1. **答案选择排序模型输入端的词向量计算方法**

3.1 研究动机

3.1 基于词语的CBOW词向量模型

根据第二章的介绍可知，CBOW模型是基于上下文来预测目标词，以词语为基本单位，选取目标词前后个特征词作为预测的上下文，假设给定一个滑动窗口内的上下文词序列，CBOW的目标就是最大化平均对数似然概率，



其中是目标词的上下文窗口大小，的计算使用（3-2）所示的softmax回归函数，

,

其中是词汇表，是目标词语 的词向量，是所有上下文词语词向量的平均值，



从以上描述可知，CBOW模型是以词语为单位，通过词与词之间的上下文来预测目标词语的，具体来讲，如下图所示，以{依法，缴纳，罚金}这组连续的词语序列为例，目标词语“缴纳”的词向量是根据其滑动窗口中上下文词语“依法”和“罚金”对其的预测过程来进行训练的；上下文词语的词向量是词语“依法”和“罚金”的词向量的求和平均值。



图1：基于词语的CBOW模型

3.2基于字和词语联合训练的CBOW词向量模型

大多数词向量生成方法都是将词作为基本单元，并且都是通过词与词之间的上下文来训练词向量，而忽略词的内部结构，然而，对于像中文这样的语言，一个词语通常包含几个字，并且这些字拥有很丰富的内部信息，一个词语的语义通常和组成它的这些字的字义息息相关；因此，将字的信息考虑进词向量计算过程中对词向量质量的提高是相当有必要的，本文以中文为例，将字和词联合训练的方法应用于CBOW词向量模型中。

3.2.1 基于字和词语联合训练的CBOW词向量模型框架

以词语“罚金”为例，这个词的语义信息可以从语料库中它的上下文训练得到，但同时，它的语义信息也可以从组成它的字“罚”和“金”的语义信息中推断得到，由于汉语言中存在的这种语义可组合的本质特性，内部字的语义信息在词语语义信息的建模中扮演着重要的角色，而这种将字的信息考虑进词向量训练的过程是可以建立在大部分词向量模型上的，根据第二章中着重介绍的几种word embedding模型，可以看出CBOW是一种训练效果好，训练速度相对较快的词向量模型，所以，本文是在CBOW模型之上，使用字和词语联合训练的方法，对其进行改进；

下图为基于字词联合训练的CBOW模型框架，和图1比较，此方法在计算向量时，不仅考虑上下文词语“依法”、“罚金”的词向量， 还考虑了组成这些词语的字的向量，它们共同决定了预测目标词的词向量。



图2：基于字词联合训练的CBOW模型

3.2.2 基于字和词语联合训练的词向量计算方法

字和词语联合训练的词向量计算方法认为，字向量对提升词向量的质量至关重要，假设，将汉字集合表示为，汉语词汇表表示为，每一个字，并且用表示其字向量，每一个词语，并且用表示其词向量，根据上节中的介绍，对于词序列，利用公式（1）最大化其平均对数似然概率，而在基于字和词语联合训练的词向量计算方法中，使用上下文词语的字向量和词向量来预测目标词，上下文词语的向量被表示为：



表示中词语的向量，是中字的个数，是中第个字的字向量，是复合操作，针对操作，有两种方案，加操作和连结操作，对于加操作而言，需要词语向量和字向量的维数相等，即，将词语向量和字向量的平均值进行简单的相加得到；另一种方法是，将词向量和字向量的平均值在维度上进行连结，即，这种情况下，字向量和词向量的维数不必完全相同，在本文的研究中，考虑到连操作的时间复杂度会高于相加操作，并且不一定会有明显优于相加操作的效果，所以在本研究中选择相加操作，即：



上式中，乘以是至关重要的，由于词汇表中存在一些非组合性词语，这样的一个系数就可以缩小组合性词语的词向量和非组合性词语的词向量之间的差距。

在基于字和词语联合训练的词向量计算中，字向量的确定是一个关键性的问题，由于汉字往往具有很强的歧义性，同一个汉字在不同的组合中或不同的语境中往往有着不同的字义，为了解决这个问题，为每一个汉字设立多重向量，每一重向量分别代表某一种含义，并建立了多标准的字向量确定方法，即基于位置的字向量，基于簇的字向量，并在此基础上将两者结合，尝试建立基于位置和簇的字向量；

1）基于位置的字向量：

在汉语中，一个字在词语中的位置不同，往往就会扮演者不同的角色，有不同的含义，因此在基于位置的字向量中为每一个字设立三重不同的向量，分别对应前、中、后三个不同的位置；以下图所示为例，词语“依法”的词向量是由“依法”这个词语的向量和组成其的字“依”“法”在其对应位置上的字向量共同决定的。



图3：基于位置的字向量

抽象的表示为，假设有上下文词语和组成词语的字，通过一个字在词中的不同位置获得不同的向量，当建立向量时，用表示词中第一个字的向量，用表示中间的字的向量，用表示末尾的字的向量，那么，公式（4）可以被重新定义：



根据公式（3），可以将得到的进一步用于计算，得到上下文词语的平均向量；

1. 基于簇的字向量

根据文献（）中介绍的多标准字向量的方法，依据字的上下文，将这个字的每种发生情况聚集在一起，形成这个字的多个标准，对于每一个字，将它的所有发生事件聚拢到个簇中，然后为每一个簇设立一个向量；由于本论文是针对法律领域社区问答的，训练所用的数据集以及系统所涵盖的语料库是限定在一定主题范围内的，这样就保证了簇的个数不会太大，避免了因为簇的个数过多导致模型计算复杂度过高，训练周期过长的问题；



图4：基于簇的字向量

该模型运用了k-means算法思想的部分原理，也就是对每一个汉字提前分配个字向量，的个数是模型的一个超参数，代表了潜在定义的每个汉字所对应的语义模式，也可以称为模式向量，如图4 所示，将其抽象化为，上下文词语，是被用来获取的，定义为余弦相似度，则：



其中



代表了汉字曾经被选择最多次的模式向量，也就是说，在对汉字进行模式向量筛选的时候，利用了该汉字对应词语的context信息，而context信息就是该汉字对应词语前后窗口内的个词语，而这些词语的向量形式由词语向量和字向量叠加形成，只是在选择字向量的模式向量的时候，直接选择过去被挑选最多次的模式向量，利用conntext向量，从一个汉字的所有模式向量中选择一个和context语义计算上最相似的作为该汉字对应的向量，即，则：

，

计算出后，通过公式（3），便可以进一步计算。

1. 基于位置和簇的字向量

 图5：基于位置和簇的字向量

本文尝试将基于位置的方法和基于簇的方法结合起来，从位置和簇的角度共同出发决定字向量，同一个字，按照其出现的不同位置分为B、M、E三种不同的类型，同一个字在不同的词语中同一位置出现，又被分为不同的簇，即不同的模式向量，假设只有B、M、E三种不同的位置，将式（）进行简化，则：



其中每一个位置上都有不同的模式向量，模式向量的选择按照前面式（）和式（）中介绍的方法，选择和该字的context语义计算最相似的作为该汉字对应的向量，这样，就有：



仍然以上图中所示为例，组成词语“依法”的字“依”和“法”，在确定其字向量的过程中，既考虑了位置的信息，也考虑了簇的信息，根据以上介绍的算法，“依”取其最前面位置的第2个簇，“法”取其最后面位置的第3个簇，同理“罚”取其最前面位置的第1个簇，“金”取其最后面位置的第2个簇，最后，通过词语向量和其字向量的结合得到“依法”和“罚金”的词向量，进而可以通过式（）计算得到。

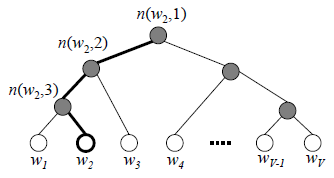
3.3 基于字和词语联合训练的词向量计算

本文提出的字词联合训练的词向量方法是可以应用于多种词向量模型中的，本文以CBOW模型为基础，将字词联合训练的方法应用于模型训练的过程，继续沿用CBOW中的Hierarchical Softmax方法代替传统的Softmax层，使用随机梯度上升法得到模型中词语的词向量和内部节点的模型参数；

本节介绍了CBOW模型训练过程中所使用的Hierarchical Softmax模型优化方法以及模型梯度计算方法，最后，介绍了基于CBOW的字词联合训练的词向量模型在相关数据集上的训练过程；

3.3.1Hierarchical Softmax优化和梯度计算

传统的神经网络语言模型一般有三层，输入层，隐藏层和输出层（softmax层），其最大的问题就在于从隐藏层到输出的softmax层的计算量很大，所以提出了 Hierarchical Softmax算法对其进行优化，这种算法为了为了避免要计算所有词的softmax概率，采用了霍夫曼树来代替从隐藏层到输出softmax层的映射。



如上图所示，霍夫曼树的所有内部节点就类似神经网络隐藏层的神经元,其中，根节点的词向量对应投影后的词向量，而所有叶子节点就类似于神经网络softmax输出层的神经元，叶子节点的个数就是词汇表的大小，在霍夫曼树中，隐藏层到输出层的softmax映射不是一下子完成的，而是沿着霍夫曼树一步步完成的,即层次softmax.每一层都采用二元逻辑回归的方法，使用sigmoid函数计算游走到左右子节点的概率，规定沿左子树走霍夫曼树编码为1，沿右子树走霍夫曼树编码为0，则：





对于上图中的，如果它是一个训练样本的输出，则期望对于里面的隐藏节点的概率大,的概率大,的概率大，而控制和的概率值大小的因素是当前节点的词向量和当前节点的模型参数，即（1）式中的和，模型训练的目标上就是找到合适的所有节点的词向量和所有内部节点，使得训练样本达到最大似然，对于词语，期望最大化下面的似然函数：



将上述过程进行更一般化的描述，定义输出的词为,从输入层词向量求和平均后的霍夫曼树根节点词向量为，从根节点到所在的叶子节点，包含的节点总数为, 在霍夫曼树中从根节点开始，经过的第个节点表示为,对应的霍夫曼编码为,其中。而该节点对应的模型参数表示为, 其中，，没有是因为模型参数仅仅针对于霍夫曼树的内部节点。

定义经过的霍夫曼树某一个节点的逻辑回归概率，其表达式为：



　那么对于某一个目标输出词,其最大似然为：



对数似然函数为：



使用梯度上升法求得模型中词向量和内部节点的模型参数，具体的计算过程如下：







同样的方法，求得的梯度表达式如下：



有了梯度表达式，就可以用梯度上升法进行迭代来一步步求解需要的所有和。

3.3.1 数据预处理

**词向量是基于某个特定语料的，单独拿出任一个词向量都没有什么意义，它的作用在于众多词向量之间的差异所表现出来的语义表达；**

本文是基于特定领域的答案选择的研究，以法律咨询领域的问答系统为课题背景，本文所用数据集都是法律领域的相关数据，采用分布式爬虫技术，爬取华律网，知乎，百科等多个网站的相关数据，其中大部分来源于华律网，

在上一章的2.6小节，介绍了用于本文研究的相关数据集的来源与获取，在本节中，针对词向量层面的研究对此数据集做相应的预处理，以用于后续的实验；

首先，对采集到的文本语料进行分词处理，由于是中文语料，一句话中词与词之间没有间隔，所以分词处理非常必要，分词效果甚至会间接影响后续的模型训练结果，本文使用中科院的NLPIR分词工具对所有问题候选答案对进行分词处理；并使用停用词表去除停用词

其次，对于模型训练初始阶段字向量初始化如何设定的问题，有两种解决方案，一种是沿用CBOW、Skip-gram和Glove的做法，将字向量的初始化同其他模型参数一样进行随机初始化，一种是使用预训练好的字向量对其进行初始化，为了获取更好的训练效果，本文采用后者的方法，将字看成单独的词，使用word2vec词向量工具对其进行预训练，并以预训练完成后的向量作为模型训练初始阶段字向量的初始化；

3.3.2 模型训练过程



1.读取语料，统计词频信息，构建词典，将词频数小于一定阈值的词语去掉；（去掉非常高频的词-下采样，和非常低频的词。去掉高频词是因为没有特殊性，去掉低频词是因为没有普适性）

3.根据词典和词频构建huffman树，词频越大的词编码越短；

4.随机初始化所有的模型参数，随机初始化所有的词向量，维度默认为200，以之前预训练好的向量作为字向量的初始化向量，在实际的训练过程中，取，分别取前、中、后三个位置，每个位置上设三个簇，即设置三重模式向量；上下文窗口大小设置为；

5.以行为单位训练模型，将输入文件以最大1000个词切割为多行；

6.获取当前行中的一个输入样本，即当前词向量及其上下文词语的词向量；

5.进行梯度上升迭代过程，对于训练集中的每一个样本，作如下处理：

a) 计算;

if CBOW-position=1 

if CBOW-cluster=1 

If CBOW-pos&clu=1 

b) for j=2 to 计算：









c) 对于中的每一个词向量(共2c个)进行更新：



　　d) 如果梯度收敛，则结束梯度迭代，否则回到步骤6继续迭代。

3.3 实验结果与分析

根据前面两节介绍的方法，对基于CBOW的字词联合训练的词向量模型进行训练，得到了输入语料中所有词的词向量，本节以CBOW词向量模型为基线方法，从词语相关度计算、词语逻辑推理两个方面对基于CBOW的字词联合训练的词向量模型进行分析和评价

3.3.1 模型复杂度分析（对比三种模型，以表格的方式呈现）

表1：模型复杂度

|  |  |  |
| --- | --- | --- |
| 模型 | 模型参数 | 计算复杂度 |
| CBOW |  |  |
| CBOW-position |  |  |
| CBOW-cluster |  |  |
| CBOW-pos&clu |  |  |

上表第二列显示了两种模型的模型参数,代表向量维数，是词典大小，是字典大小，代表一个词语中字的个数，代表每一个字所拥有的簇的个数。

第三列中显示了两种模型的计算复杂度，CBOW模型中，窗口大小为，数据集的大小为，每个词语的平均字长是，计算每一个目标词语的Hierarchical softmax计算复杂度是，表示以CBOW模型训练词向量时的计算复杂度，当使用CBOW-CE模型时，计算复杂度额外增加了和用以计算字向量字向量，但是，和的取值往往比较小，在实际训练中一般取2到4之间，所以计算字向量的复杂度要远远小于。

3.3.2 词语相关度计算对比

衡量一个词向量模型的功能和性能，仅凭模型训练出来的单个词语的词向量去衡量是无法得出有说服力的结论的，因为词向量是基于某个特定语料的，单独拿出任一个词向量都没有什么意义，它的作用在于众多词向量之间的异同所表现出来的语义表达，所以，通过判断模型是否能准确计算词语之间的语义相关性来衡量模型性能的好坏是一种常用的词向量模型的评测方法。（原创）

在本实验中，分别构建了两个由180对词语和230对词语组成的汉语词对数据集，并以0-10之间的实数人工标注每对词语的相关性得分，在第一组数据集wordpair-180中，有172对词语是出现在训练语料库中的，而在另外的8对词语中有8个未在训练语料库中出现的新词语，在第二组数据集wordpair-230中，有214个词语对是存在于训练语料库中的，在剩余的16对词语中有16个未在训练语料库中出现的新词语，图3-7中截取了wordpair-180和wordpair-230中的部分词语对和人工进行标注的相关性得分；

对于CBOW模型和其各种改进模型来说，是通过计算两个词向量之间的余弦相似度来获得两个词语的相关性得分的，在模型训练出词向量的基础上，计算词语对数据集中每对词语的相关性得分，假设数据集中的每对词语的词向量表示为和，则相关性得分的具体计算如下式：



具体代码实现如下所示：

public class GetScore{

public static void Main()

{

BufferReader br2 = new BufferReader(InputReader(new

FileInputStream("b.txt")))

BufferedWriter bufWrite = new BufferedWriter(new

OutputStreamWriter(new FileOutputStream(filename), "UTF-8");

while ((string str = br.readLine()) != null)

{

string []words = str.sqlit(' ');

double []a = findVector(words[0],'a.txt');

double []b = findVector(words[1],'a.txt');

double result = calResult(a[],b[],200);

str = str + " " + result + "\r";

bufWrite.write(str);

}

br2.close();

bufWrite.close();

}

public static double calResult(double []a,double []b,int num)

{

double sumAB=0;double sumAA=0;double sumBB=0;

for(;num>0;num--)

{

sumAB+=a[num]\*b[num];

sumAA+=a[num]\*a[num];

sumBB+=b[num]\*b[num];

}

return sumAB/(sqrt(sumAA)\*sqrt(sumBB));

}

}

针对词语对数据集中出现的新词语，即没有在训练集中出现的词语，CBOW模型因无法对其做进一步的操作，将包含新词语的词语对的相关性得分置为0，而CBOW的改进模型可以通过字向量计算得到这些新词语的词向量，从而进行相关性计算，

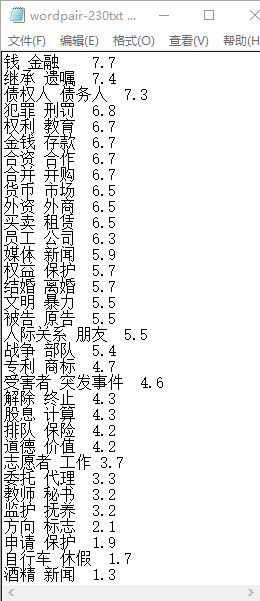
 

图3-7 wordpair-180和wordpair-230中部分词语

不同的词向量模型训练出不同的词向量，不同的词向量可以计算得到不同的词语对相关性得分，本实验在wordpair-180和wordpair-230两个词语对数据集上，通过计算不同模型得到的词语对相关性得分和人工标注的词语对相关性得分之间的相关系数，来评价模型的性能,的计算公式如下：



相似度计算正确率（表格）

实验中训练好的模型会计算wordpair-180和wordpair-230词语对集合中

|  |  |  |
| --- | --- | --- |
| Dataset | wordpair-180 | wordpair-230 |
| Method | 172pairs 180pairs | 214pairs 230pairs |
| CBOW | 53.72 53.97 | 60.53 53.69 |
| CBOW-position | **58.03**  58.26 | 62.46 61.65 |
| CBOW-cluster | 56.59 57.93 | 62.40 61.03 |
| CBOW-pos&clu | 57.62 **58.80** | **64.52 63.93** |

图3-8 wordpair-180和wordpair-230上的评估结果

上图显示了各个词向量模型分别在wordpair-180和wordpair-230两个词语对集合上进行评估的结果，从图表中可以看到，在wordpair-180和wordpair-230这两个词语对数据集上，基于字词联合训练的CBOW模型相比传统的CBOW模型，整体性能较为突出；在180对数据集上和去掉包含7个新词语的剩余172对数据集上，基于字词联合训练的CBOW模型相对于传统CBOW模型，效果并没有太高的提升，原因是包含7个新词语的词语对本身的相关性就比较小，在传统 CBOW模型中以0作为其相关性打分影响并不会很大；而在230对数据集和去掉包含16个新词语的剩余214对数据集上，传统的CBOW模型效果有很明显的浮动，而字词联合训练的CBOW模型效果相对稳定，原因是传统的CBOW模型由于无法计算新词语的词向量而将本身具有一定关联的词语对的相关性置为0，而基于字词联合训练的CBOW模型可以通过字向量计算词向量，从而得到一个较为合理的相关性得分；在基于字词联合训练的CBOW模型的所有方法中，CBOW-position相比于CBOW-cluster效果更好，CBOW-pos&clu的效果最好，原因是，本文的研究数据集主要是针对法律领域，专业领域内同一个词出现在不同的词语的同一位置，对词语语义的影响并不会很大，比如“公平”和“公正”这两个词语中都出现的“公”字，虽然在两个不同的簇内，但意义却是一样的，对词语语义的影响也是很小的，所以CBOW-cluster效果并不明显，而CBOW-pos&clu结合了CBOW-position和CBOW-cluster的方法，将字的位置信息和簇的信息进行结合，取得了更好的效果。

3.5 本章小结

本章介绍了一种改进的词向量模型，在传统的以词语为单位训练词向量的方法上进行改进，提出了字和词语联合训练的CBOW词向量模型，在字和词语联合训练的方法中，字向量的选取对词向量的生成有着直接的影响，本章针对法律领域的数据集，建立了多标准的字向量确定方法，基于位值的字向量，基于簇的字向量，并尝试将两者结合，提出基于位置和簇的字向量确定方法；

本章在相关数据集上对所提出的改进方法进行了验证，并从模型的复杂度和词语对相关度计算两个方面对模型性能进行了评估，确定了基于字和词语联合训练的词向量模型的可靠性和有效性，并将此模型训练出来的词向量作为下章研究内容的基础。

3.1研究动机



图3-1 基于深度学习的答案选择排序研究框架图

图3-1是基于深度学习的答案选择排序的研究框架图，从图中可以比较清楚地看出，基于深度学习的答案选择排序的基本过程和研究核心，灰色部分即为本文研究的重点内容，分别在本章和第四章作以详细的介绍，如图所示，首先将问句和候选答案对组成的自然语句语料库进行预处理，通过词向量模型，生成词向量，再通过神经网络模型对问句和候选答案进行句子对建模，生成句子的语义特征向量，然后将其输入到全连接层，通过softmax函数计算得到问句和候选答案的语义相关性得分，即每个候选答案的准确性得分；

在模型中，问句和候选答案以词向量矩阵的形式输入到深度神经网络模型端进行句子对建模，那么词向量作为模型的输入，可将其视为模型的一个超参数，其表达词语信息的性能的好坏直接影响着后期句子语义建模的效果，本章就是针对如何构建性能更好的词向量模型，以及如何在相关数据集上获得能够更好地表达词语信息的词向量的问题而展开的；