1 介绍

情感分析是自然语言处理中一个重要的领域，其价值在于使政府、电商等组织了解人们对某一事件或是某一商品的态度，从而做出更好的决策。因此，如何使得情感分析，即情感分类的准确度得到提升，是一个充满价值的研究问题。

大多数情感分析都会使用词嵌入的方法，词嵌入通俗来说就是把句子中的每个单词映射成一个固定长度的密集实值向量，用来描述单词的语义、语法等特征。最常使用的代表性方法包括Mikolov等人[52]提出的Word2Vec词嵌入以及Pennington等人[53]提出的GloVe词嵌入。很多研究者采用这两种方法得到词的向量化表示，然后应用于情感分类等任务中[使用了这两种方法的文献]，并得到了很好的结果。

然而，即便这些方法非常优秀，但它们也存在一些局限。首先，这些方法的训练需要大量的数据。比如说，Word2Vec方法就是在千亿词汇量的语料库上进行训练的，所得出的词嵌入维度为300维；此外，在用于特定的任务如情感分析时，这些方法没有考虑词的上下文信息或是情感信息，而只包含了单词的语义和语法特征。而且，对于一词多义的情况，比如说：apple在不同的语境下可能代表一种水果，也可能代表苹果手机，但是这些方法会把apple这个词映射成固定的向量，这显然是不合理的。因此，如何使得单词嵌入技术更加准确地描述单词的各项特征，且具有较低的维度，是一个值得研究的方向。

目前，随着深度学习的不断发展，越来越多的研究者也使用深度学习来进行情感分析。比如[一些经典的深度学习情感分析论文]。使用深度学习进行情感分析的主要步骤是：首先利用词嵌入技术得到单词的向量表示，再将这些向量传入深度学习模型中进行训练，调整模型的参数，从而得到适用于情感分类任务的模型。Yoon Kim等人[31]使用卷积神经网络（CNN）来进行情感分类，虽然这种方法比起传统的基于机器学习或基于词典的方法得到了较大的准确度的提升，但是CNN模型只能捕获句子中几个单词之间的局部特征，无法提取整个句子的长期依赖关系；针对这个问题，[经典的RNN论文]采用RNN来进行情感分类，RNN是一种循环神经网络，可以在一定程度上捕获句子的长期依赖关系，然而当序列很长时，RNN面临着梯度消失或爆炸的问题，为此，有研究者采用一种更高效的RNN模型——LSTM来进行处理。LSTM可以改善上述问题，但由于LSTM的结构比较复杂，在多层LSTM堆叠时，模型参数过多难以训练。

针对上述问题，本文提出了一种基于扩展词向量的CNN-Bi-GRU模型(EWE-CBG)，扩展的词向量可以得到单词除了语义、语法之外更多的信息，CNN用于对高维的词向量进行降维，提取出单词向量中最重要的特征，双向GRU在捕获句子前向和后向依赖关系的同时，简化了LSTM的结构，使得模型在准确度与LSTM相当的基础上，缩短了参数训练的时间。我们的实验在基准数据集上与传统方法进行了比较，结果表明，这种基于词扩展-降维-训练的模型有助于情感分类精度的提升。

我们的论文组织如下：第二节介绍关于深度学习情感分析的相关工作和文献综述，第三节介绍我们所提出的模型的架构以及每个组件的实施细节，第四节展示了提出的方法与基准方法的对比，第五节对整篇论文的工作进行总结，并讨论该方法可以改进的方向及未来的展望。

2 相关工作

2.1 单词的向量表示

近几年来，深度学习方法在自然语言处理领域充当了重要的角色。NLP中的深度学习任务大多是面向使用单词向量表示的方法。单词的连续向量表示算法，如Word2Vec和GloVe，是一种可以将单词转换为有意义向量的深度学习技术。词汇的向量表示在文本分类、聚类和信息检索中具有非常重要的意义。与词袋表示相比，词嵌入技术有一些优势。例如，意义相近的单词在单词嵌入空间中是相近的。此外，词嵌入的维数也低于词袋(Mikolov et al.，2013)。

Word2vec和GloVe的准确性取决于文本语料库的大小。这意味着，准确率随着文本语料库的增大而提高。Tang et al.(2014)提出在大型社交网络数据集Twitter上学习连续词表征进行情绪分析的方法。Severyn和Moschitti(2015)使用Word2Vec方法学习了5000万条tweet上的单词嵌入，并将生成的预训练向量作为深度学习模型的输入。Fu等(2017)将Word2Vec应用于英文维基百科数据集和中文维基百科数据集的嵌入。Lauren, Qu, Zhang, and Lendasse(2018)最近提出了一种判别式文档嵌入方法，该方法使用skip-gram来生成文本的词嵌入，嵌入词用作递归自动编码器的输入，从而进行情感分析。Ren et al.(2016)提出了一种新的推特情绪分类词表示方法。他们使用Word2Vec在他们的方法中生成一些数据集的单词嵌入。Qin, Xu, and Guo(2016)利用英语维基百科语料库对Word2Vec算法进行训练，该语料库有4.08亿个单词。他们使用这些预先训练的向量作为数据驱动任务的卷积神经网络的输入。

然而，如上所述，这些词嵌入算法需要大量的文本语料库进行训练(Giatsoglou et al.，2017)，其中大部分忽略了文本的情感信息(Araque et al.，2017;唐等，2014)。由于某些语料库的局限性，研究人员更倾向于使用预先训练好的词嵌入向量作为机器学习模型的输入。Kim(2014)将训练好的Word2Vec向量作为卷积神经网络的输入，提高了情绪分类的准确性。CamachoCollados等(2016)也使用了预先训练好的Word2Vec向量来表示概念。Zhang和Wallace(2015)将训练过的GloVe和Word2Vec向量应用于深度学习模型中，提高了句子和情绪分类的准确性。Caliskan等人(2017)使用事先训练的GloVe词嵌入向量来提高他们提出的方法的准确性。Wang et al.(2016)将预先训练的GloVe向量作为基于注意力的LSTMs模型的输入，用于aspect-level的情绪分析。Liu et al.(2018)使用经过训练的Word2Vec作为文章写作中推荐习语的嵌入表示。然而也有一些方法在使用了Word2Vec和GloVe之后准确率降低了，如Zhang和Wallace(2015)将训练前的Word2Vec和GloVe向量结合在他们的深度学习模型中，但是准确率却下降了。

因此，提高预训练词嵌入的准确性非常重要，在情绪分类方法中起着至关重要的作用。

2.2 RNNs

基于深度学习的情感分析中，在得到了词的向量表示后，一般将词向量传入深度学习模型以得到句子特征的向量表示，从而得到情感极性分类。卷积神经网络(CNN)是一种常用的深度学习模型，Yoon Kim等人[31] 使用CNN来进行情感分类，得到了比RAE、SVM[详见论文31-P4]等机器学习方法更高的准确度。然而CNN只能提取句子中的局部特征，而RNN（循环神经网络）由于允许同一隐藏层的神经元之间传递信息，因此更适合处理句子这样的长序列。Tang等[78]提出了一个包含文档级SA的门控RNN模型，其关键思想是从单词到句子级，从句子到文档级的单词表示的迭代生成。在这种情况下，CNN和门控RNN分别用于生成句子和文档表示。门控RNN能够对文档语义中的句子之间的内在关系进行编码，因此比IMDB数据集上的几种最新算法具有更好的性能。除了直接用于SA之外，基于RNN的方法也适用于提取文本特征。 Lee等[79]使用RNN提取短文本的顺序信息，然后将短文本表示形式输入到两层ANN中以预测电影评论的情感极性。随后的实验证明，RNN给出的顺序信息显着提高了预测的质量。

然而，RNN单元结构简单，容易出现梯度消失或爆炸的问题。因此，越来越多的方法使用LSTM来进行情感分析。LSTM是RNN的一种变体，通过遗忘门、输入门和输出门控制信息的传递从而改善了简单RNN梯度消失和爆炸的问题。LSTM用于情感分析的代表性方法包括TD-LSTM (TargetDependent LSTM),82 AT-LSTM (Attention LSTM)83 and ATAE-LSTM (Attention LSTM with Aspect Embedding)83 ,这些方法应用于面向方面的情感分析并取得了很好的结果。Pal等人[84]提出了一种双向LSTM，与较简单的LSTM版本相比，它显示了具有双向连接的分层深LSTM在精度方面具有更好的性能。Chen等人[85]设计了一个LSTM网络用于中国产品评论，以进行细粒度的SA。 实验结果表明，在对中国产品评论进行分类时，所提出的模型优于基线方法。Ma et al.86增强了基本的LSTM网络，它具有由目标级别的注意力组成的分层注意力机制。 在他们的工作中，将与情感相关的概念的常识知识纳入了用于情感分类的深度神经网络的端到端训练。 提出的方法称为Sentic LSTM，在目标方面的情感任务方面胜过了最新技术。Ruder等[89]提出了一种用于情感分析的分层双向LSTM模型。 单词嵌入首先被输入到句子级别的双向LSTM中。 然后，将前向和后向LSTM输出的级联句子级别向量进一步馈入双向文档级别LSTM。 在11个数据集上进行的大量实验表明，通过允许模型考虑句子间的关系，它超越了仅依赖句子信息的方法。

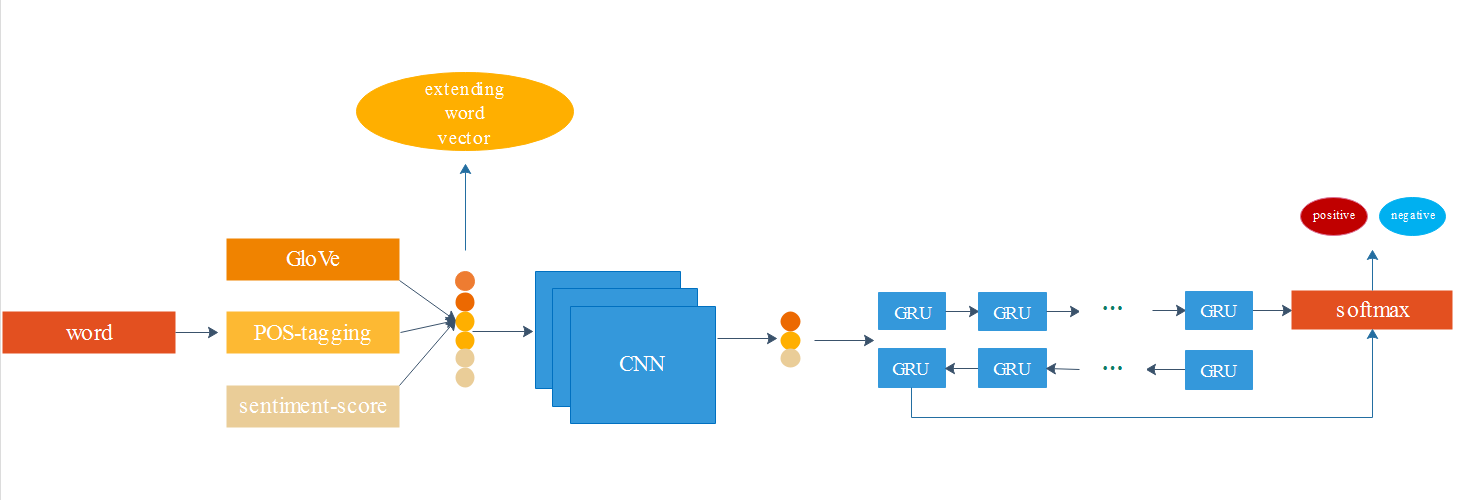
虽然大部分基于LSTM的情感分析都取得了良好的结果，但是LSTM的结构比较复杂，因此[55]采用GRU——一种和LSTM效果相当但简化的模型来进行情感分析。该方法使用CNN注意力机制与GRU相结合来进行情感分析并在SemEval-2018数据集上取得了良好的结果。Luo等人[57]采用一种与LDA主题模型相结合的GRU-CNN方法进行网络文本的情感分析，并提升了分类的准确度。[58]提出了一种集成了LSTM和GRU的语言模型对恶意软件进行分类；[59]通过减少更新门和重置门的参数，提出了GRU的三种变体，在MNIST和IMDB数据集上的实验结果表明，这些方法使用更简单的模型得到了和GRU等同的效果。

可以看出LSTM和GRU在文本分类任务上取得了出色的成绩，这些方法也为本文所述的基于扩展词向量的CNN-Bi-GRU方法奠定了坚实的基础。

3 基于扩展词向量的CNN-Bi-GRU情感分析模型

虽然GRU在提取序列特征时表现出了突出的性能，但是它只能顺序地处理序列的信息，无法捕捉句子中从后到前的依赖关系，从而造成了准确度的下降。因此在提出的方法中，我们采用双向GRU来捕获前向和后向的序列特征，以提升分类准确度。此外，大多数基于深度学习的情感分析都使用Word2Vec或者GloVe预训练词嵌入作为词的向量表示，这两种方法虽然能够刻画每个单词的语法及语义信息，但是却忽略了其他一些重要的信息比如单词的情感和词性特征。因此，我们采用GloVe+情感特征+词性特征所组合而成的词嵌入向量，以扩展神经网络的输入。由于得到的嵌入向量是高维的，直接作为深度学习的输入会使训练变得困难且漫长，因此我们采用CNN对得到的嵌入向量提取特征，降低输入数据的维度，从而进一步提升模型的效率。最后，我们把双向GRU最后一层的输出结果传入一个softmax分类器中，得到情感极性的分类。通过整合扩展词嵌入、CNN和双向GRU，就得到了我们的基于扩展词向量的CNN-Bi-GRU模型。

整个模型的构架如图\*所示，其中，输入的是句子，输出的是情感极性类别。



3.1 扩展的词向量

为了使得输入深度学习的词向量具有更丰富的表示，我们使用情感特征和词性特征扩展传统的GloVe嵌入，使得词向量不仅包含语义信息，还包含情感和词性特征，从而得到更高的分类准确度。

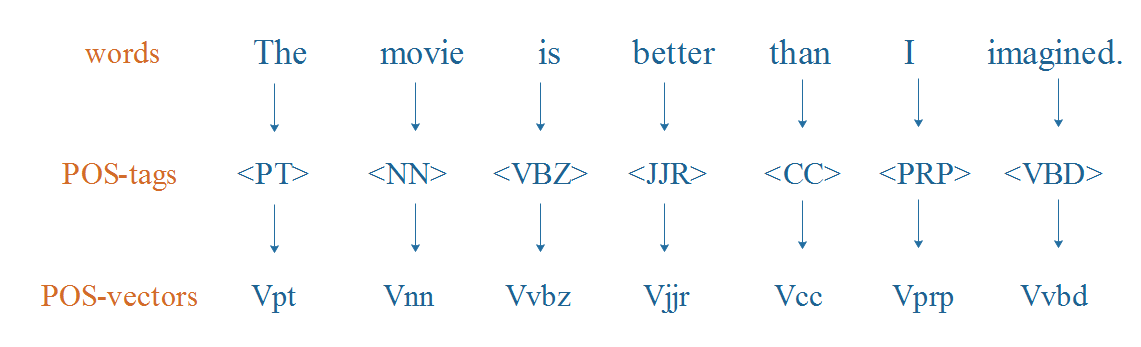
3.1.1 GloVe词嵌入

为了刻画单词的语义及语法特征，我们需要选用合适的词嵌入方法把单词映射成向量。传统的one-hot方法将单词表示成长度为词汇量、仅在对应位置上为1、其它位置全为0的高维稀疏向量，且忽略了单词的顺序特征。Word2Vec和GloVe是词嵌入领域中非常成功的两种方法，Word2Vec使用Skip-gram和CBOW两种模型把意义相似的单词映射到距离相近的向量空间；GloVe word embedding is a global log-bilinear regression model and is based on co-occurrence and factorization of a matrix in order to get vectors.它和Word2Vec相比，考虑了全局信息，且容易并行化。本文采用包含20亿条推文的预训练语料库来训练我们的GloVe模型，并选取200维作为GloVe词向量的维度。

假设文本包含M个单词，则第k（其中k∈[1,2,…,M]）个单词的的GloVe词嵌入向量表示为，其中表示嵌入的维度。

3.1.2 POS-Tagging特征

词性标注是自然语言处理中一个重要的步骤，为了增加情感分析的准确度，我们引入单词的词性特征以扩充词的向量表示。首先我们利用词性标注技术得到每个单词的词性标记（名词、动词、形容词等），再将这些词性标记转化成一个称为词性嵌入的实值向量，并与GloVe词向量相连接，从而使得模型能够从词性的角度来理解句子的含义。我们采用Stanford parser来得到每个单词的词性标签，再利用类似于one-hot嵌入的方法将标签转化为对应的实值向量，如图（\*）所示。由此我们得到了第k个单词的词性嵌入为,其中表示词性嵌入的维度。



3.1.3 情感特征

除了语法、语义和词性特征之外，单词的情感特征也有助于提升情感分类的准确性。为了引入单词的情感特征，我们参照[50]中的方法，使用TS-Lex, S140-Lex, SD-Lex and SWN-Lex这四个词典来提取每个单词的情感得分。这些词典的积极词数、消极词数和总词数见表（\*）

|  |  |  |  |
| --- | --- | --- | --- |
| Lexicon | positive | negative | total |
| TS-Lex | 2547 | 2448 | 4995 |
| S140-Lex | 15568 | 17412 | 32980 |
| SD-Lex | 33997 | 32026 | 66023 |
| SWN-Lex | 24156 | 28312 | 62468 |

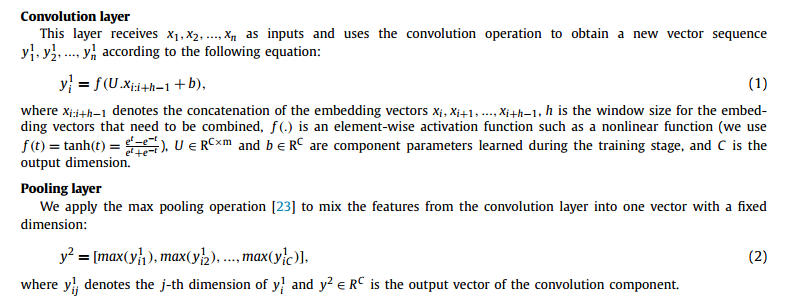
**TS-Lex**3 is a large-scale sentiment lexicon built from Twitter by Tang et al. (2014a) for learning  
sentiment-specific phrase embeddings. **S140-Lex**4 is the *Sentiment140* lexicon, which is built from point-wise mutual information using distant supervision (Go et al., 2009; Mohammad et al., 2013).**SD-Lex** is built from SST. We construct a sentiment lexicon from the training set by excluding all neutral words and adding the aforementioned offset -2 to each entry. **SWN-Lex** is a sentiment lexicon extracted from SentimentWordNet3.0 (Baccianella et al., 2010). For words with different part-of-speech tags, we keep the minimum negative score or the maximum positive score. The original score in the SentimentWordNet3.0 is a probability value between 0 and 1, and we scale it to [-2, 2]5.

When building these lexicons, we only use the sentiment scores for unigrams. Ambiguous words are discarded. Both TS-Lex and S140-Lex are Twitter-specific sentiment lexicons. They are used in the Twitter sentiment classification task. SD-Lex and SWN-Lex are exploited for the Stanford dataset.这样我们就得到了单词k的情感特征，表示情感嵌入的维度，在这里我们使用了四个词典，则情感嵌入的维度为4.

最后，对于每个单词，我们将和连接起来，就得到了扩展的词嵌入（EWE）。

其中.

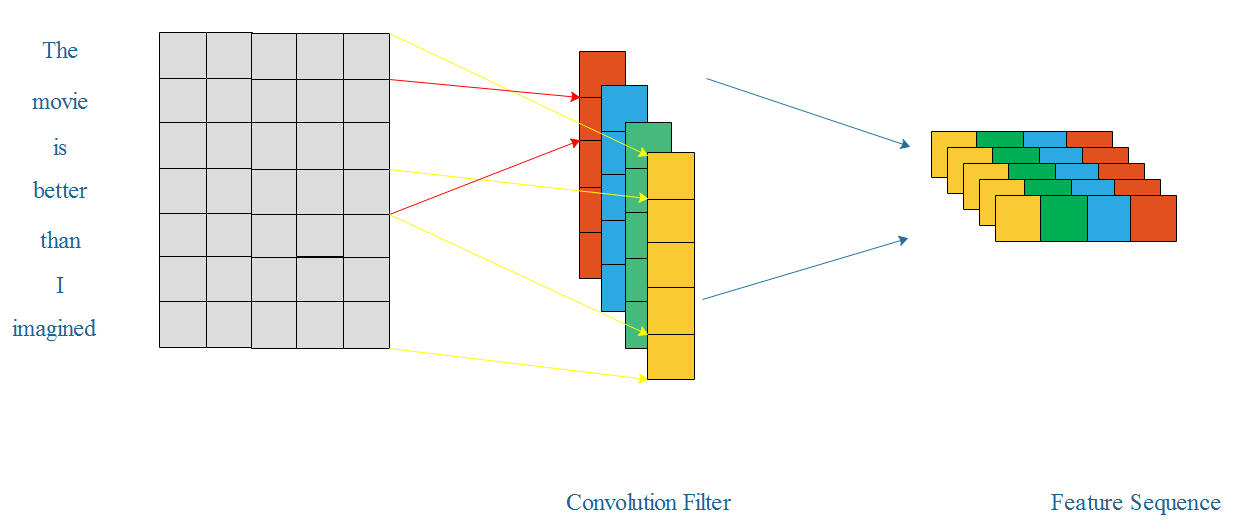
3.2 CNN层用于特征提取和降维



https://www.cnblogs.com/baiting/p/6101981.html

一般情况下，在得到了词的向量表示之后，就会把单词向量组成的句子矩阵输入CNNs或RNNs进行训练，学习模型的参数，从而得到适用于情感分类任务的深度学习模型。然而，句子矩阵是一个高维的矩阵，直接输入RNNs会降低模型的性能。因此在将句子矩阵输入双向GRU之前，我们引入一个单层的CNN提取出数据中最重要的特征，再将低维的特征序列传入双向GRU中，从而达到了降维的目的。

在本论文提出的方法中，我们采用窗口大小为3,4,5的100个卷积滤波器提取单词序列的n-gram特征。滤波器在句子矩阵上每滑动一次就会提取一次特征，这些特征包含了单词之间复杂的语义关联和语法特征。它们将作为双向GRU的输入，来进一步提取句子中各个单词前向和后向的长期依赖关系。



我们采用窗口大小为h的100个卷积滤波器在句子矩阵中从上至下滑动提取局部特征。令表示句子的第k个扩展词嵌入，则表示从到组成的单词序列矩阵，其中表示连接操作。令表示某个卷积滤波器第i次滑动所提取的特征，则：

其中W表示该卷积滤波器的权重，b表示偏置项。f是一个非线性函数例如双曲正切或ReLU函数。假定句子的长度为n，则对于数据集中长度不足n的句子以0填充，超过n的部分则截去，再用公式\*进行处理，就得到了该卷积滤波器的输出：

它表示的是整个句子在窗口大小为h下的局部特征。

我们采用100个卷积滤波器提取特征，相当于将每个单词原有的240维特征压缩成了100维，输出的是n-h+1个长度为h的序列的100维特征向量，但是这样的做法对于句子两端的单词是不公平的（比如说句子的第一个单词和最后一个单词都只提取了一次特征），因此，我们将矩阵前后采用0向量补充，再进行宽卷积操作，从而得到了降维后的特征序列。每个序列的特征从原来的240维降低到了100维，从而降低了输入数据的维度。这些特征序列将被输入到双向GRU中，进一步捕获句子的长期依赖关系。

**全连接层**

为了将卷积层得到的局部信息整合起来，我们采用全连接层输出序列的总体特征。令表示全连接层第i个神经元的输出，其计算如下：

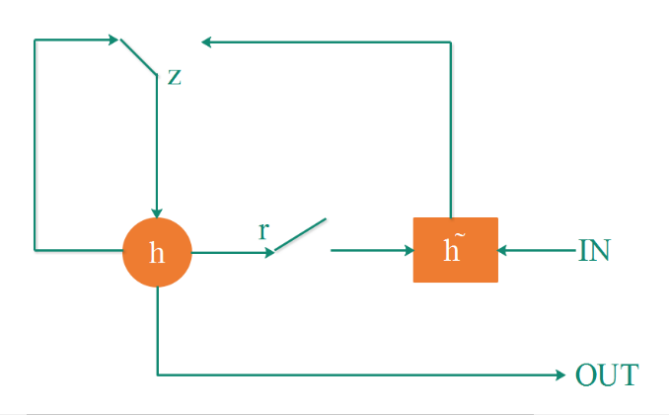
其中表示全连接层第i个神经元和卷积层得到的每个局部特征之间的权重，表示偏置项,g是一个非线性函数例如双曲正切或ReLU函数。全连接层的神经元个数设为m，为后续将要调整的超参数之一，根据上述计算公式，得到了由全连接层输出的特征序列：

它代表句子在大小为h的窗口上一系列整合后的局部特征，特征序列层和过滤器窗口层中颜色相同的块对应于同一窗口的特征

3.3 双向GRU和softmax

3.3.1 GRU

论文[61]首次提出了Gated recurrent unit(GRU)，GRU和LSTM类似，具有门控单元，门控单元可以确定信息流是否要通过该单元。与LSTM相比，GRU没有记忆单元，GRU仅具有两个门，即更新门和重置门。 图\*是GRU的基本结构。



由于省去了记忆门，所以在大多数情况下，GRU的训练速度都比LSTM快一些，且能得到和LSTM准确度相当的结果。GRU的计算过程如下：

GRU的激活函数在t时刻的值是上一个时刻激活函数和候选激活函数的线性插值，即：

其中是GRU单元中的更新门，它决定了先前的记忆有多少需要流入当前的单元。表示重置门，它用来确定如何将当前的输入和先前的记忆组合起来，GRU使用门控机制来捕获序列长期依赖关系的基本思想和LSTM是一致的。如果将重置门全部置为1，更新门全部置为0，就得到了最普通的RNN模型。

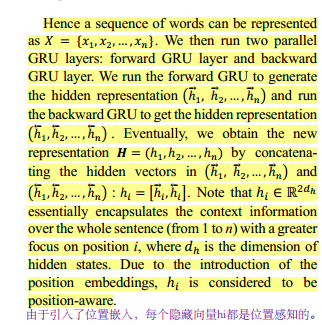
GRU候选激活的计算类似于传统的递归神经网络，如下：

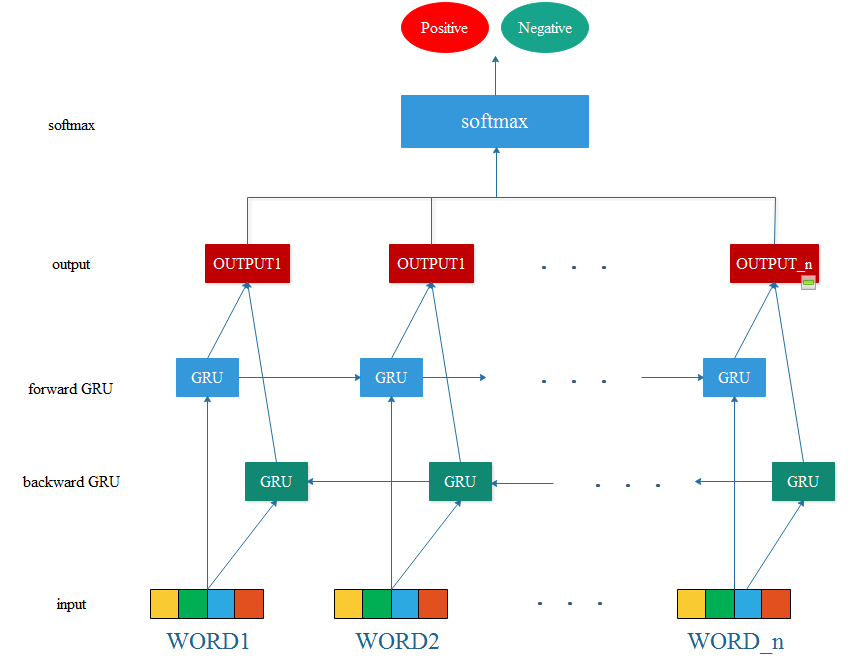
其中表示element-wise multiapplication.

3.3.2 Bi-GRU

对于传统的递归神经网络（RNN），它们只能处理从前到后的序列。 由于无法学习未来的信息，单向的RNN导致了信息的丢失。 因此，[62]提出了一种双向递归神经网络（Bi-RNN），它增加了从后到前的另外一个RNN来处理未来的信息。 Bi-RNN的基本结构主要是将一个公共RNN分为两个方向，一个方向按正向时序，另一个方向按反向时序，再将每个单词在两个方向的RNN的最后一层连接到相同的输出层。 这样的结构可以为输出层的输入序列提供完整的上下文信息。

使用Bi-GRU对方面级别的情感分析进行建模需要输入的历史数据同时输入正向GRU和反向GRU，从而捕获完整时间序列的上下文信息。图\*展示了使用Bi-GRU对降维后的特征序列进行学习的过程，其中每个输入表示的是由卷积层得到的每个单词的100维特征向量。





我们将每个单词前向和后向的GRU单元的每个输出传入到一个softmax分类器中，得到最终的情感极性分类。其中，softmax层是一个激活函数，其目的是将GRU得到的特征表示向量转化为各个标签的概率分布。 在此二分类类任务中，我们将双向GRU的输出记作特征表示向量v，并把它作为softmax层的输入，计算如下：

其中L表示情感标签的类别个数，表示情感类别l的预测概率。

在采用反向传播算法训练模型时，我们使用交叉熵损失函数作为模型的损失函数，通过最小化损失函数来不断优化模型，并采用L2正则化器来消除模型的过拟合。损失函数计算方式如下：

其中，表示句子的真实情感极性（准确值），表示模型的预测值。I表示第i个句子，j表示情感类别标签，为L2正则化器的权重，为模型训练中所有的参数。

我们提出的算法如算法\*：

|  |
| --- |
| Algorithm 1 Extending Word Embeddings with CNN and Bi-GRU model (EWE-CNN-BiGRU) |
| Inputs:  ,sentence with n words  GloVeV:vectors of GloVe embeddings  PT:36 POS taggings from Stanford Parser  SL:sentiment lexicons  h:number of sentiment lexicons  Output:sentiment polarity   1. For each in S: 2. If exists in GloveV: 4. Else 5. ← RandomGenVec (Vec())  8. For j = 1 to h: 9. If exists in : 11. else 13. For i in 1 to n: 14. ← 15. Employ the convolution layer to obtain the feature sequence EWE-; 16. Employ the Bi-GRU layer to obtain the feature sequence EWE-CNN-; 17. Employ the softmax layer to obtain the sentiment polarity; |

综上所述，我们提出的EWE-CNN-Bi-GRU模型的主要贡献如下：

1. 除了Word2Vec/GloVe词向量之外，我们采用词性特征和情感特征扩充每个单词的信息，以得到更适用于情感分类任务的单词向量表示。
2. 采用CNN层对高维的词向量进行降维，得到单词更加密集的词向量，作为单词最重要的特征表示。
3. 和传统的RNN相比，使用双向GRU可以提取句子的长期依赖关系，进一步对单词的上下文信息建模，并且与LSTM相比可缩短模型的训练时间。

4 实验

我们通过在不同的情感分类基准数据集上对我们的方法进行实验，以验证模型的有效性。本章主要介绍实验用到的数据集、用以对比的基准方法，以及实验的结果。除了主方法之外，我们还在副实验中比较了模型中每个组成部分对最终结果的影响程度，以探究各个组件在情感分类方面的重要性。

4.1 实验设置

4.1.1 参数设置

在我们的实验中，词向量采用300维的word2vec词嵌入来表示。句子的最大长度以数据集中句子的平均长度来确定，并只保留最常出现的前10000个单词。对于CNN层，我们使用滑动窗口大小分别为3,4,5的卷积滤波器在句子矩阵上进行滑动，每个大小的滤波器数量为100. 对于双向GRU层，我们使用100个隐藏单元来处理单词间的依赖关系，全连接层使用的激活函数为ReLU，最后一层全连接层使用softmax激活函数得到最终的情感分类。

在训练阶段，我们设置dropout比率为0.5,并在每个全连接层中采用l2正则化器以防止过拟合，批处理的大小为64，设置最大迭代轮次为60，并采取早停策略得到最优的模型参数。

4.1.2 基准数据集

我们采用的数据集如下：

IMDB：IMDB数据集包含来自互联网的50000条严重两极分化的评论，该数据被分为用于训练的25000条评论和用于测试的25000条评论，训练集和测试集都包含50%的正面评价和50%的负面评价。该数据集已经经过预处理：评论（单词序列）已经被转换为整数序列，其中每个整数代表字典中的某个单词。

SST-1：Standford Sentiment Tree，包含Rotten Tomatoes网站的10662条影评，每条影评被划分为非常积极/积极/中立/消极/非常消极，是一个五分类任务的数据集。

SST-2：将SST-1数据集中去掉中立类，转化成标签为积极/消极的二分类问题。

4.2 基准方法

为了衡量提出的模型的性能，我们选择了以下几种最先进的方法进行比较。

SVM：SVM with uni-bi-trigrams, wh word, head word, POS, parser, hypernyms, and 60 hand-coded rules as features from Silva et al. (2011).

NBSVM&MNB:Naive Bayes SVM and Multinomial Naive Bayes with uni-bigrams from Wang and Manning (2012).

RAE：Recursive Autoencoders with pre-trained word vectors from Wikipedia

CCAE:Combinatorial Category Autoencoders with combinatorial category grammar operators (Hermann and Blunsom, 2013)

DCNN:Dynamic Convolutional Neural Network with k-max pooling (Kalchbrenner et al., 2014).

CNN-static:1d-CNN with pre-trained word embedding vector from word2vec proposed by Kim

CNN-non-static:1d-CNN with pre-trained word embedding and fine-tuning optimizing strategy proposed by Kim

CNN-multichannel:1d-CNN with two sets of pre-trained word embeddings proposed by Kim

MV-RNN:Matrix-Vector Recursive Neural Network with parse trees (Socher et al., 2012)

DRNN:DRNN: Deep recursive neural networks with stacked multiple recursive layers proposed by Irsoy and and Cardie

LSTM:Long short term memory.

Tree-LSTM:A generalization of LSTM to tree structured network topologies proposed by Tai et al.

Multi-task LSTM:A multi-task learning framework using LSTM to jointly learn across multiple related tasks proposed by Liu et al.

P-LSTM:A model introduces the phrase factor mechanism which combines the feature vectors of the phrase embedding layer and the LSTM hidden layer to extract more exact information from the text proposed by Lu et al.

C-LSTM:A model combining with the strengths of CNN and RNN for sentence representation and text classification proposed by Zhou et al.

Bi-LSTM:Bidirectional long short term memory.

4.3 结果

在这一小节中，我们将本文提出的方法与上述基准方法在不同的数据集上的准确度进行了对比，并探究了本文所提出的方法中各个组件以及实验中超参数对分类效果的影响。

4.3.1总体比较

|  |  |  |  |
| --- | --- | --- | --- |
| Model | IMDB | SST-1 | SST-2 |
| SVM | 89.2 | 40.7 | 79.4 |
| NBSVM | 91.2 | - | - |
| MNB | 86.6 | - | - |
| RAE | - | 43.2 | 82.4 |
| CCAE | - | - | - |
| DCNN | - | 48.5 | 86.8 |
| CNN-static | - | 45.5 | 86.8 |
| CNN-non-static | - | 48.0 | 87.2 |
| CNN-multichannel | - | 47.4 | 88.1 |
| MV-RNN | - | 44.4 | 82.9 |
| DRNN | - | 49.8 | 86.6 |
| LSTM | 87.0 | 48.0 | 86.4 |
| Tree-LSTM | - | 50.6 | 86.9 |
| Multi-task LSTM | - | 49.6 | 87.9 |
| P-LSTM | 90.5 | - | - |
| C-LSTM | - | 49.2 | 87.8 |
| Bi-LSTM | 87.9 | 48.4 | 88.0 |
| EWE-CBG | **91.6** | **51.0** | **90.2** |

表（\*）展示了本文提出的EWE-CBG方法与上述基准方法在不同数据集上的准确度的比较，每个数据集上的最优结果采用黑体字加粗表示。从整体比较的结果来看，我们得到了如下结论：

1. 实验中采用的数据集包括长评论（IMDB）、短评论（SST-1&SST-2），二分类问题（IMDB&SST-2）和多分类问题（SST-1），从表中的结果可以看出，我们的方法在不同任务的数据集上都达到了最好的效果。在IMDB、SST-1和SST-2这三个数据集上分别达到了91.6%、50.9%以及83.4%的准确度，比起现有的最好结果分别提升了0.3%（NBSVM on IMDB）、0.4%（Tree-LSTM on SST-1）、1.1%（CNN-Multichannel on SST-2）。
2. 对于前5种传统的机器学习方法（SVM、NBSVM、MNB、RAE和CCAE），在这几个数据集上，提出的方法和它们的平均准确度相比分别提升了2.6%、9.05%和9.3%；对于CNN及其不同变体所对应的方法（DCNN、CNN-static、CNN-non-static和CNN-multichannel），本文方法的准确度在这些数据集上平均提升的准确度为-（在IMDB数据集上，CNN类方法没有给出相应的结果）、3.65%和2.98%，对于RNN类的方法（MV-RNN，DRNN，LSTM，Tree-LSTM，Multi-task LSTM，P-LSTM，C-LSTM以及Bi-LSTM），本文的方法在这些数据集上提升的平均准确度为3.13%、2.43%和3.56%（见下表）。可以看出对于SST这样的短评论数据集，我们的方法相对于传统机器学习方法的优势更为明显，因为RNN的引入可以进一步捕获复杂的句法关系，从而获得更高的准确度；对于CNN-based方法和现有的RNN-based方法，本文的方法由于对词向量进行了特征的补充，丰富了深度学习模型的输入表示，使得模型在训练时引入了单词的词性特征和情感特征，因此得到了比现有方法更好的结果。

|  |  |  |  |
| --- | --- | --- | --- |
| Method | IMDB | SST-1 | SST-2 |
| Traditional Machine Learning | 2.6% | 9.05% | 9.3% |
| CNN-based | - | 3.65% | 2.98% |
| RNN-based | 3.13% | 2.43% | 3.56% |

4.3.2 各组成部分的影响

本文的主要贡献在于以下三点：

1. 将词性特征和情感特征连接至词向量，扩充深度学习的输入表示；
2. 采取CNN对输入向量降维，提取最重要的语义特征
3. 采用双向GRU捕获句子序列的长期依赖关系

为了探究各个组成部分对实验结果的影响，我们又进行了如下实验。

**输入特征的影响：**由于采用了三种特征来将单词映射成向量，分别是词嵌入、词性嵌入和情感嵌入。对于词嵌入，我们分别使用不同的词嵌入方法，包括GloVe，CBOW和Skip-gram来刻画单词的语义信息；对于词性嵌入，我们考虑是否引入该特征，来探究词性信息对分类效果的影响；对于情感嵌入，我们考察不引入词典、引入单个词典、引入多个词典对分类效果的影响。具体实验结果见表（\*）

|  |  |  |  |
| --- | --- | --- | --- |
| Word Embedding Method | IMDB | SST-1 | SST-2 |
| GloVe | **91.6** | **51.0** | **90.2** |
| CBOW | 91.4 | 49.9 | **90.2** |
| Skip-gram | 91.2 | 49.9 | 90.1 |

从表（\*）中可以看出，在使用GloVe嵌入来表示单词的语义特征时，比Word2Vex提供的CBOW和Skip-gram这两种模型效果更好。这是由于GloVe词嵌入基于单词的共现矩阵，考察了全局信息，可以加快模型的训练速度，又可以控制词的相对权重。而CBOW和Skip-gram每次仅使用一个窗口的信息来更新词词向量，很容易使得高曝光词汇得到更高的权重，因此在后续实验中，我们保持使用GloVe词嵌入描述单词的语义信息。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| POS feature | Sentiment Lexicons | IMDB | SST-1 | SST-2 |
| USE | NOT USE | 89.2331 | 49.4232 | 87.6830 |
| TS-Lex | 90.7140 | 50.6032 | 89.3069 |
| S140-Lex | 91.2422 | 50.4873 | 89.2501 |
| SD-Lex | 90.6865 | 49.9581 | 89.7258 |
| SWN-Lex | 91.5276 | 50.9930 | 89.6944 |
| ALL | **91.6424** | **51.0206** | **90.2370** |
| NOT USE | NOT USE | 88.9637 | 48.6705 | 87.7506 |
| TS-Lex | 90.5306 | 50.3396 | 89.0331 |
| S140-Lex | 91.0024 | 50.3014 | 89.1012 |
| SD-Lex | 90.4745 | 49.6968 | 89.4799 |
| SWN-Lex | 91.2689 | 50.7225 | 89.3260 |
| ALL | **91.4509** | **50.9966** | **89.8565** |

为了探究词性特征和情感特征对分类准确度的影响，我们分别添加或者去掉词性特征、选用一个或者多个情感词典进行实验，所得结果如表（\*）所示。我们可以看出，在使用词性特征并且考虑所有情感词典时，该模型能获得最高的准确度。在使用词性特征时，使用所有情感词典比不使用情感词典在3个数据集上的准确度分别提升了2.4039%，1.5974%和2.554%，在不使用词性特征时，使用所有情感词典比不使用情感词典在3个数据集上的准确度分别提升了2.4872%，2.3261%和2.1059%。对词性特征来说，在使用全部情感词典时，使用词性特征比不使用词性特征在3个数据集上的准确度只提升了0.1915%，0.024%和0.3805%，这说明在处理情感分类任务时，词性特征和情感特征的引入能有效地提升分类准确度，相比之下，情感特征对于提升模型性能所发挥的作用更大，因此在计算单词的向量表示时，引入情感特征有助于模型训练出更适合情感分类任务的参数。

**CNN层的影响：**哈哈加油