为短文本推荐合适的emoji — 基于上下词及语境的*CBOW*多层神经网络分类模型研究

张逸凯[1]

[1]南京大学, 学号: 171840708, 年级: 大二

Recommend the right emoji for suitable text — Research on CBOW Multi-layer Neural Network Classification Model Based on Upper and Lower Words and Context

ZHANG Yi-Kai[1]

[1](Department of Computer Science and Technology, Nanjing University, Nanjing, China)

Abstract Aiming at the feature sparsity and context dependence of short texts, this paper proposes a short text classification method based on *CBOW* multilayer neural network. Using the sentiment tendency of short text itself, this paper adopts TF-IDF-CF characterization method, on the other hand, associates the short text context with the *CBOW* model to optimize the emoji of the chat data. Experiments show that this hybrid model is higher in classification performance better than the traditional Bayesian or SVM classification model. In the Kaggle competition private leaderboard it achieved a higher accuracy of 0.17554.

Key words Short text classification *CBOW* model Improved weight TF-IDF-CF method Multi-layer neural network classification Data Mining

摘要 针对短文本的特征稀疏性和上下文依赖性等特点, 本文提出一种基于*CBOW*多层神经网络的短文本分类方法. 利用短文本本身的情感倾向, 一方面采用改进权值的TF-IDF-CF特征化方法, 另一方面关联短文本上下文语境运用*CBOW*模型对聊天数据进行最优emoji推荐. 实验表明这种混合模型在分类性能上比传统的贝叶斯或支持向量机分类模型高出很多, 在Kaggle竞赛private榜上取得0.17554的较高准确率.

关键词 短文本分类 *CBOW*模型 改进权值TF-IDF-CF方法 多层神经网络分类 数据挖掘

1 引言

在互联网高速发展的时代, 每天有数以亿计的信息流涌现, 海量的短文本信息中有许多关键的信息, 对于未标记的聊天信息语料库, 很难从中获取有价值的信息, 因此如何从短文本(聊天信息)中整合已有的标记样本, 并对未标记样本进行学习, 成为了短文本分类里一个至关重要的问题.

本文基于一个有趣的例子: "给短文本配上相应的emoji", 即给定训练集聊天记录以及每条聊天信息对应的emoji, 模型将给测试集中每条聊天信息推荐最合适的emoji. 给定一条聊天信息(短文本信息), 模型能给出这条信息最适配的emoji, 这可以很好地规约为一个短文本多分类问题.

本文具体叙述了一种基于改进权值的TF-IDF-CF特征化结合朴素贝叶斯的文本分类算法, 并将其和其他多分类算法例如支持向量机进行对比; 本文重点提出了一种嵌入特征的集成学习框架以及向量空间模型*CBOW*. 并构造全局平均池化等隐含层的全连接神经网络, 来获得较高的分类准确率.

2 具体方法

本节简述了在短文本分类中各方法的实现细节. 因为短文本推荐合适的emoji其实可以化归为一个短文本分类问题, 在下面的叙述中将以短文本分类为研究目标.

本文认为这一类的数据挖掘问题都可以化归为一下几个步骤:

1. 数据预处理. 旨在尽可能最大程度留下更多信息, 剔除干扰的离群的文本信息.
2. 特征化处理. 获取、处理和提取有意义的特征和属性, 数值化特征化文本数据.
3. 建模分析. 利用统计模型或机器学习模型等对数据集进行分类.

2.1 预处理

本文数据预处理采用人民日报 1947-2017, 知乎问答, 微博语料库, 对不符合要求的词进行剔除, 分词器选择了jieba接触, jieba分词使用了基于前缀词典实现高效的词图扫描, 生成句子中汉字所有可能生成词情况所构成的有向无环图(DAG), 再采用了动态规划查找最大概率路径, 找出基于词频的最大切分组合. 还过滤了无意义的停用词, 标点等.

本文选择保留在中文以外的外国语言, 以及聊天信息中带有意义的符号.

在优化部分本文还使用了类库snowNLP进行短文本的情感分析.

2.2 特征选择方法 TF-IDF

TFIDF 函数常用于特征项权值的计算, 是向量空间模型中经典的特征权值函数, 用术语频率乘逆文档频率来表示特征项的权值, 即:

$$TF \times IDF = TF \times \frac{1}{DF}$$

其中术语频率表示特征词出现的次数, 反映了特征相对于某个文档的重要程度. 特征项W在文档中出现的次数越多, 对于文档的类别贡献越大, 因而特征项越重要. 逆文档频率表示出现特征项W的文档次数的倒数. 某特征项的文档越多, 则该特征对于文档类别的贡献越小, 因而特征项越不重要.

本文将特征的类别信息引入函数, 对特征权值函数进行改造, 将其用于特征选择. 改进的TDF方法使用类内术语频率、类内逆文档频率和类外术语频率、类内逆文档频率来计算特征对文档类别的贡献大小. TDF 函数定义如下:

$$\text{Largest of}\left(\frac{tf_i \times idf_i}{tf_{\text{other}} \times idf_{\text{other}}}\right) - \text{Second largest of}\left(\frac{tf_i \times idf_i}{tf_{\text{other}} \times idf_{\text{other}}}\right)$$

其中tf_i 表示第 i 类词的频率, tf_{other} 表示其他类词的频率, idf_i 表示第i 类的逆文档频率, idf_{other} 表示其他类的逆文档频率

TF-IDF是一种统计方法, 用以评估特征度量. 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.

2.3 朴素贝叶斯分类器

基于概率的分类问题, 就是要求使得$P(Y|X)$最大的Y的取值. 设输入空间$X \in R^n$为n维向量的集合, X是定义在X上的随机变量, 输出空间为类标记集合$Y = \{y_1, ..., y_k\}$, $Y = \{y_1, ..., y_k\}$, Y是定义在输出空间Y上的随机变量, 训练数据集共有N个样本：

$$T = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$$

根据贝叶斯定理, 给定样本x的条件下, 其类别取y_k的概率为:

$$P(Y = y_k|X = \mathbf{x}) = \frac{P(X = \mathbf{x}|Y = y_k)\, P(Y = y_k)}{\sum_j P\left(X = \mathbf{x}|Y = y_j\right) P\left(Y = y_j\right)}$$

朴素贝叶斯法对条件概率分布做了独立性假设, 极大减少了参数数量. 朴素贝叶斯法假设样本的所有特征在给定所属类别的情况下相互独立, 即:

$$\begin{aligned} P(X=\mathbf{x}|Y=y_k) &= P\left(X^{(1)}=\mathbf{x}^{(1)},\ldots,X^{(n)}=\mathbf{x}^{(n)}|Y=y_k\right) \\ &= \prod_{j=1}^{n} P\left(X^{(j)}=\mathbf{x}^{(j)}|Y=y_k\right) \end{aligned}$$

其中, $x^{(j)}$为样本x的第j个特征. 因此, 基于朴素贝叶斯假设, 后验概率为:

$$P(Y=y_k|X=\mathbf{x}) = \frac{P(Y=y_k)\prod_{i=1}^{n} P\left(X^{(i)}=x^{(i)}|Y=y_k\right)}{\sum_j P\left(Y=y_j\right)\prod_{i=1}^{n} P\left(X^{(i)}=x^{(i)}|Y=y_j\right)}$$

朴素贝叶斯法学习样本的类条件概率, 属于生成模型. 朴素贝叶斯分类器将后验概率最大的类别作为样本的归属, 因此分类模型可表示为:

$$y = \underset{y_k}{\operatorname{argmax}} P(Y=y_k|X=\mathbf{x}) = \underset{y_k}{\operatorname{argmax}} \frac{P(Y=y_k)\prod_{i=1}^{n} P\left(X^{(i)}=x^{(i)}|Y=y_k\right)}{\sum_j P\left(Y=y_j\right)\prod_{i=1}^{n} P\left(X^{(i)}=x^{(i)}|Y=y_j\right)}$$

要判断测试集的数据是否属于某一类标记. 可以由训练集数据计算先验概率及类条件概率, 然后计算测试集数据属于各类别的概率, 最终确定测试集数据的类标记.

本文在研究初期使用TF-IDF + 朴素贝叶斯处理, 效果并不理想(提交后分数大致为0.14), 主要原因是短文本部分的词频分布不是很好的特征提取方法, 贝叶斯等传统分类器也不能处理"语境"下的分类任务. 具体在下面的章节将会详述.

2.4 词向量

在某种程度上, 词向量只是一个权向量. 在一个简单的1-of-N编码中, 向量中的每个元素都与词汇表中的一个单词相关联. 给定单词的编码仅仅是一个向量, 其中对应的元素被设置为1, 而所有其他元素都为零这样一个向量以某种抽象的方式来表示一个单词的"意义".

但是上面的编码方式太简单了, 除了判断相等, 无法在单词向量之间进行有意义的比较. 在word2vec中, 使用一个单词的分布式表示. 取一个几百维的向量. 每个单词都由这些元素之间的权重分布表示. 因此一个单词的表示不是向量中的一个元素和一个单词之间的一对一映射, 而是分布在向量中的所有元素上, 向量中的每个元素都有助于定义许多单词.

在一个大型语料库就有可能学习单词向量, 可以捕捉单词之间的关系. 学习得到的单词表示实际上以一种非常简单的方式捕获有意义的语法和语义规则. 具体地说, 这些规则是作为共享特定关系的单词对之间的常量向量偏移量来观察的.

这里本文尝试过最大的预训练中文词向量库(Chinese Word Vectors 中文词向量), 并融合了人民日报 1947-2017, 知乎问答, 微博语料库, 注意这里不是上文提到的预处理, 而是作为训练数据输入给分类器.

从上面对词向量的生成过程分析其实可以看出为什么单纯的word2vec并不能带来很好的效果了. 除了词向量不能很好地刻画词与词之间的相似性, 其实只是单一的词向量并不能很好的表达语境和情感在短文本中的作用. 虽然预训练词向量库中其实带有词向量语义之间的距离度量, 但是毕竟是短文本, 词量少可提取的特征本来就少. 在PCA降维之后就更不用说了.

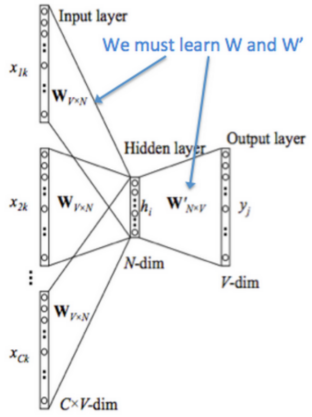
接下来将引入*CBOW*模型, 它是学习高质量分布式向量表示的一种有效方法, 捕获了大量精确的上下文和语义单词关系.

2.5 *CBOW*模型

为了获得较高的准确率, 本文还重点探索了*CBOW*(结合带有全局平均池化等隐含层的全连接神经网络)模型.

Continuous Bag of Words 模型, 其中每个短文本都表示为一个向量. 向量分量表示文档中每个单词的权重或重要性. 两个短文本之间的相似度是用余弦相似度度量来计算的. 使用大量文本创建单词的高维(本文选定EMBEDDING_DIM = 100)表示, 来捕获单词之间的关系.

因为本文的训练目标: 短文本(聊天记录) 是可以用多个词来代表一条的价值的,可以从周围词的语境来预测这个短文本的类标记的, 例如训练集数据180729:{ ”买了一盒不带海鲜的寿司, 味道好极了！ ” }, 可以用 { ”买”, ”一盒”, ”不带海鲜的”, ”寿司”, ”味道好极”}, 来作为特征化前的代表词. 需要更新神经网络的结构来适应这样的多词化输入:



其中 $X_ik...X_ck$ 表示多词化的输入; 因此每一个短文本都是1-out-of-v的表达方式. 即多个特

征化词向量的平均形式(输入向量经矩阵加权后的平均值), 这也可以说明为什么在隐含层中加入了全局平均池化层.

2.5.1 *CBOW*模型实现细节

定义

$$\mathcal{V} \in \mathbb{R}^{n\times|V|} \quad \mathcal{U} \in \mathbb{R}^{|V|\times n}$$

其中 n 是embedding空间的大小, $\mathcal{V}$ 是上文提到的输入embedding向量矩阵, $\mathcal{V}$的第i列w_i 是n维embedding向量. v_i表示这个$n\times 1$ 的向量 . $\mathcal{U}$是输出矩阵. $\mathcal{U}$的第j行是w_j的一个n维嵌入向量, 它是模型的一个输出.

更详细的, 我们定义隐含层输出$\mathbf{h}$

$$\mathbf{h} = \frac{1}{C}\mathbf{W}\cdot\left(\sum_{i=1}^{C}\mathbf{x_i}\right)$$

可以得到:

$$u_j = \mathbf{v}'^{T}_{\mathbf{w_j}}\cdot\mathbf{h}$$

其中$\mathbf{v}'_{\mathbf{w_j}}$ 是 $\mathbf{W}'$的 j^{th} 列.

以上详述了如何从输入到输出(正向传播)的.

模型的实现细节步骤如下:

1. 上文所叙述的输入层词向量矩阵: $\left(x^{(c-m)},\ldots,x^{(c-1)},x^{(c+1)},\ldots,x^{(c+m)}\right)$
2. embedding 向量矩阵: $\left(v_{c-m}=x^{(c-m)},\ldots,x^{(c-1)},x^{(c+1)},\ldots,x^{(c+m)}\right)$
3. 求均值: $\hat{O}=\frac{v_{c-m}+v_{c-m+1}+\ldots+v_{c+m}}{2m}$
4. 生成词频向量矩阵: $z=\mathcal{U}\hat{v}$
5. 转化成概率: $\hat{y}=\text{softmax}(z)$
6. 不断匹配优化这个概率矩阵

接下来需要学习$\mathcal{U}$ 和 $\mathcal{V}$, 可以借助交叉熵:

$$H(\hat{y},y) = -\sum_{j=1}^{|V|} y_j \log\left(\hat{y}_j\right)$$

这提供了一个很好的距离度量, 完美的预测(例如 $H(\hat{y},y)=-1\log(1)=0$不应受到惩罚, 而那些比较糟糕的预测(例如: $H(\hat{y},y)=-1\log(0.01)\approx 4.605$) 惩罚值也较大.

接下来学习权重矩阵, 从随机初始化的值开始. 然后训练实例依次输入后使用梯度下降等算法优化下面的损失函数(目标是在给定输入上下文的情况下最大化输出单词的条件概率), 由隐藏输出层权值的更新方程更新输入隐藏层权值的方程:

$$\begin{aligned}\text{minimize } J &= -\log P\left(w_c | w_{c-m}, \ldots, w_{c-1}, w_{c+1}, \ldots, w_{c+m}\right) \\ &= -\log P\left(u_c | \hat{v}\right) \\ &= -\log \frac{\exp\left(u_c^T \hat{v}\right)}{\sum_{j=1}^{|V|} \exp\left(u_j^T \hat{v}\right)} \\ &= -u_c^T \hat{v} + \log \sum_{j=1}^{|V|} \exp\left(u_j^T \hat{v}\right)\end{aligned}$$

如上, 本节详述了*CBOW*模型结合神经网络的实现原理及过程.

3 难点与方法选择动机

3.1 难点①: 短文本特征化

这是非常刺激的一个部分, coursera的课程: Advanced Machine Learning（高级机器学习）里面有讲很多技巧.

数据和特征决定了机器学习的上限, 而模型和算法只是逼近这个上限而已. 特征工程往往是打开数据密码的钥匙, 是数据科学中最有创造力的一部分.

在本文的前期一直沉浸与模型的训练, 没有考虑到上面那句话的内涵, 也因此耗费了很多时间, 可见特征化的重要性与难度.

难点在需要最大限度地从原始数据中提取特征以供算法和模型使用. 总的来说就是两步: 数据预处理→特征选择以及降维. 难点可细分为如下几类:

1. 信息冗余

   对于某些特征, 其包含的有效信息为关键表达词, 那些无意义的停用词可以被删除;

   离群点判定与删除, 有些过短的短文本可以作为离群点删除.

2. 特征选择

   使用不同的特征构造方法, 来从多个层面来判断这个特征的选择是否合适.

3. 定量化数据

机器学习算法和模型只能接受定量特征的输入, 将定性特征(文本)转换为定量特征. 比如emoji分类的类别需要one-hot encoding, 对于短文本数据, 本文尝试了TF-IDF, 以及词向量合成两种方式的处理, 并将其输入到特定的模型, 模型部分在下文中将详述. 还需要标准化归一化.

本文选择了改进TF-IDF权重作为特征提取方法, 下节将对这点进行详细叙述

4. 信息利用率低

   不同的机器学习算法和模型对数据中信息的利用是不同的, 需要选择合适的特征化方法, 比如词向量生成的短文本向量如果用传统机器学习方法贝叶斯等进行训练, 结果惨不忍睹. 反之亦然.

5. 维度灾难

   当特征选择完成后, 可能由于特征矩阵过大, 导致计算量大, 训练时间长的问题, 因此需要降低特征矩阵维度. 常见的降维方法除了以上提到的基于L1惩罚项(loss = 'l1')的模型以外, 还有主成分分析法（PCA）和线性判别分析（LDA）(Assignment *I*), 线性判别分析本身也是一个分类模型. PCA和LDA有很多的相似点, 其本质是要将原始的样本映射到维度更低的样本空间中, 但是PCA和LDA的映射目标不一样：PCA是为了让映射后的样本具有最大的发散性；而LDA是为了让映射后的样本有最好的分类性能. PCA是一种无监督的降维方法, 而LDA是一种有监督的降维方法.

6. 多分类

   分类的emoji表情共有多达72个表情符号. 不能只在那些最频繁的讨论, 因为评估分数是平均F_1 score, 这意味着所有的emoji分类都同等重要.

   其实就是精度问题, 如何提高多分类的精度.

在引出*选择改进权值的TF-IDF方法动机*之前, **本文还要重点详述短文本之间或者词之间相似性度量的难点**:

- 关联性和相似性, 比如*咖啡奶茶*和*马克杯*, 在主观上可以认为这是相关的, 但是完全不是相似的, 因为*咖啡奶茶*是液体, 可以喝的饮料, *马克杯*是固体, 通常用来装咖啡奶茶的杯子. 如何考虑 *咖啡*和*奶茶*之间的度量, 以及*咖啡*和*马克杯*之间的度量? 需要合理区分相似性和关联性的概念.

- 提出了关联性和相似性之后(两者截然不同), 本文继续提出更有挑战的部分: 语义相似和形态相似, 模型可能认为*店小二*和*小三*这两个词是相似的, 因此, 根据单词相似性来评估这种特定于任务的单词嵌入可能会不公平地惩罚它们. 特征化需要捕捉合适的单词相似性.

3.1.1 改进权值的TF-IDF方法选择动机

综合前节对简单版TF-IDF的概述, 其实在这过程中, 还是需要考虑不同部分的权重. 短文本中心部分的权重比结尾部分权重更大, 用计算得到的权重之和除于文档中的词语总数, 得到TF值.

一个词出现得越多, 权重就越小, 定义如下:

$$a_{ij} = tf_{ij} \times \log\left(\frac{N}{n_j}\right)$$

tf_{ij}表示文档i中短文本j的词频, N表示数据集中文档的总数, n_j表示文档i出现的数量.

当$N = nj$时, a_{ij}变为0, 这经常出现在小数据集中, 因此需要运用一些平滑技术改进:

$$a_{ij} = \log\left(tf_{ij} + 1.0\right) \times \log\left(\frac{N + 1.0}{n_j}\right)$$

但是有时候用词频来衡量文章中的一个词的重要性不够全面, 为了找到更好的特征化方法, 本文查阅了许多国内外文献, 这里采用了*An improvement of TFIDF weighting in text categorization*这篇文章上的方法: *TF-IDF-CF*用一个新的参数来表示类内特征, 并将其称为类频, 它计算一个类内文档中的词频.

下面给出引用文献中的定义:

$$a_{ij} = \log\left(tf_{ij+1.0}\right) * \log\left(\frac{N + 1.0}{n_j}\right) * \frac{n_{cij}}{N_{ci}}$$

并对其进行归一化处理, 来适应不同长度的短文本, 还有标准化处理:

$$a_{ij} = \frac{\log\left(tf_{ij} + 1.0\right) * \log\left(\frac{N}{n_j}\right)}{\sqrt{\sum_{p=1}^{M}\left[\log\left(tf_{ip} + 1.0\right) * \log\left(\frac{N}{n_p}\right)\right]^2}}$$

在此文献中使用x平方分布统计方法选择1000个特征, 进行了不同特征提取方法的对比测试:

Weighting Method	Naïve Bayes		Bayes Network		KNN		SVM	
	Reuters	20news	Reuters	20news	Reuters	20news	Reuters	20news
TFC	67.1%	62.3%	72.2%	67.8%	70.7%	58.9%	81.3%	63.8%
LTC	62.9%	61.8%	74.7%	63.4%	71.2%	53.1%	83.1%	67.2%
TF-IDF	61.6%	61.9%	76.9%	65.3%	72.8%	55.3%	84.7%	69.1%
TF-IDF-CF	88.6%	77.1%	91.4%	77.7%	81.4%	64.9%	92.8%	78.7%

改进权值的TF-IDF-CF方法显著提高了精度.

如上, 本文有充分理由采用TF-IDF-CF来进行特征提取.

中华文明博大精深, 对于中文这样丰富的语言(语法格式结构上的丰富), 需要很有耐心去处理特征.

3.2 难点②: 发现与创造优秀的模型

本文在研究初期尝试了基于主题和情感的多种贝叶斯分类, 但是由于短文本内容较短, 蕴含的信息有时太少, 而且特征稀疏, 难以计算文本之间的相似度, 导致分类的效果并不是很好.

支持向量机(Support Vector Machine, SVM)模型的效果略好于贝叶斯模型, 但是由于短文本内容较短, 对于支持向量机模型特征的稀疏性会被放大, 同时汉语中一义多词的现象导致短文本分类的召回率较低(TP较低). 又因为当训练文本集较大, 特征空间的维数将达到几万维甚至更高. 特征空间的高维性和数据的稀疏性将严重影响短文本分类的效果.

首先需要定义短文本之间的距离度量基础, 即相似度计算, 本文创新地提出了综合短文本中的上下文语境的词间相似度计算以及短文本表达的情感度来考虑短文本之间的距离.

以上考虑是基于汉语特点和数据基础的, 比如下面这个例子: "train.data" 中 181941 { 心情不好, 不过看到了哥哥们的帅照以后, 嘿嘿 }, 训练集标记结果为emoji: {色}, 但是如果只是基于词结构的特征化模型, {心情不好} 这个词结构会占很大的权重, 并把分类结果引向我们不需要的方向, 虽然文本短, 但是还是存在一些转折语境需要根据文本上下文解析的语境.

3.2.1 *CBOW*模型选择动机

由前节详述的*CBOW*模型, 注意到在输入层上下文词向量也参与计算, 在预测中心标记之前, 对上下文词中的向量求平均. 包含了嵌入向量的平均. 对于短文本的一些少见词(较少出现, 但并不生僻), 少见词的向量不与其他上下文词平均时, 这可以更好地表示罕见词.

在选择模型时, 本文还参考了许多国内外论文, 这里以*The Effects of Data Size and Frequency Range on Distributional Semantic Models* 为例:

这篇参考论文研究了数据大小和频率范围对分布语义模型的影响, 引用的测试结果如下图:

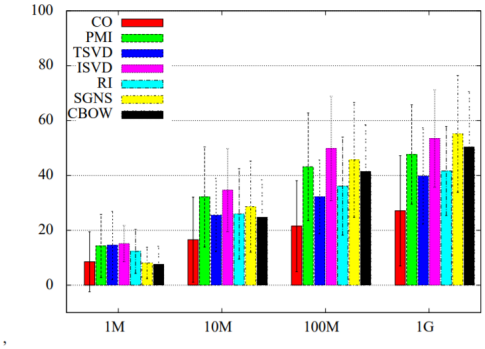


Figure 1: Average results and standard deviation over all tests.

在这篇参考论文中, 为了研究每个模型如何处理不同的词频率范围, 测试项被分成三个不同的类, 每个类包含测试项频率质量的大约三分之一, 数据测试显示了神经网络模型*CBOW*分别在高、低范围内产生了非常好的结果. *CBOW*模型对于低频率嵌入的词模型给出了一个适度的改进使得它的效果更好.

None of the tested models perform optimally for low-frequent items. The best results for low-frequent test items in our experiments were produced using the CBOW model, the PPMI model and the RI model, all of which uses weighted context

图 1: 参考论文辅助模型选择部分

因为train.data以及test.data数据特征的稀疏性(聊天记录), 综合上文提到的*CBOW*模型特性, 主题广少见词范围大. 本文有理由相信*CBOW*模型将会有优秀的表现.

如上, 本节详述了选择*CBOW*模型的原因.

本节是在基于已经研究了多种贝叶斯模型, 支持向量机模型等传统数据挖掘分类模型基础上的. 这些方法考虑的特征属性太少, 或者说不能很好利用所给的特征属性, 无法考虑语境在短文本上的作用, 不能得到很好的效果.

3.3 难点③: 构建神经网络训练模型

本文采用的卷积神经网络(CNN)结构像单层卷积块和平均池层, 然后是多个紧密连接的层. 最后全连接密集层有一个softmax激活函数和每个潜在对象类别的节点.

下面展示一个简要版本:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 300)               0
_________________________________________________________________
embedding_1 (Embedding)      (None, 300, 100)          2000000
_________________________________________________________________
global_average_pooling1d_1 ( (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 72)                7272
=================================================================
Total params: 2,007,272
Trainable params: 2,007,272
Non-trainable params: 0
_________________________________________________________________
```

本文选择了Global Average Pooling Layers对embedding向量进行均值化.

考虑到全连接的神经网络易于过度拟合, 妨碍了整个网络的泛化能力. 本文不仅使用了失活一部分神经元的方法, 还考虑了平均池化层, 这提供了一种向下采样特征的方法, 降低特征的检测样本, 总结了一个特性的平均存在性.

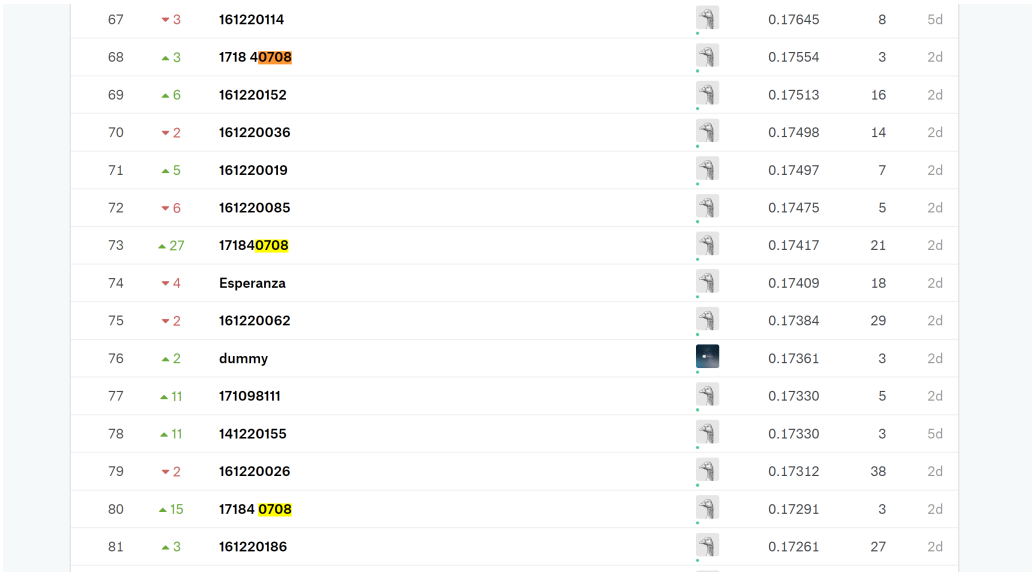
平均池化层可以较低分辨的地方, 采用其仍包含大的或重要的结构元素, 而没有可能对任务无用的精细细节. 这里做的是平均合并：计算短文本上每个词embedding向量的平均值. 这有助于使表示形式对输入的小平移保持近似不变. 平移的不变性意味着, 如果我们对输入进行少量平移, 那么大多数合并输出的值都不会改变.

神经网络结构复杂多样, 调整参数同样极具挑战.

4 总结

本文详述了基于改进权值的TF-IDF-CF特征化方法构建*CBOW*模型的原理与实现. 阐明了特征化方法以及模型选择的动机, 对本次实验的难点进行了深刻剖析.

功夫不负有心人, 回顾本次大作业的历程真是一把辛酸泪, 自学了很多很多很多, 很辛苦也很开心, 作者虽然是大二外院系小菜鸡, 但是还是拿到了排名前20%这样还不错的好成绩(学号*171840708*):



67	▼3	161220114		0.17645	8	5d
68	▲3	1718 40708		0.17554	3	2d
69	▲6	161220152		0.17513	16	2d
70	▼2	161220036		0.17498	14	2d
71	▲5	161220019		0.17497	7	2d
72	▼6	161220085		0.17475	5	2d
73	▲27	171840708		0.17417	21	2d
74	▼4	Esperanza		0.17409	18	2d
75	▼2	161220062		0.17384	29	2d
76	▲2	dummy		0.17361	3	2d
77	▲11	171098111		0.17330	5	2d
78	▲11	141220155		0.17330	3	5d
79	▼2	161220026		0.17312	38	2d
80	▲15	17184 0708		0.17291	3	2d
81	▲3	161220186		0.17261	27	2d

图 2: private榜

谢谢助教哥的耐心批改~ {心}

5 说明点及复现代码说明

本文使用了LaTeX排版, 可能和论文模板要求有微小出入, 请谅解.

复现结果运行 Loading_model_code.ipynb 即可

对于除了 Loading_model_code.ipynb 的其他源代码文件, 因为方便代码文件移动, 可能有部分代码中数据文件采用了绝对路径, 麻烦及时改为本机路径. 给您带来的不便, 我深表歉意.

请留意notebook分cell运行的特性.

所有代码保留了作者执行时的输出, 尽量提供了最多信息给助教哥~

参考文献

1 **Magnus Sahlgren, Alessandro Lenci,** The Effects of Data Size and Frequency Range on Distributional Semantic Models

2 **R Feldman, J Sanger,** The text mining handbook: advanced approaches in analyzing unstructured data

3 **Mo Yu, Mark Dredze,** Improving Lexical Embeddings with Semantic Knowledge

4 **Qun Luo, Weiran Xu, Jun Guo,** A Study on the CBOW Model's Overfitting and Stability

5 **Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean**, Efficient Estimation of Word Representations in Vector Space

6 **张群, 王红军,** 词向量与LDA相融合的短文本分类方法

7 **JC Martineau, T Finin**, Delta tfidf: An improved feature space for sentiment analysis

8 **杜慧, 徐学可, 伍大勇, 刘悦, 余智华**, 基于情感词向量的微博情感分类

9 **李国臣**, 文本分类中基于对数似然比测试的特征词选择方法

10 **LP Jing, HK Huang, HB Shi**, Improved feature selection approach TFIDF in text mining

11 **P Soucy, GW Mineau** , Beyond TFIDF weighting for text categorization in the vector space model

12 **费洪晓, 康松林, 朱小娟**, 基于词频统计的中文分词的研究

13 **J Ramos**, Using tf-idf to determine word relevance in document queries

14 **Marwa Naili, Anja Habacha Chaibi,** Comparative study of word embedding methods in topic segmentation

目录