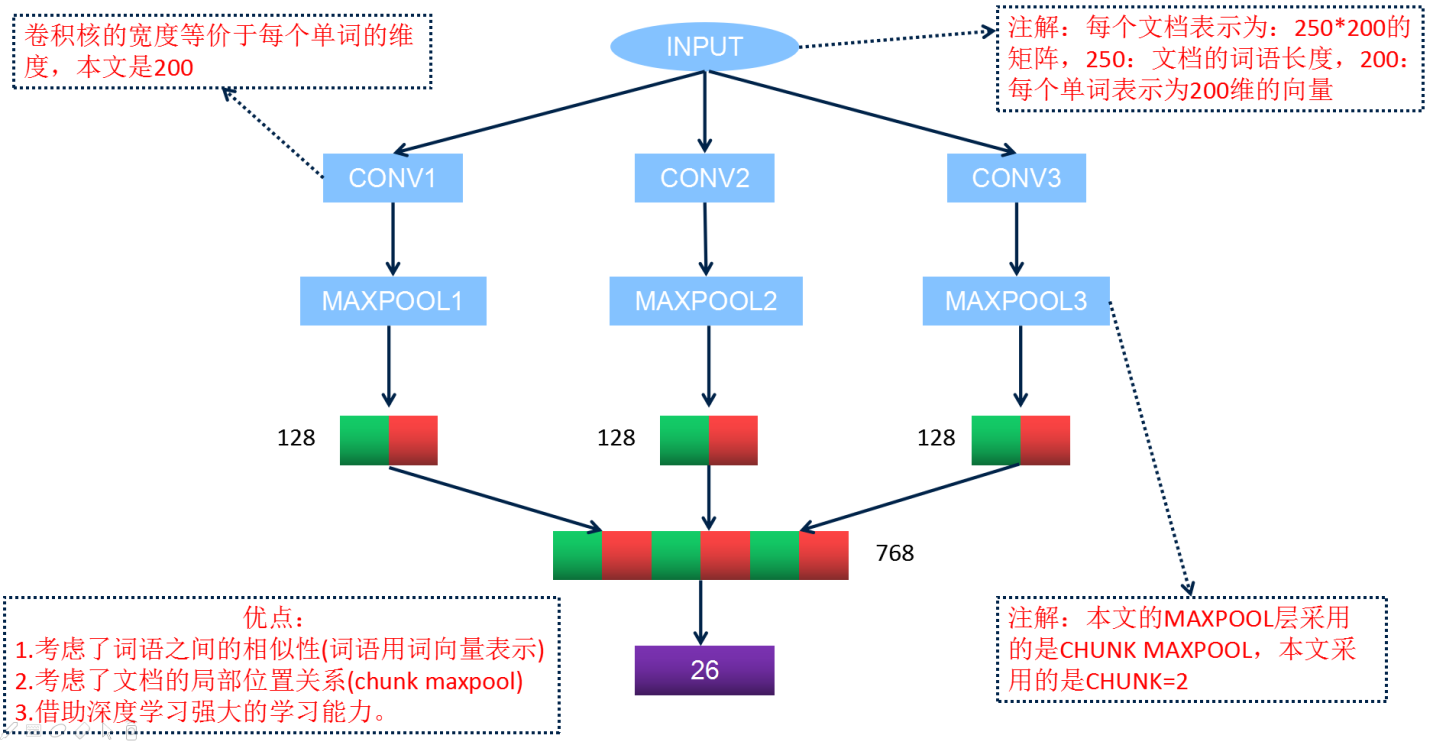


TextCNN网络概览图

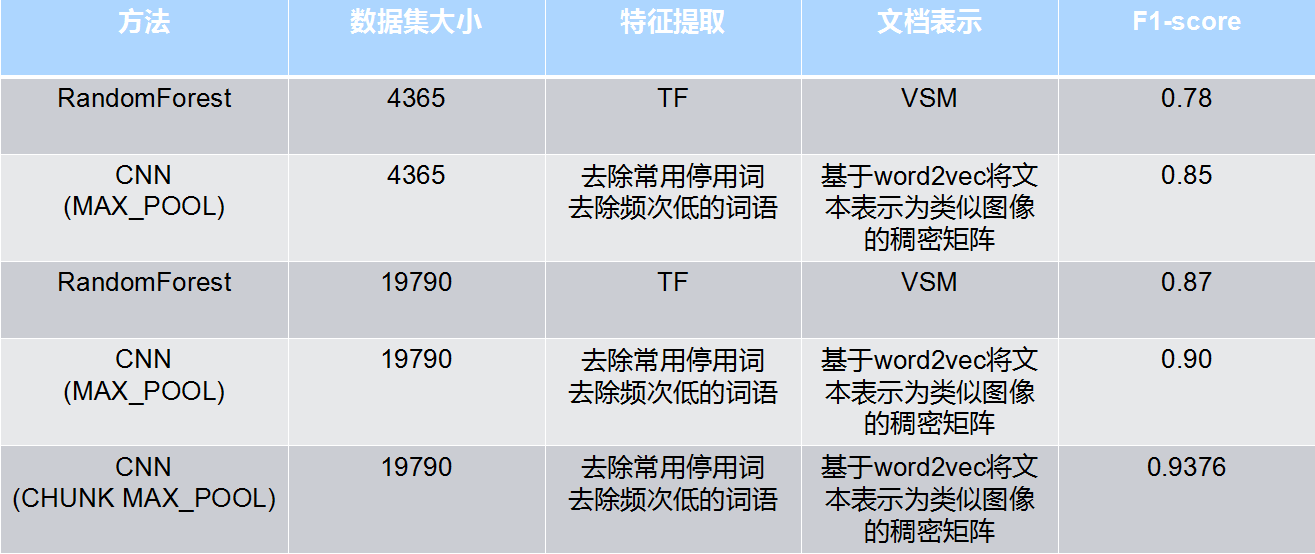
### 3.1 实践步骤

* 根据警情详情首先训练词向量模型，vector.model
* 把警情详情文本进行分词，去除停用词，然后利用词向量来表示，每篇文档表示为250\*200的矩阵(250:文档包含的词语个数，不够的以200维-5.0填充，200：每个词语用200维向量来表示)
* 把警情训练样本分割为train-set,validation set,test set。
* 利用设计好的卷积神经网络进行训练，并测试。

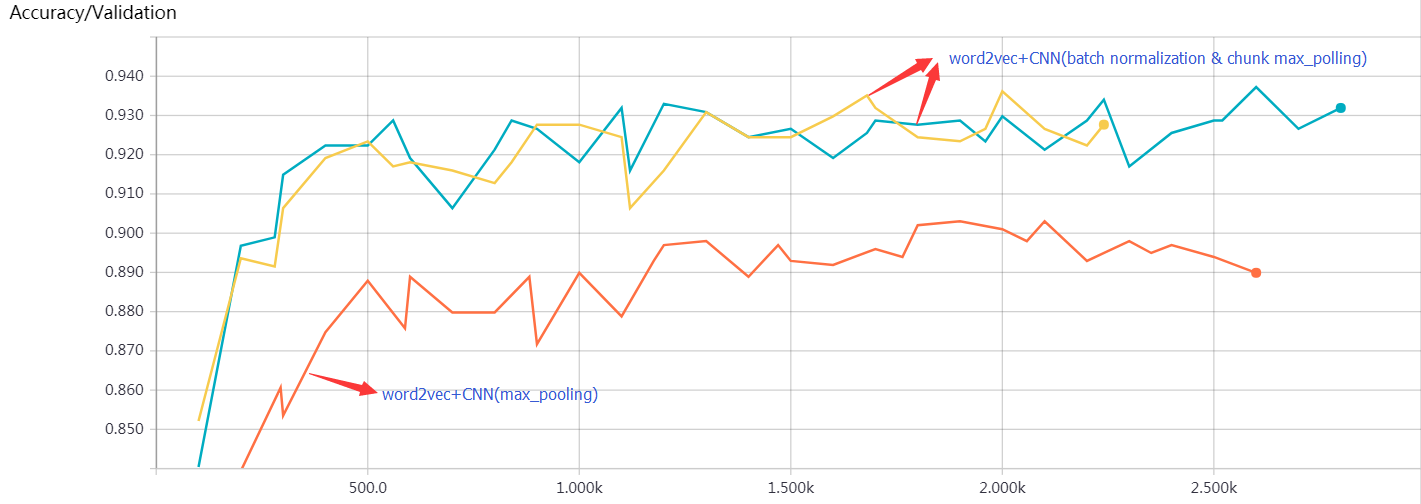
### 3.2 设计的卷积神经网络结构



### 3.3 实验结果



为了检验模型在真实数据上的分类准确率，我们又额外人工审核了1000条深圳地区的案情数据，相较于原来分类准确率的68%，提升到了现在的90%，说明我们的模型确实有效，相对于原来的模型有较大的提升。



* 红色：word2vec+CNN(max\_pooling)在验证集上的准确率走势图
* 黄色和蓝色：word2vec+CNN(batch normalization & chunk max\_pooling:2 chunk)在验证集上的准确率走势图