学号 20164981 密级\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**东北大学本科毕业论文**

基于深度学习的对象级情感分析算法的研究与实现

学 院 名 称：软件学院

专 业 名 称：软件工程

学 生 姓 名：张馨

指 导 教 师：张伟 副教授

XXX 工程师

**2020 年 5 月**

基于深度学习的对象级情感分析算法的研究与实现

**标题段落格式：**

段前1行，段后1行，

行距32磅，居中、。

|  |  |  |
| --- | --- | --- |
| 作者姓名： | 张馨 | |
| 校内指导教师： | 张伟 | 副教授 |
| 校外指导教师： |  | 工程师 |
| 单位名称： | 软件学院 | |
| 专业名称： | 软件工程 | |

东 北 大 学

2020年5月

**Research and Implementation of Aspect Level Sentiment Classification Algorithm Based on Deep Learning**

**此页段落格式：**

段前1行，段后1行，

行距32磅，居中。

by Zhang Xin

|  |  |  |
| --- | --- | --- |
| Supervisor: | Associate Professor | Liu Hongyi |
| Associate Supervisor: | Engineer |  |

Northeastern University

May 2020

**郑 重 声 明**

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

本人签名： 日期：

摘 要

情感分析是自然语言处理的基础模块。近年来，由于分析粒度不断细化，对象级情感分析的应用和研究价值逐渐增加，使得其成为情感分析中最重要的研究方向之一。基于深度学习的对象级情感分析可以相对准确的获得大批量文本中情感对象对应的情感极性，使其超越传统技术占据对象级情感分析的主导地位，但不可否认仍有许多问题存在。

本文首先对对象级情感分析算法的相关研究现状进行分析，重点对深度学习在对象级情感分析算法的应用进行研究。本文着重于对文本中的情感对象与上下文语境进行分析，利用深度学习算法来进行情感分析任务，并且针对现有的算法存在的主要问题、缺陷进行改进，提出了一种新的对象级情感分析网络模型。模型重点针对多个词的对象或上下文可能带来信息丢失问题和忽视具有相同上下文的对象之间的对象级别交互可能带来信息丢失问题进行构建。针对具有多个词的对象或上下文，模型利用细粒度的注意力机制，用以捕获对象和上下文之间的单词级交互。同时，针对具有相同上下文的对象，模型通过设计方面对齐损失来描述其之间的方面级别交互。另外，本文在三个情感分析经典数据集上评估所设计的模型：SemEval2014的laptop和restaurant以及twitter数据集。另外，还使用显着性检验对该算法进行了通用性测试，测试结果表明了算法的准确性。

然后，本文提出的结合细粒度和粗粒度的注意力机制来构成多注意力粒度交互网络（MGIAN）框架用以对象级情感分析具有较高的分类准确性，并且分析算法简单、易实现，可以有效应用到对象级情感分析上。

**关键词：**情感分析；注意力机制；对象级情感分析

摘要正文：

宋体，小四号，段前0行，段后0行，1.5倍行距，首行缩进2字符。

400到700字之间。

**ABSTRACT**

Sentiment analysis is the basic module of natural language processing. In recent years, due to the continuous refinement of analysis granularity, the application and research value of object-level sentiment analysis have gradually increased, making it one of the most important research directions in sentiment analysis. Object-level sentiment analysis based on deep learning can relatively accurately obtain the sentiment polarity corresponding to sentiment objects in a large number of texts, making it beyond the traditional technology to occupy the dominant position of object-level sentiment analysis, but there are still many problems that cannot be denied.

This article first analyzes the current research status of object-level sentiment analysis algorithms, focusing on the application of deep learning to object-level sentiment analysis algorithms. This article focuses on the analysis of emotional objects and context in the text, uses deep learning algorithms to perform emotional analysis tasks, and improves on the main problems and defects of existing algorithms, and proposes a new object-level emotion Analyze the network model. The model focuses on objects or contexts with multiple words that may cause information loss and ignoring object-level interactions between objects with the same context may cause information loss. For objects or contexts with multiple words, the model uses a fine-grained attention mechanism to capture word-level interactions between objects and context. At the same time, for objects with the same context, the model describes aspect-level interaction between them by designing aspect alignment loss. In addition, this paper evaluates the designed model on three classic datasets of sentiment analysis: laptop and restaurant of SemEval2014 and twitter dataset. In addition, the algorithm was tested for versatility using a significance test. The test results show the accuracy of the algorithm.

Then, this paper combines the fine-grained and coarse-grained attention mechanisms to form a multi-grained attention network framework for object-level sentiment analysis with high classification accuracy, and the analysis algorithm is simple and easy to implement and can be effectively applied to objects Sentiment analysis.

**Key words:** Sentiment analysis ; Attention mechanism ; Aspect sentiment analysis

目 录

[摘 要 II](#_Toc19304)

[ABSTRACT III](#_Toc22538)

[第1章　绪 论 1](#_Toc23936)

[1.1 论文排版总体要求 1](#_Toc24502)

[1.1.1 打印 1](#_Toc9021)

[1.1.2 页码编排 1](#_Toc13406)

[1.1.3 页眉设置 1](#_Toc32586)

[1.1.4 装订 1](#_Toc12105)

[1.2 论文主要部分 2](#_Toc2550)

[1.2.1 前头部分 2](#_Toc8998)

[1.2.2 主体部分 2](#_Toc24674)

[1.2.3 结尾部分 3](#_Toc16630)

[1.3 本文结构 3](#_Toc32336)

[第2章　论文前头部分 5](#_Toc28980)

[2.1 封面 5](#_Toc32544)

[2.1.1 论文题目 5](#_Toc11079)

[2.1.2 学生信息 5](#_Toc18429)

[2.1.3 论文封面日期 5](#_Toc25991)

[2.2 中英文题名页 5](#_Toc27733)

[2.2.1 题目 5](#_Toc19031)

[2.2.2 姓名 6](#_Toc26370)

[2.2.3 学校与日期 6](#_Toc800)

[2.3 郑重声明 6](#_Toc21972)

[2.4 中英文摘要 6](#_Toc12351)

[2.4.1 摘要标题 6](#_Toc4702)

[2.4.2 摘要内容 6](#_Toc7442)

[2.4.3 关键词 6](#_Toc6705)

[2.5 目录 7](#_Toc31185)

[2.5.1 目录 7](#_Toc2540)

[2.5.2 索引条目 7](#_Toc28889)

[第3章 正文部分 8](#_Toc18888)

[3.1 各级标题 8](#_Toc23433)

[3.2 正文 8](#_Toc16266)

[3.3 图 8](#_Toc15084)

[3.4 表 9](#_Toc14593)

[3.5 程序代码 10](#_Toc31424)

[3.6 公式 10](#_Toc7035)

[3.7 注释 11](#_Toc31126)

[3.8 引文标示 11](#_Toc2003)

[第4章 结尾部分 13](#_Toc26403)

[4.1 参考文献 13](#_Toc24586)

[4.2 参考文献格式 13](#_Toc20993)

[4.3 参考文献举例 14](#_Toc17031)

[参考文献 15](#_Toc18532)

[致 谢 16](#_Toc28784)

第1章　绪 论

本章首先描述了对象级情感分析技术的发展现状；然后说明了本文提出的基于深度学习的对象级情感分析算法的研究意义，接着阐述了对象级情感分析技术在国内外的研究现状；之后描述了本课题的主要研究内容，引出本课题的研究重点；最后说明本论文的组织结构。

## 1.1 课题研究背景

论文三级标题：

黑体，小四号，段前0.5行，段后0.5行，单倍行距，居左，编号用Times New Roman。

由于互联网的兴起以及各种网络媒体的出现，互联网中用户参与度越来越高，越来越多的互联网用户从被动地接受互联网信息向主动创造互联网信息迈进。因此，互联网（微博、论坛）上产生了大量用户参与的、对诸如产品、服务、事件等有价值的文本评价信息。这些文本评价信息包含了用户的主观情感倾向，表达了用户对某一事件或产品的看法，是宝贵的数据资源。但由于数据量庞大且产生速度极快，远远超过人工自身可以处理的能力，对使用人工的方法进行统计分析产生了巨大挑战，所以迫切需要利用计算机形成对互联网文本中的观点进行快速获取和整理的可行性解决方案。

同时，对互联网中产生大量评价信息利用深度学习进行对象级情感分析在经济、社会等多个维度具有重要的意义。在商品零售领域，其可以帮助产品商家通过这些数据量化用户对产品的诉求及其竞品的褒贬程度，从而了解确定产品改进和发展方向；在社会舆情领域，其可以帮助人们通过分析大众对于社会热点事件的点评有效的掌握舆论的走向，从而更加全面的思考和判断；在金融交易领域，其可以帮助投资者通过这些数据分析股票及其他金融衍生品的态度，为行情交易提供辅助依据。

因此，越来越多的学者通过不断研究对象级情感分析的特点，增加其分析的准确性和高效性，解决大量文本信息的情感极性提取问题。

## 1.2 课题研究意义

自2006年Hinton等提出深度学习后，深度学习在计算机视觉方面取得了巨大的成就，越来越多的学者开始将深度学习应用到自然语言处理领域。由于短文本包含的信息量较少，同时传统方法不能发现文本中的深层语义信息，所以深度学习凭借其抽象特征，减少人工成本成为情感分析的主流方法。

## 1.3 国内外研究现状

情感分析，是自然语言处理(NLP)的一个领域，它构建的系统用于在文本中识别和提取观点。情感分析的目标是从文本中分析出人们对于实体及其属性所表达的情感倾向以及观点，这项技术最早的研究始于2003年Nasukawa和Yi两位学者的关于商品评论的论文。随着推特等社交媒体以及电商平台的发展而产生大量带有观点的内容，给情感分析提供了所需的数据基础。时至今日，情感分析已经在多个领域被广泛的研究应用。

根据已有的研究，情感分析可分为以下三种粒度，文档级，句子级，对象级。文档级情感分析将整篇文档作为分析的基本单元，输入一篇文档，利用情感分析模型，输出这篇文档的情感极性，正面、负面或者中性。但这种级别的情感分析在文档设计多个实体或对象并且表达出不同的情感倾向时，无法有效传达出准确的信息。句子级情感分析将每个句子作为分析的基本单元，其输出是每个句子的情感极性为正面、负面或者中性。文档级和句子级情感分析都无法精准分析出用户对于具体的实体或者对象表达出的情感极性。而对象级情感分析直接关注在情感极性和对应的对象上，其任务就是分出出文本中设计到的每一个对象上的情感极性。

针对对象级情感分析这个问题上，已有的研究方向主要包括两个方面，传统分类起方法以及深度学习方法。传统分类器方法，将这个问题视作一个分类问题来处理，手动设计一组特征，之后用机器学习的方法训练文本分类器。随着情感词典的丰富，Mohammad等[]构建了基于词典的特征以进行情感分析。这些研究大多集中在使用支持向量机（SVM）构建具有特征的情感分类器上，包括词袋和句子词典（Mullen和Collier，2004）。 但是，这种传统分类器的分析结果在很大程度上取决于设计的特征的质量。 另外，特征工程是劳动密集型的。

随着深度学习的发展，很多深度学习模型迁移到对象级情感分析上。经典模型包括Socher等[]提出的递归神经网络，Socher等[]提出的递归神经张量网络，Mikolov等[]提出的递归神经网络，Hochreiter和Schmidhuber[] 提出的LSTM和Tai等【】提出的Tree-LSTMs。 通过利用句子的语法结构，基于树的LSTM已被证明对于许多NLP任务非常有效。Wang等人【】还引入了注意力机制来表征情感对象对执行模型的影响，以更加关注上下文中的重要词语。之后引入注意力机制研究对象级情感分析工作主要采用简单的平均方面向量来学习上下文词的注意力权重。 Ma等人【】进一步提出了一种双向注意力机制，该机制以交互方式分别针对情感对象/上下文单词的平均向量，学习上下文/情感对象单词的注意力权重。

以上所有这些注意力机制都是在粗粒度级别上进行的，它们仅对情感对象/上下文向量进行平均即可指导学习上下文/方面单词上的注意力权重。这样简单的平均池化机制可能会导致信息丢失，尤其是对于具有多个单词或较大上下文的方面而言。

因此，本文提出了一种细粒度的注意机制，用于刻画情感对象词和上下文词之间的单词级交互，并缓解在粗粒度注意机制中发生的信息丢失。此外，本文还将双向粗粒度注意力与细粒度注意力向量结合，以构成用于最终情感极性预测的多注意力粒度交互网络（MGIAN），从而可以利用其两者的优势。同时为了利用相同情感对象的交互信息，本文在目标函数中设计了方面对齐损失，以增强注意力在具有相同上下文和不同情感极性情感对象的权重的差异。

## 1.4 论文研究内容

本文主要研究了基于深度学习的对象级情感分析算法。通过分析其算法的原理，以及实现方式，并对比不同苏娜发的解决思路以及缺陷，生成初步的改进方向。然后依据对象级情感分析国内外先进的研究成果，选定基本的对象级情感分析解决模型，并将模型与之前的改进方向结合，对模型进行进一步的改良，使其满足对象情感分析的分析准确度需求，同时也能满足处理大批量数据的高效性需求，最终完成一个准确、高效的基于深度学习的对象级情感分析算法。

本文首先了解对象级情感分析的相关工作，分析对象级情感分析的主要切入点以及现有的对象级情感分析算法的不足和缺陷，同时结合现有达到较好效果的注意力网络结构，设计一种更加有效的对象级情感分析网络模型架构、损失函数；然后结合模型架构、神经网络设计技巧等，对神经网络的网络结构进行研究与设计，使其能充分获取文本中的情感信息，最终提出行之有效的对象级情感分析算法。

之后对提出的对象级情感分析算法进行一系列的实验比对评估，判断其情感分析能力的准确性，有效度，检验提出的对象级情感分析算法是否符合预期结果。

## 1.5 论文组织结构

本论文分为五章，每章组织结构安排和内容如下：

第一章绪论，主要介绍了对象级感情分析的研究背景，讲述了目前对象级情感分析在国内外的研究现状，并简要说明了本课题的意义和内容，最后介绍了本论文的组织结构。

第二章相关工作与理论基础，介绍了对象级情感分析现有的相关工作，给出对象级情感分析算法的相关概念，针对对象级情感分析模型的评估，还介绍了常用的情感分析评估指标。

第三章基于深度学习的对象级情感分析算法的分析与设计，首先分析了当前基于RNN的情感分析算法的缺陷与不足，然后提出本文设计的基于LSTM的对象级情感分析算法，阐述其改进方法，接着对生成对象级感情分析的神经网络模型的结构设计、目标函数的选择进行说明，最后给出模型训练的总体设计和伪代码。

第四章算法的实现与评测，首先介绍了对象级感情分析中重要部分的关键代码和流程图，接着说明了实验环境，介绍了训练、测试过程中选择的数据，以及将进行的相关测试；然后对模型中的算法实现部分先进行了实验与分析并将多个算法进行了对比；之后对不同算法分析的结果利用直方图等方法进行实验并分析结果，最后给出了最终的实验结论。

第五章总结与展望，首先总结了全文的工作，介绍了本文取得的研究成果，分析其中的创新点和不足之处，而后说明了本研究进一步的研究方向和大致思路。

第2章　相关工作与理论基础

本章将对现有的对象级情感分析算法进行介绍，分析现有对象级情感分析算法存在的问题，接着阐述长短期注意力网络，注意力机制，Dropout等处理对象级情感分析的基本概念和基本原理，最后说明常用的对象级情感分析的评估指标。

## 2.1 现有对象级情感分析算法

论文三级标题：

黑体，小四号，段前0.5行，段后0.5行，单倍行距，居左，编号用Times New Roman。

对象级别的情感分析是情感分类的一个分支，需要同时考虑句子和对象信息。

传统方法如（Jiang等人[]; Kiritchenko等人[];Duy-Tin Vo等人[]）把这一任务视为文本分类的问题，设计有效的特征，并利用统计学习算法训练分类。Kiritchenko等人[]提出使用基于n-gram特征，解析特征和词典特征的SVM（支持向量机），从而在SemEval 2014中获得最佳性能。Duy-Tin Vo等人[]提出特定于情感的单词嵌入和情感词典作为预测的额外特征。 然而，这些方法是劳动密集型的，其高度依赖费力的特征描述工作的有效性，并且很容易达到性能瓶颈。

近年来，由于基于深度学习的方法能够在没有特征工程的情况下将原始特征编码为连续的低维向量，因此有越来越多学者进行研究。Socher等人[] 和Dong等人[]以及Nguyen和Shirai []提出了递归自动编码器，即根据句子的树型递归结构，使用递归神经网络对句子进行语义合成，并生成用于预测的特征表示。例如，假设一个句子中的词表示为一个m维的向量，递归自动编码器可以把整个句子也表示成为一个m维的向量。在这样的树型递归结构中，叶子代表句子中的词，中间节点代表其所有子节点的复合表示。从而根节点就可以代表整个句子的含义。

Hochreiter 和 Schmidhuber []提出了基于LSTM的方法来对上下文信息进行建模，并使用集合矢量进行预测。Tang等人[]提出的TD-LSTM对情感对象前后两部分文本分别使用LSTM建模（LSTM-L，LSTM-R），其中前后各部分都包含目标词，LSTM-L从左向右，LSTM-R从右向左。模型对LSTM-L和LSTM-R得到的最终的隐藏向量进行拼接作为预测的表示形式。但是，这些工作仅专注于对上下文建模而不考虑情感对象，而情感对象在估计情感极性方面起着重要作用。

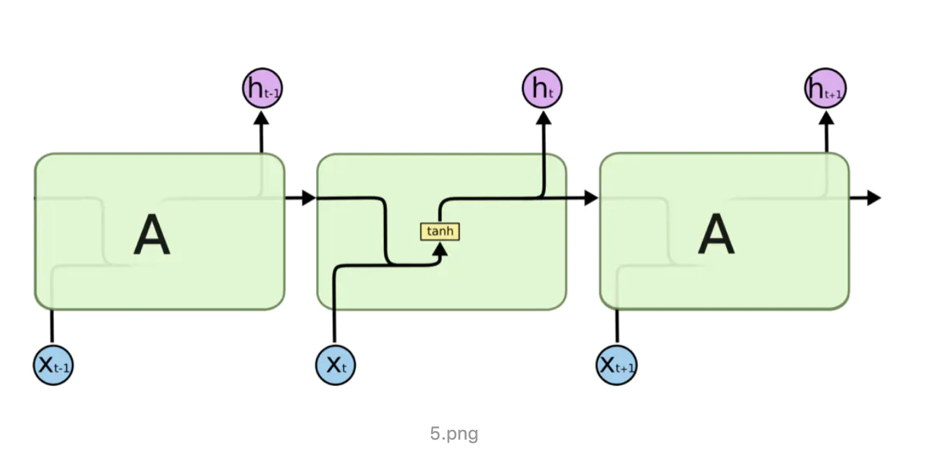
注意力机制能够降低对其他信息的关注度，甚至过滤掉无关信息，并提高任务处理的效率和准确性。Tang等人[]采用平均对象向量来学习上下文词的隐藏向量上的注意力权重。Ma等人[]进一步提出了双向注意机制，该机制有效利用了情感对象和上下文的交互关系，不仅学习了上文词对情感对象词平均向量的注意权重，还学习了情感对象词对上下文词平均向量的注意权重。这些注意力方法通过使用简单的平均情感对象向量/上下文向量来进行对上下文/方面词的注意力权重学习，这种仅仅考虑粗粒度级别的注意力，这可能会在具有多个词的长情感对象或长上下文的情况下导致某些信息丢失。

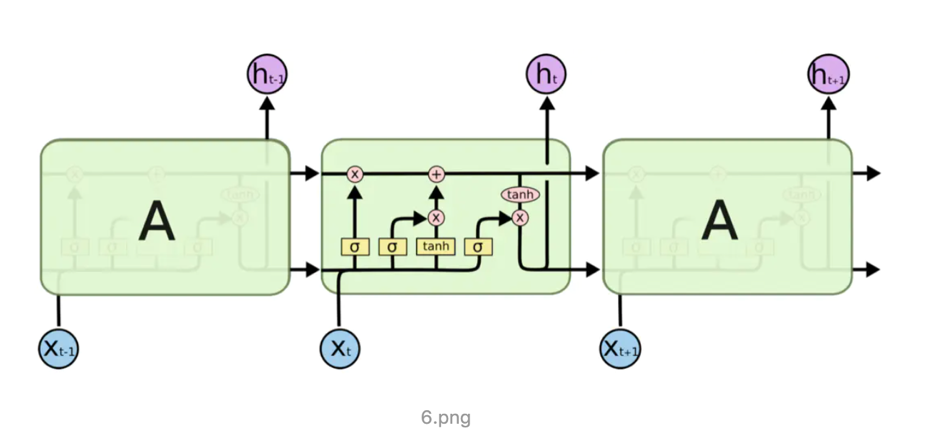
## 2.2 长短期注意力网络

论文三级标题：

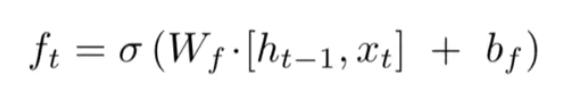
黑体，小四号，段前0.5行，段后0.5行，单倍行距，居左，编号用Times New Roman。

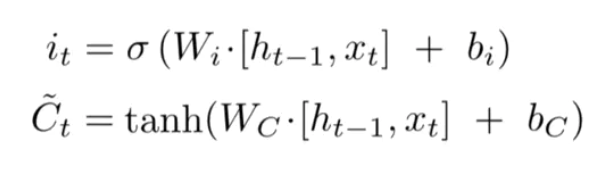
长短期记忆网络(Long Short Term Memory network，LSTM)是为了RNN学习解决长期依赖时梯度消失问题由Hochreiter 和Schmidhuber【】所提出的一种特殊的 RNN网络。

所有循环神经网络都具有神经网络的重复模块链的形式。 在标准的RNN中，该重复模块将具有非常简单的结构，例如单个tanh层。标准的RNN网络如下图2.2所示。

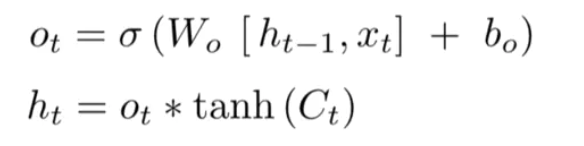
LSTM也具有这种链式结构，但是它的重复单元不同于标准RNN网络里的单元只有一个网络层，它的内部有四个网络层，并以特殊的方式进行交互。LSTMs的结构如下图3.3所示。

LSTM神经网络包含两个状态:隐层状态和单元状态和三个门：遗忘门、输入门和输出门。LSTM网络由三个门来控制细胞的状态。

首先，LSTM的第一步就是决定细胞状态需要丢弃哪些信息。这部分操作是通过一个称为忘记门的sigmoid单元来处理的。它通过查看h\_{t-1}和x\_{t}信息来输出一个0-1之间的向量，该向量里面的0-1值表示细胞状态C\_{t-1}中的哪些信息保留或丢弃多少。0表示不保留，1表示都保留。遗忘门公式如公式2.2所示。

下一步是决定给细胞状态添加哪些新的信息。这一步又分为两个步骤，首先，利用和通过一个称为输入门的操作来决定更新哪些信息。然后利用和通过一个tanh层得到新的候选细胞信息，这些信息可能会被更新到细胞信息中。这两步描述如公式2.3和2.4所示。

下面将更新旧的细胞信息C\_{t-1}，变为新的细胞信息C\_{t}。更新的规则就是通过忘记门选择忘记旧细胞信息的一部分，通过输入门选择添加候选细胞信息\tilde C\_{t}的一部分得到新的细胞信息C\_{t}。更新操作如公式2.5所示。

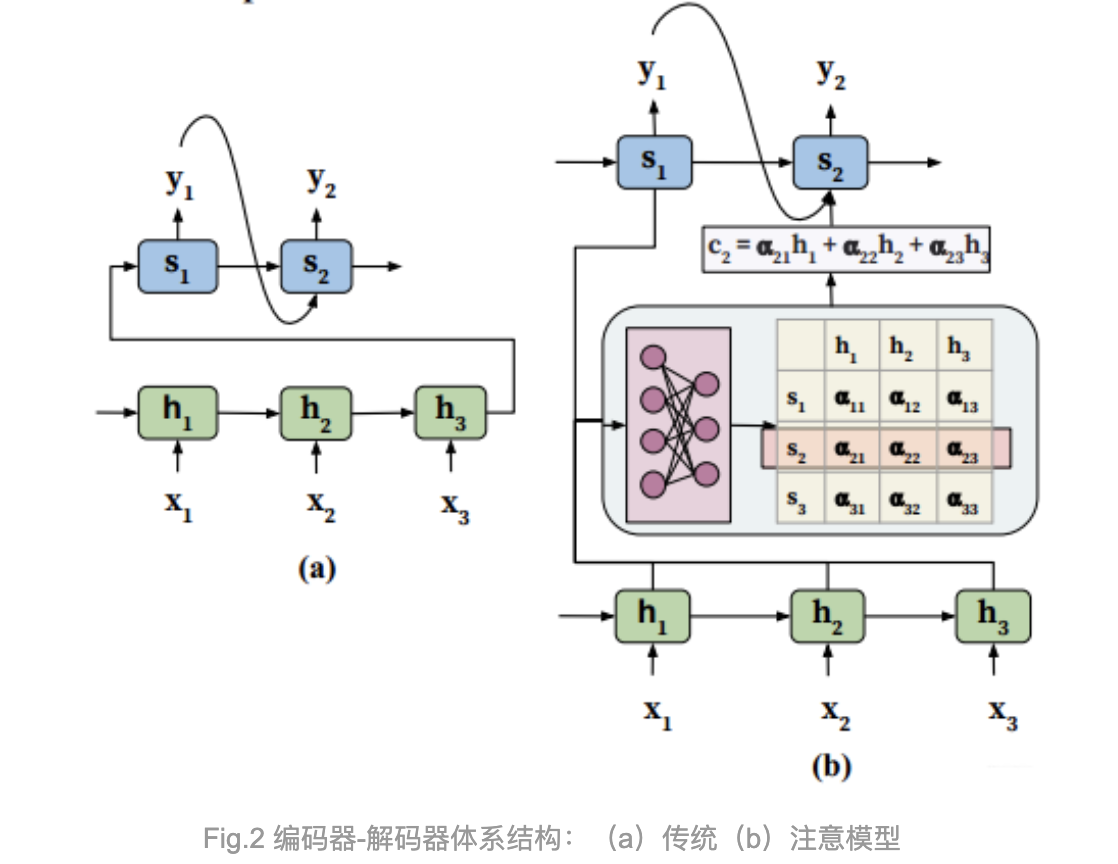
更新完细胞状态后需要根据输入的和来判断输出细胞的哪些状态特征，这里需要将输入经过一个称为输出门的sigmoid层得到判断条件，然后将细胞状态经过tanh层得到一个-1~1之间值的向量，该向量与输出门得到的判断条件相乘就得到了最终该RNN单元的输出。该步骤如公式2.6及2.7所示

## 2.2 注意力机制

论文三级标题：

黑体，小四号，段前0.5行，段后0.5行，单倍行距，居左，编号用Times New Roman。

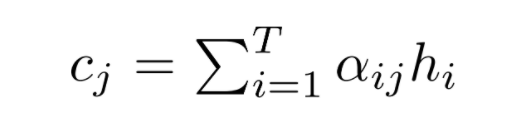
注意力机制（AM）最初被用于机器翻译，现在已成为深度学习领域的一个重要概念，并在自然语言处理领域有着大量的应用。深度学习中的注意力机制从本质上讲和人类的选择性视觉注意力机制类似，核心目标也是从众多信息中选择出对当前任务目标更关键的信息。



注意力机制最初依附于RNN的Encoder-Decoder框架下，但Encoder-Decoder框架如图2（a）所示有两大弊端。首先，编码器必须将所有输入信息压缩成一个固定长度的向量ht，然后将其传递给解码器。使用一个固定长度的向量压缩长而详细的输入序列可能会导致信息丢失 [Cho et al., 2014a]。其次，它无法对输入和输出序列之间的对齐进行建模，导致其缺乏任何机制在生成每个输出tokens时选择性地关注相关的输入tokens，而这又恰是结构化输出任务（如翻译或汇总）的一个重要方面[Young et al., 2018]。

注意力机制旨在通过允许解码器访问整个编码的输入序列（h1，h2，…，ht）来减轻Encoder-Decoder框架的不便。其核心思想是在输入序列上引入注意权重α，以优先考虑存在相关信息的位置集，以生成下一个输出token。

具有注意模型的相应编码器-解码器体系结构如图2（b）所示。网络结构中的注意力模块负责自动学习注意力权重αij，它可以自动捕获hi（编码器隐藏状态，我们称之为候选状态）和sj（解码器隐藏状态，我们称之为查询状态）之间的相关性。然后，这些注意力权重用于构建内容向量C，该向量作为输入传递给解码器。在每个解码位置j，内容向量cj是编码器所有隐藏状态及其相应注意权的加权和。



注意力权重是通过在体系结构中加入一个额外的前馈神经网络来学习的。该前馈网络学习一个特殊的注意权αij，作为两种状态h(i)和s(j-1)的函数。简单地说，就是用这两种状态作为神经网络的输入，进而学习αij关于他们的函数。

## 2.3 Dropout

论文三级标题：

黑体，小四号，段前0.5行，段后0.5行，单倍行距，居左，编号用Times New Roman。

在机器学习的模型中，如果模型的参数太多，而训练样本又太少，训练出来的模型很容易产生过拟合的现象。Hinton在2012年《Improving neural networks by preventing co-adaptation of feature detectors》中提出Dropout，即在每个训练批次中，通过忽略一半的特征检测器，来减少特征检测器（隐层节点）间的相互作用，这里检测器相互作用是指某些检测器依赖其他检测器才能发挥作用，从而明显减少过拟合现象。

Dropout本质上是指在前向传播的时候，让某个神经元的激活值以一定的概率p停止工作，这样可以使模型泛化性更强。如图2.1所示。

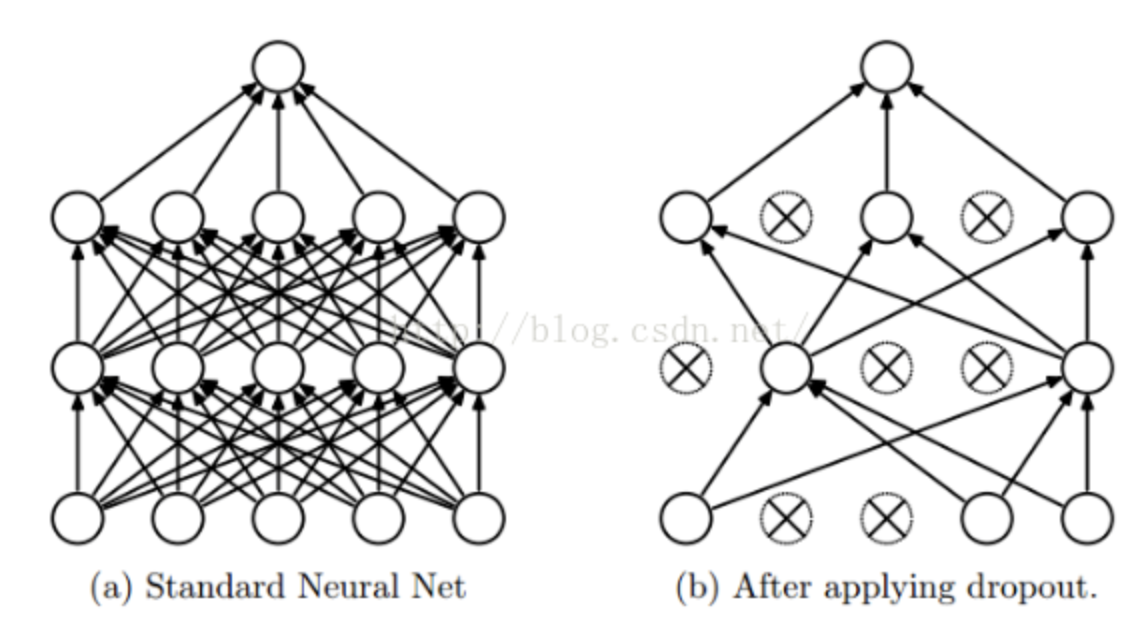


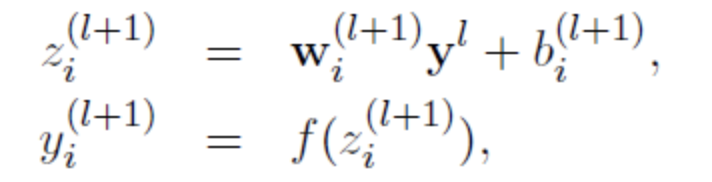
图2.1使用Dropout的神经网络模型（参考论文）

Dropout在神经网络上实现需要在训练模型阶段和测试模型阶段增加如下操作：

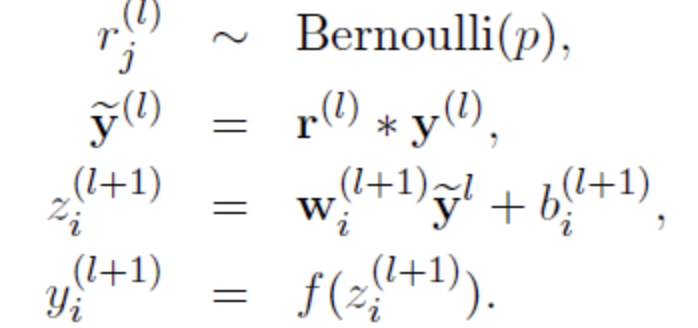
(1)在训练模型阶段

无可避免的，在训练网络的每个单元都要添加一道概率流程。对应的公式改变如下：

没有Dropout的网络计算公式：



采用Dropout的网络计算公式：



上面公式中Bernoulli函数是为了生成概率r向量，也就是随机生成一个0、1的向量。

(2)在测试模型阶段

预测模型的时候，每一个神经单元的权重参数要乘以概率p。

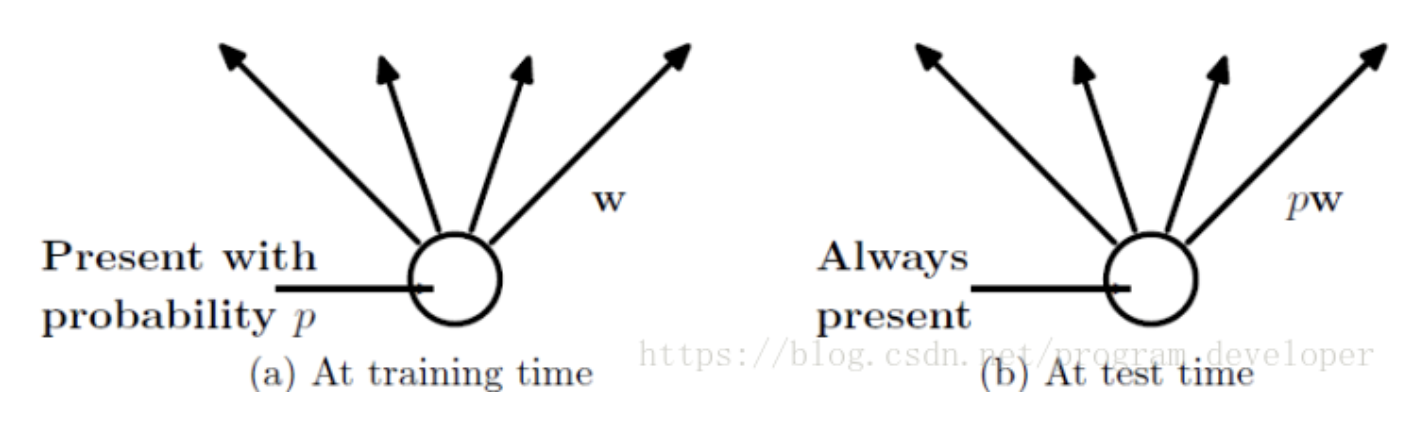
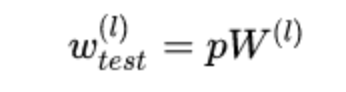


图2.2预测模型时Dropout的操作（参考论文）

测试阶段Dropout公式：



## 2.4 对象级情感分析的常用评估指标

论文三级标题：

黑体，小四号，段前0.5行，段后0.5行，单倍行距，居左，编号用Times New Roman。

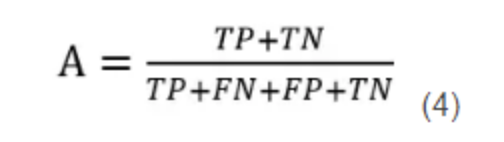
对象级情感分类本质上是一个二分类或多分类问题，常用的评估指标和分类问题的评估指标相同，主要包含精准率，召回率，查准率，F1-Score，Micro-F1和Macro-F1。

对于二分类问题，可将样例根据其真实类别与学习器预测类别的组合划分为真正例(true positive)、假正例(false positive)、真反例(true negative)、假反例(false negative)四种情形，令TP、FP、TN、FN分别表示其对应的样例数，则显然有TP+FP+TN+FN=样例总数。分类结果的“混淆矩阵”(confusion matrix)如表2.1所示。

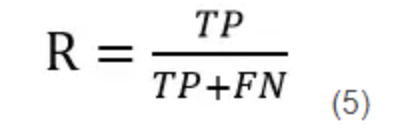
表2.1 分类结果混淆矩阵

|  |  |  |
| --- | --- | --- |
| **真实情况** | **预测情况** | |
| 正例 | 反例 |
| 正例 | TP（真正例） | FN（假反例） |
| 反例 | FP（假正例） | TN（真反例） |

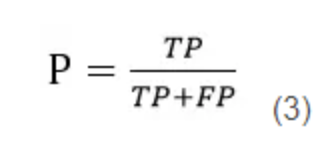
精确度(Accuracy)，缩写表示为A。精确度则是分类正确的样本数占样本总数的比例。Accuracy反应了分类器对整个样本的判定能力(即能将正的判定为正的，负的判定为负的)。定义见公式（2.1）。



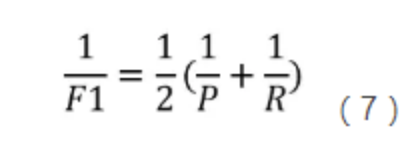
召回率(Recall)，又称查全率，缩写表示为R。查全率是针对我们原来的样本而言的，它表示的是样本中的正例有多少被预测正确。定义见公式（2.1）。



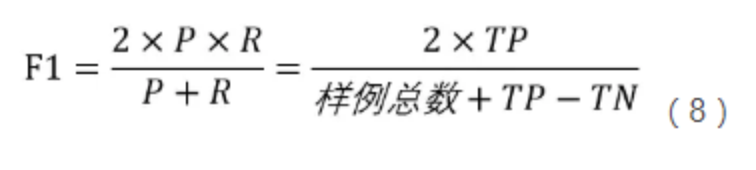
查准率(Precision)，又叫准确率，缩写表示用P。查准率是针对我们预测结果而言的，它表示的是预测为正的样例中有多少是真正的正样例。定义见公式（2.1）。



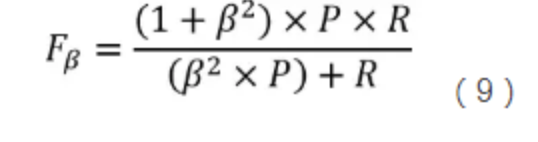
F1-Score是基于查准率与查全率的调和平均定义的，定义见公式（2.1）。



上述公式求倒数，即得F1-Score公式。



为适应不同实验对查准率与查全率的重视程度的不同，可将F1-Score公式变换为公式（2.1）。



其中，β>0度量了查全率对查准率的相对重要性。β=1时，退化为标准的F1；β>1时查全率有更大影响；β<1时，查准率有更大影响。

将只适用于二分类的F1-Score公式推广至多分类问题，就引出了Micro-F1和Macro-F1。统计各个类标的TP、FP、FN、TN，加和构成新的TP、FP、FN、TN，然后即可计算Micro-Precision和Micro-Recall，得到Micro-F1。

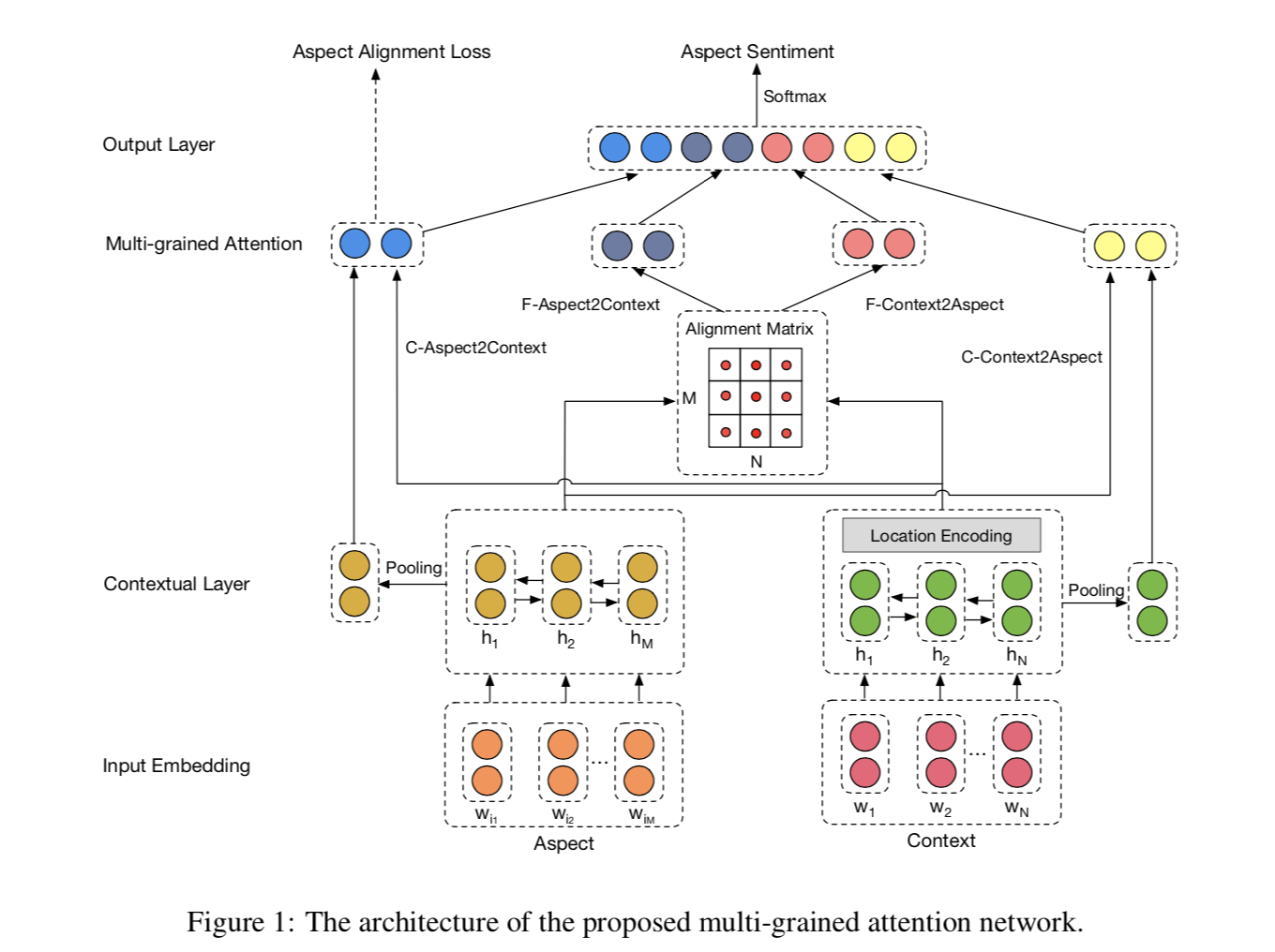
统计各个类标的TP、FP、FN、TN，分别计算各自的Precision和Recall，得到各自的F1值，然后取平均值得到Macro-F1

第3章　算法的分析与设计

## 3.1 算法定义

给定一个由N个单词组成的句子 和一个情感对象列表，其中情感对象列表的大小为k，每个方面是句子s的子序列，包含个单词。对象级情感分析针对每个方面评估句子的情感极性。例如：句子“I like coming back to Mac OS but this laptop is lacking in speaker quality compared to my $400 old HP laptop”对“Mac OS”对象的情感极性为积极的，对“speaker quality”对象的情感极性为消极的。

对上下文单词进行注意力加权时，仅仅对其进行简单平均向量可能会丢失信息。类似地，当强制模型对上下文词给予不同关注时，情感对象的简单平均向量（即“speaker quality”）可能会偏离直觉的核心含义（即“quality”）。 从另一个角度来看，以前的研究都将对象词及其上下文词视为一个实例，并分别训练每个实例。但是，他们没有考虑具有相同上下文字词的实例之间的关系。具有相同上下文的实例之间的方面级别交互可能会带来额外的有用信息。考虑到以上示例，从直觉上讲，与“Mac OS”相比，“speaker quality”方面应更多地关注“lacking”，而对“like”则较少关注，因为它们具有不同的情感极性。



本文提出了多注意力粒度交互网络（见图3.1）来解决对象级情感分类的上述两个问题。具体而言，我们提出了一种细粒度的注意力机制，该机制用于刻画情感对象词与上下文词之间的词级交互，并缓解在粗粒度注意力机制中发生的信息丢失。此外，我们利用双向粗粒度注意力，并将它们与细粒度的注意力相结合，以构成用于最终情感极性预测的多注意力粒度交互网络，从而可以同时利用它们两者的优势。与此同时，为了利用有价值的情感对象层次的交互信息，我们在目标函数中设计了方面对齐损失，以增强针对具有相同上下文和不同情感极性的情感对象的注意权重的差异。

## 3.2 嵌入层

嵌入层将每个单词映射到高维向量空间。 我们采用Pennington等人【】提出的基于全局词汇共现的统计信息来学习词向量的词表征工具来获得每个词的固定词嵌入。

GloVe 的实现可以分为以下三步:

(1)根据语料库构建共现矩阵X，其元素为Xij。Xij的意义为，在整个语料库中，单词i和单词j共同出现在特定大小的上下文窗口中的次数。GloVe根据单词i和单词j在上下文窗口中的距离𝑑，计算出衰减函数decay = 1⁄𝑑代表权重。

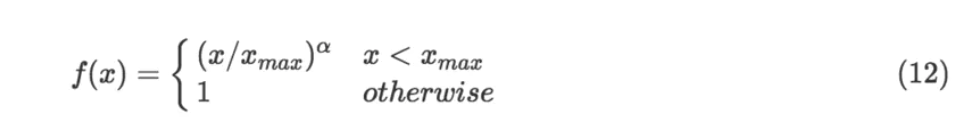
(2)用公式3.1表达构建词向量和共现矩阵之前的近似关系:

(3.1)

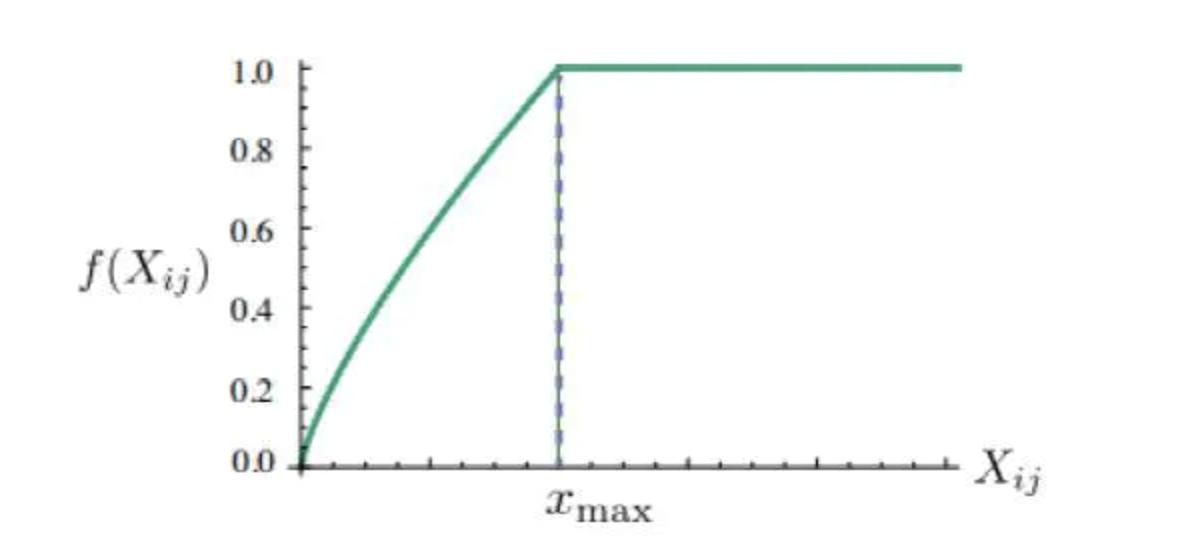
其中，和是最终要得到词向量，和是两个词向量所对应的偏置项。

(3)根据公式3.1可以构造得到如公式3.2所示的代价函数J

其中，和是偏置项, 是均方误差函数。为权重系数，且需要满足以下三个条件,为单调不减函数，在值很大的时应该相对较小，即的表达式确定为如公式3.3所示。



当时，的图像如图3.3所示：



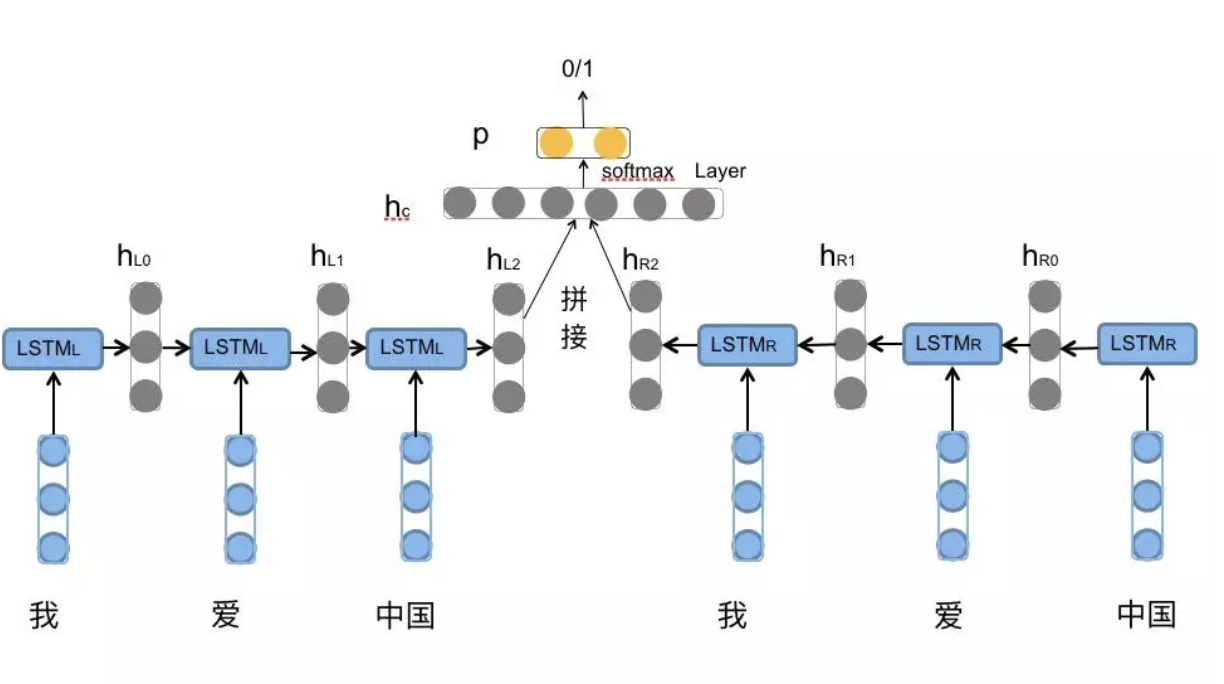
最终，本文将嵌入查找矩阵表示为L∈Rdv×| V |。 ，其中dv是词向量维，| V | 是词汇量。

## 3.3 上下文层

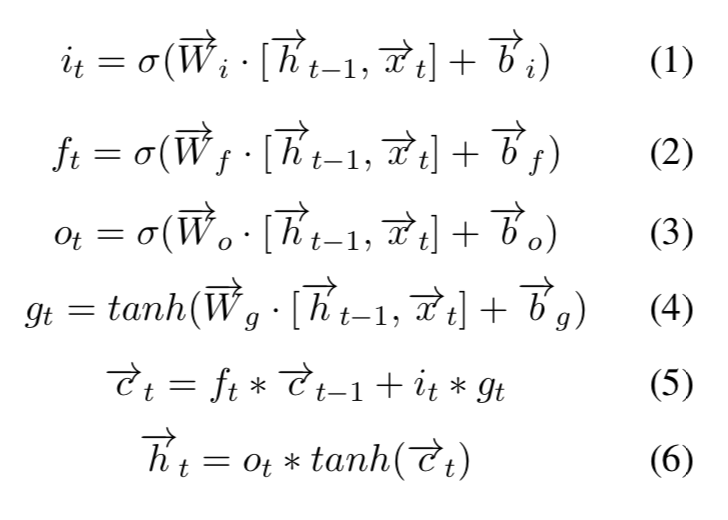
将词的表示组合成句子的表示，可以采用相加的方法，即将所有词的表示进行加和，或者取平均等方法，但是这些方法没有考虑到词语在句子中前后顺序。如句子“我不觉得菜好吃”。“不”字是对后面“好吃”的否定，即该句子的情感极性是贬义。LSTM模型因其通过训练过程可以选择性的学习输入的信息，可以更好的捕捉到较长距离的依赖关系。

但由于LSTM对句子进行建模时无法编码从后到前的信息，在情感对象之后的语句无法被LSTM网络所捕获。例如，“这个酒店服务差得不行，没有隔壁好”，这里的“不行”是对“差”的程度的一种修饰，通过双向长期短期记忆网络（BiLSTM）可以更好的捕捉双向的语义依赖。

因此本文在嵌入层之上采用如图3.4所示双向长期短期记忆网络（BiLSTM），以更好捕获单词之间的交互作用。

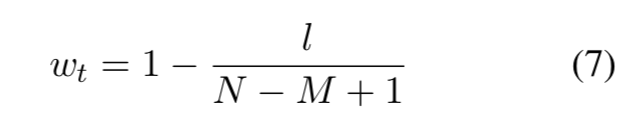


具体来说，在时间步t，给定输入词嵌入x，正向LSTM网络的更新过程可以如下形式化：



其中σ是sigmoid激活函数，ft和ot分别是输入门，忘记门和输出门。，，并且是隐藏维度大小。反向的LSTM执行类似的过程，我们可以得到级联的输出。给定上下文句子的词嵌入和相应的情感对象，我们将分别使用BiLSTM并获得句子上下文输出和情感对象上下文输出。

另外，考虑到距某个情感较近的上下文词可能会对该方面产生更大的影响，因此我们利用位置编码机制来模拟观察。形式上，和情感对象距离为个单词距离（此处将情感对象短语视作一个单词距离），上下文词的权重定义如公式3.7所示。



具体来说，我们将方面中单词的权重视为0，以便关注句子中的上下文单词。然后我们可以获得上下文词的最终上下文输出

## 3.4 多注意力粒度层

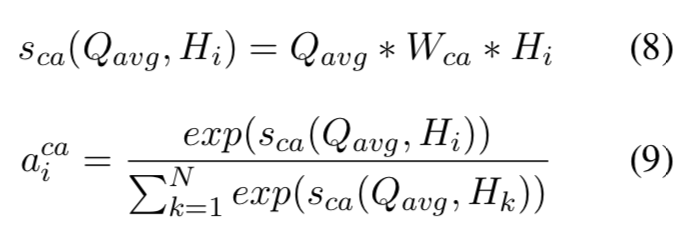
本文提出的多注意力粒度需要从粗粒度和细粒度两个维度以及情感对象对上下文和上下文对情感对象两个方向利用注意力机制生成有交互的情感对象向量或上下文向量。通过这种方式，在较长的情感对象词或上下文词中仍可以有效获取句子中情感对象词和上下文词的有效交互信息，提高情感分析结果的准确度。

3.4.1 粗粒度注意力机制

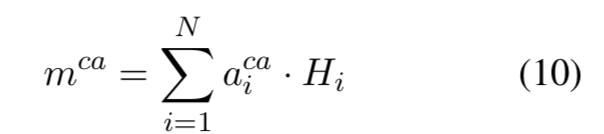
粗粒度注意力机制是捕获情感对象和上下文之间相互作用的一种广泛使用的机制，该机制利用平均情感向量来引导上下文单词上的注意力权重。 根据Ma等人【】中的工作，本文采用了双向注意机制，即粗粒度-情感对象与上下文词注意力机制和粗粒度-上下文词与情感对象注意力机制。

(1) 粗粒度-情感对象与上下文词注意力机制

粗粒度-情感对象与上下文词注意力机制根据情感对象的平均向量为每个上下文单词分配注意力权重。在这里，本文在情感对象上下文输出Q之上使用平均池化层，以生成情感对象的平均向量。 对于上下文中的每个单词向量，可以根据公式3.1计算其注意力权重：

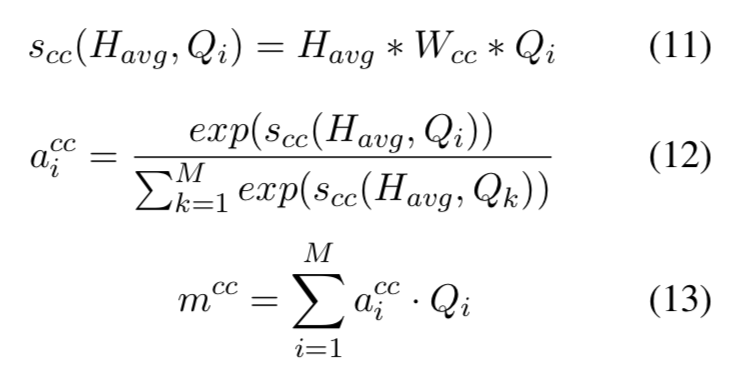


其中，函数计算权重，该权重代表上下文词对情感对象词的影响力程度。是注意力权重矩阵。然后，上下文粗输出的加权组合计算如下：



(2) 粗粒度-上下文词与情感对象注意力机制

粗粒度-上下文词与情感对象注意力机制根据上下文词的平均向量分配每个情感对象单词的注意力权重，这遵循粗粒度-情感对象与上下文词注意力机制类似的学习过程。 本文利用平均池化机制来获取上下文词的平均向量，并计算情感对象短语中每个单词的权重。计算情感对象向量粗输出的最终加权组合计算如下：

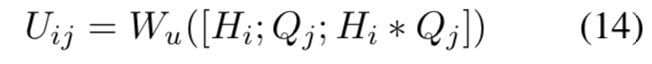


其中，是注意力权重矩阵。

3.4.2 细粒度注意力机制

如上所述，本文提出了一种细粒度的注意机制来表征单词级交互，并评估每个方面/上下文单词如何影响每个上下文/方面单词。同粗粒度注意力机制相同，本文在细粒度注意力机制下同样采用了双向注意机制，即细粒度-情感对象与上下文词注意力机制和细粒度-上下文词与情感对象注意力机制。

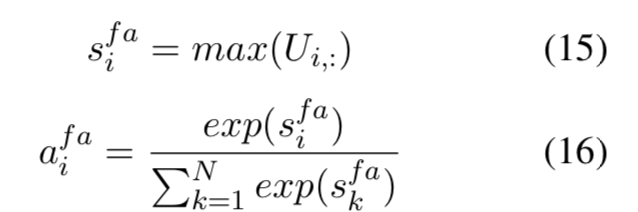
形式上，利用上下文层生成的上下文输出向量H和情感对象输出向量Q定义对齐矩阵，其中表示第个上下文词和第个情感对象词之间的相关性。对齐矩阵计算公式如公式3.5所示。



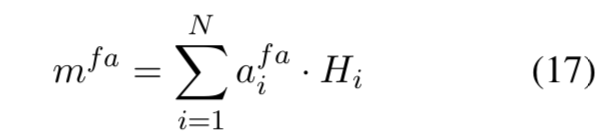
其中，是权重矩阵，是跨行的向量串联，\*是向量乘法。接下来使用对齐矩阵U来计算两个方向上的注意力向量。

(1) 细粒度-情感对象与上下文词注意力机制

细粒度-情感对象与上下文词注意力机制用于确定对于一个情感对象词最相关的上下文单词，因此对于确定情感对象的情感至关重要。利用下述的公式3.5和3.6即可计算得出每个上下文词对于特定的情感对象词的注意力权重。

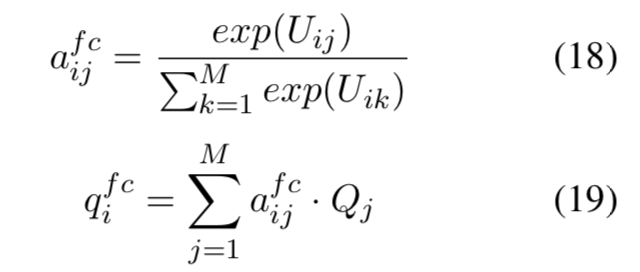


其中，是所有列的最大相关度。因此，利用下面的公式3.6可以得出如下上下文细输出。

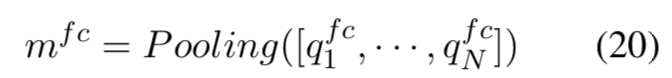


(2) 细粒度-上下文词与情感对象注意力机制

细粒度-上下文词与情感对象注意力机制用于确定对于一个上下文词最相关的情感对象单词。定义利用上下文层生成情感对象输出向量Q关于第个上下文单词的注意力权重，注意力权重可根据下面公式3.7和3.8得出。

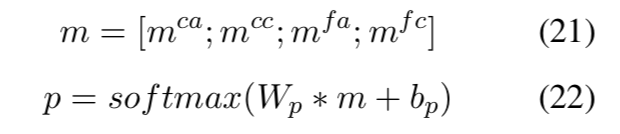


最后，在上使用平均池化层来获得情感对象细输出。



## 3.5 输出层

在最终的输出层将粗粒度和细粒度的注意力向量连接起来，作为最终表示，并将其馈送到softmax层以确定对应情感对象的情感极性。



其中，是情感对象的情感极性的概率分布，和分别是权重矩阵和偏置矩阵。在这里，我们将C 设置为3，即情感对象分类后最终情感类别的数量（正面、负面、中性）。

第4章　算法的实现与测试

## 4.1 算法实现

4.1.1 数据预处理

第一步：

将语料文件中的句子，按照固定格式来划分成如下格式：

A longer battery life would have been great - but it meets it 's $T$ quite easily .  
spec  
1

第一行放具体的句子，并将句子描述的具体的描述对象用$T$表示，并将描述对象放到第二行。在第三行，来标注这一句话的极性标签（0:中性，1:正向，-1:负向）

第二步：

把固定格式的句子，合成一整段长字符串，并将长字符串中所有单词取出，生成Tokenizer，此Tokenizer为这部分语料的所有单词的token集合，并将生成的Tokenizer存储在文件中，方便后续使用。关键代码如下：  
fin = open(fname, 'r', encoding='utf-8', newline='\n', errors='ignore')  
lines = fin.readlines()  
fin.close()  
for i in range(0, len(lines), 3):  
 text\_left, \_, text\_right = [s.lower().strip() for s in lines[i].partition("$T$")]  
 aspect = lines[i + 1].lower().strip()  
 text\_raw = text\_left + " " + aspect + " " + text\_right  
 text += text\_raw + " "

def fit\_on\_text(self, text):  
 if self.lower:  
 text = text.lower()  
 words = text.split()  
 for word in words:  
 if word not in self.word2idx:  
 self.word2idx[word] = self.idx  
 self.idx2word[self.idx] = word  
 self.idx += 1

第三步：将每个单词映射到多维空间

我们使用基于维基百科子集预训练的300维词Glove词向量。将我们获取到的每个Tokenizer对应的词向量取出，单独存放在一个文件中。关键代码如下：

embedding\_matrix = np.zeros((len(word2idx) + 2, embed\_dim))

# idx 0 and len(word2idx)+1 are all-zeros  
fname = './glove.twitter.27B/glove.twitter.27B.' + str(embed\_dim) + 'd.txt' \  
 if embed\_dim != 300 else './glove.42B.300d.txt'  
word\_vec = \_load\_word\_vec(fname, word2idx=word2idx)  
print('building embedding\_matrix:', dat\_fname)  
for word, i in word2idx.items():  
 vec = word\_vec.get(word)  
 if vec is not None:  
 # words not found in embedding index will be all-zeros.  
 embedding\_matrix[i] = vec  
pickle.dump(embedding\_matrix, open(dat\_fname, 'wb'))

4.1.2 模型训练

(1)模型的基本结构

模型的基本结构如下：

MGAN(

(embed): Embedding(3600, 300)

(ctx\_lstm): DynamicLSTM(

(RNN): LSTM(300, 300, batch\_first=True, bidirectional=True)

)

(asp\_lstm): DynamicLSTM(

(RNN): LSTM(300, 300, batch\_first=True, bidirectional=True)

)

(location): LocationEncoding()

(alignment): AlignmentMatrix()

(dense): Linear(in\_features=2400, out\_features=3, bias=True)

)

模型包括最外层的MGAN模型以及MGAN中使用到的双向LSTM模型，这里叫做ctx\_lstm(Context LSTM) 和asp\_lstm(Asspect LSTM)。这里需要增加分析

(2).模型的训练过程

在基本数据处理好之后，我们使用Torch.Module作为父类来实现模型。

Module可以创建一个可调用的，其表现类似于函数，但又可以包含状态(比如神经网络层的权重）的对象。该对象知道它包含的Parameter(s），并可以将梯度置为0，以及对梯度进行循环以更新权重等。

Parameter：是一个对张量的封装，它告诉Module在反向传播阶段更新权重。只有设置了requires\_grad属性的张量会被更新。

functional：一个包含了梯度函数、损失函数等以及一些无状态的层，如卷积层和线性层的模块(通常使用F作为导入的别名）。

需要添加对具体模型与设计对照的地方

## 4.2 算法评测

4.2.1 实验环境

针对以上约束条件以及目标函数设计编码序列模型, 在Python3.7 环境下, 训练MGAN模型，运行环境是Pycharm CE，512MB，MAC OS。

使用相关工具类包括：

numpy>=1.13.3  
 torch>=0.4.0  
 transformers==1.2.0

4.2.2 实验数据

本文在三个数据集上进行了实验，如表4.1所示。前两个数据来自SemEval 2014 Task 4，其中分别包含笔记本电脑和餐厅的评论。 第三个是由Dong等人收集的推特集合。

表4.1 数据集统计表

|  |  |  |
| --- | --- | --- |
| **数据集** | **训练集大小** | **测试集大小** |
| 笔记本电脑 | 2328条 | 638条 |
| 餐厅 | 3608条 | 1120条 |
| 推特 | 6248条 | 692条 |

与上下文相关的每个情感对象的情感极性有三种状态：正面，中性和负面。使用[1,0,-1]这三个值对情感极性进行描述，“1”表示正面，“0”表示中性，“-1”表示负面。情感极性的含义对照如图表4.2所示。

表4.2 情感极性及其含义对照表

|  |  |  |  |
| --- | --- | --- | --- |
| **情感极性** | **1** | **0** | **-1** |
| 含义 | 正面 | 中性 | 负面 |

本文通过对部分数据集手动状态标注，设置成对应的test\_set并通过test\_set训练出的模型，来对整个数据集进行处理，以此来计算整个模型的识别率。并通过控制不断提高模型的epoch，可以不断的提高整个模型识别的准确率，第一次训练得到的结果如图4-1(左)所示，平均准确率为0.21，在第10次训练之后，大约为0.60结果如图4-1(右)所示。

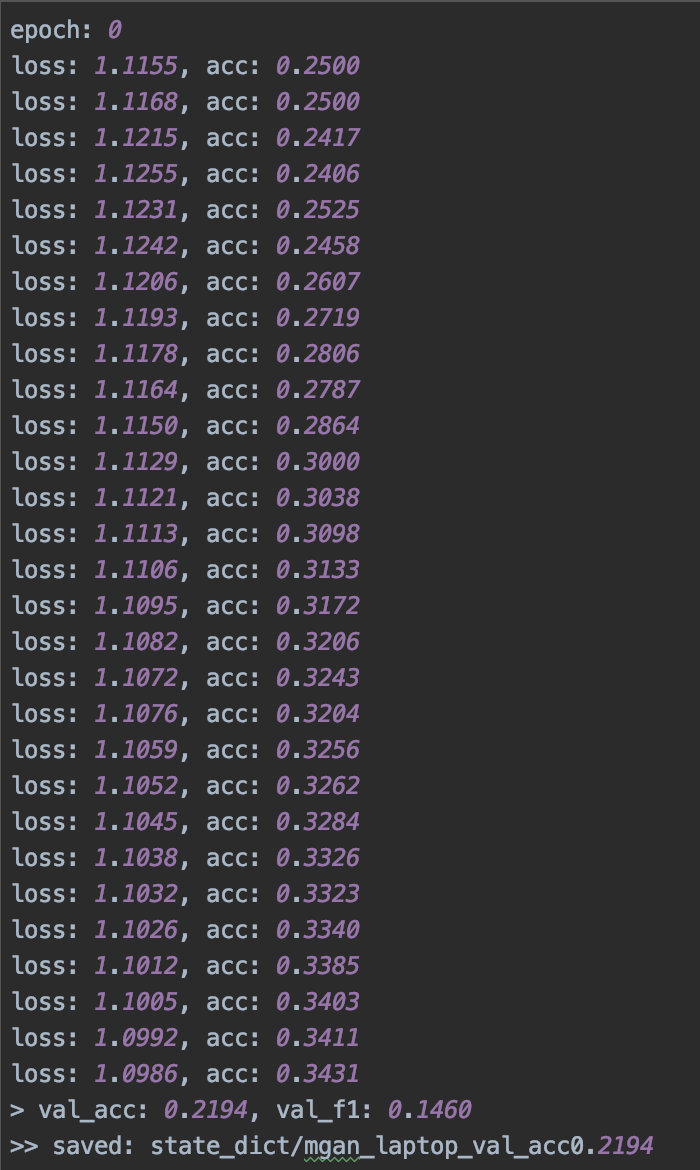
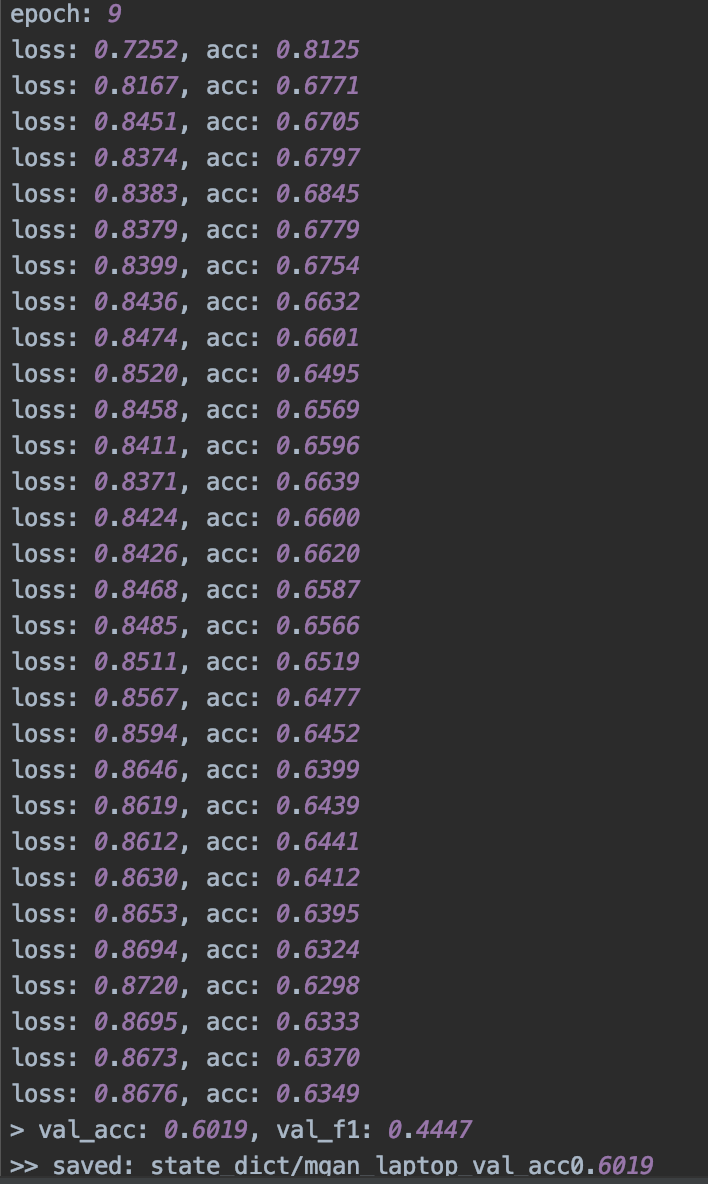
 

图4-1 第一次训练结果和第10次训练结果

4.2.3 实验结果与分析

## 4.3 实验结论

第5章　总结与展望

## 5.1 总结

## 5.2 展望

用学院下

参考文献

[1] Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In IJCAI, pages 1347–1353.

[2] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter senti- ment classification. In Proceedings of the 49th An- nual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 151–160. Association for Computational Linguistics.

[3] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. Nrc-canada-2014: Detect- ing aspects and sentiment in customer reviews. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 437– 442.

[4] 张志祥. 间断动力系统的随机扰动及其在守恒律方程中的应用[D]. 北京: 北京大学, 1998.

[5] World Health Organization. Factors regulating the immune response: report of WHO Scientific Group[R]. Geneva: WHO, 1970.

[6] 河北绿洲生态环境科技有限公司. 一种荒漠化地区生态植被综合培育种植方法:中国, 01129210.5[P]. 2001-10-24.

[7] GB/T16159-1996, 汉语拼音证词法基本规则[S]. 北京: 中国标准出版社, 1996.

[8] 毛侠. 情感工学破解“舒服之谜”[N]. 光明日报, 2004-04-17(B1).

[9] 陈剑. 上博简《民之父母》“而得既塞於四海矣”句解释[EB/OL]. 简帛研究网站，http://www.bamboosilk.org/Wssf/2003/chenjian03.htm．2003-01-18.

致 谢

应以简短的文字对题目与论文撰写过程中曾直接给予帮助的人员(例如指导教师、答疑教师及其他人员)表示自已的谢意 ，这不仅是一种礼貌，也是对他人劳动的尊重，是治学者应有的思想作风。对整个毕业设计（论文）进行总体性、概括性总结，表达出设计（论文）的思路、学习收获、对未来进一步学习的设