北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

情感分析汇报

汇报人：莫璐雅 彭书蕾 黄彦宸 管树言 赵旻基

导　师：张华平

时　间：2022/3/25

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

目录

CONTENTS

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

概述

莫璐雅

概述——基本概念

情感分析又称意见挖掘、倾向性分析。

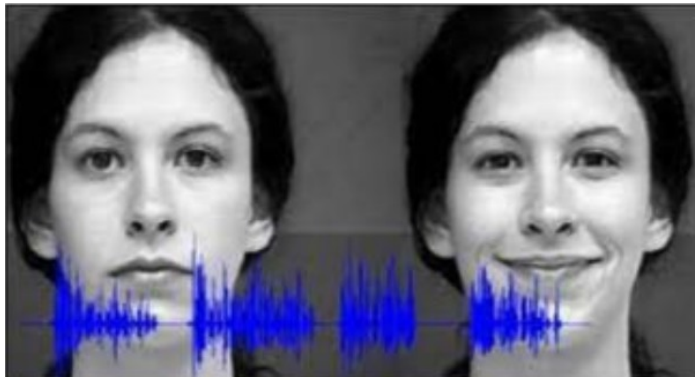
对象：

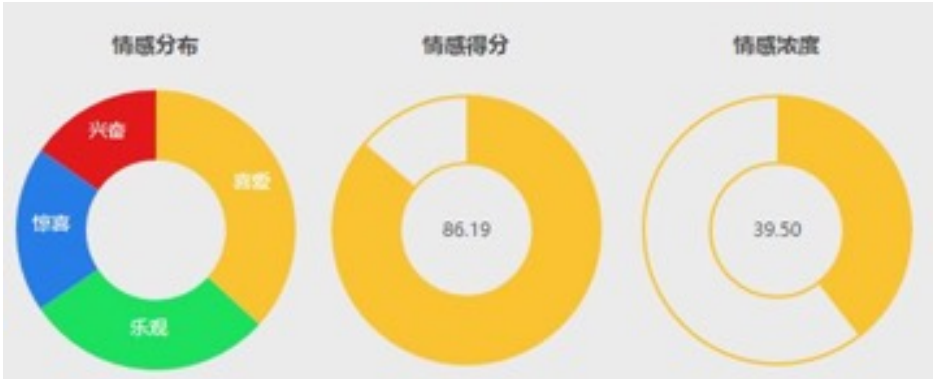
带有情感色彩的信息

目标：

挖掘其对目标的观点

这是国产惊悚片的诚意之作





概述——产生背景

大量以数字格式记录的观点文本

- **交互式网络**

  读网络-写网络

- **社交媒体**

  信息发布平台

- **大数据**

  数据爆炸增长

概述——应用场景

舆情分析

通过对热点事件进行情感剖析，寻找情感原因，对政府了解民意，预防危害事件的发生具有一定的意义

在线评论

对评价对象和表达进行抽取，识别评论中的情感倾向性，对消费者挑选商品，商家改进商品/服务具有辅助作用

商业投资

通过对投资者情绪的分析，预测股票市场、股票价格的发展趋势

其他

情感对话，个性化系统推荐

概述——对象分类

情感分析

根据所处理的对象

- **单模态**
  - 文字
  - 音频
  - 图像
- **多模态**
  - 结合多个模态的数据并将其统一建模
  - 更多的信息来源、更优的决策。

Utterance: *"Become a drama critic!"*

Emotion: *Joy* **Sentiment:** *Positive*

Text	Audio	Visual
Ambiguous	Joyous tone	Smiling Face

Utterance: *"Great, now he is waving back"*

Emotion: *Disgust* **Sentiment:** *Negative*

Text	Audio	Visual
Positive/Joy	Flat tone	Frown

概述——级别分类

文本情感分析

根据所处理文本的粒度

篇章级

句子级

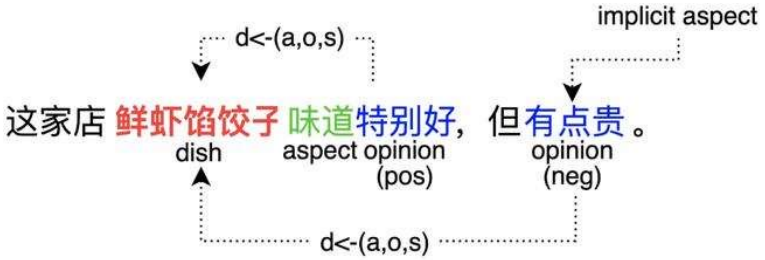
整体

假设文本只对一个实体进行了评论，与实际情况不符

属性级/细粒度

实体对象或其属性

目前大部分业界的情感计算系统



概述——数据集及评价指标

数据集	文本粒度	介绍
IMDb	篇章级	由电影评论构成，包含篇章级的1 000篇褒义评论和1 000 篇贬义评论，句子级的5 331 个褒义评论。
SST-2	细粒度	在电影评论中的11,855个句子的语法分析树中包含215,154个带有细粒度情感标签的短语。
Sentihood	细粒度	识别针对特定方面的细粒度极性。 数据集包含5,215个句子，其中3,862个包含单个目标，其余多个目标。
SemEval-ABSA	细粒度	给出某个topic下的推特数据集，推断推特内容关于特定topic的积极或消极情感倾向
weibo_senti_100k	篇章级	包含约12 万条新浪微博，正负向约各6 万条
外卖评论数据集	篇章级	包含某外卖平台正向4 000 条评论和负向8 000 条评论

概述——数据集及评价指标

- **精确度Precision——**度量正确性

  较高的精确度意味着更少的误报，而较低精度意味着更多的误报。

- **召回率Recall——**度量完整性/灵敏度

  较高的召回意味着更少的假负，而较低的召回意味着更多的假负。

- **F值F-measure Metric**

  精确度和召回率的加权调和平均数。

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times P \times R}{P + R}$$

$$Acc = \frac{TP + TN}{TP + TN + FN + FP}$$

概述——基本流程

1 → 2 → 3

原始数据获取

一般通过爬虫、关键词检索获取

文本预处理

根据任务需要进行文本清理、标准化、纠错、改写等

特征提取

将输入数据转换为向量表征，如one-hot，TF-IDF，Embedding

3 → 4 → 5

情感分类器

根据输入的特征向量计算结果

结果输出

输出情感类别与得分

北京理工大學 BEIJING INSTITUTE OF TECHNOLOGY

词典与规则

- 基于关键词
- Bag of words & Syntactic Rules（Turney 2002）
- SentiWordNet(2006)

- 20世纪90年代末国外情感分析起步

研究发展-情感词典

Bag of words(词袋)

词频

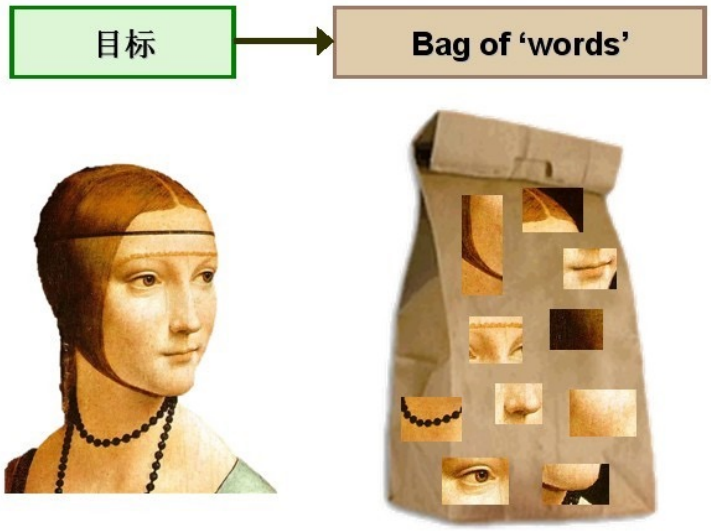
情感词组合、否定/程度词等问题

词典&规则

词序

- SentiWordNet
  ——国外最早的情感词典
- 知网情感词典

……



基于关键词

- 符号化表示
- 人工构建知识

概述——研究发展

词典与规则

- 基于关键词
- Bag of words & Syntactic Rules（Turney 2002）
- SentiWordNet(2006)

机器学习

- 基于频率
- SVMs（Moraes，2013）
- Naive Bayes分类器（Tan，2009）

- 20世纪90年代末国外情感分析起步

- 2000年以来，在线的主观信息快速增长，情感分析研究逐渐活跃
- 2008，大数据时代，SA进入上升期

研究发展-机器学习

特征提取：

OneHot、TF-IDF 、Ngram

分类器：

KNN、SVM、logistic Regression

- 句子级研究

我爱中国 -> 1, 1, 0, 0, 1

爸爸妈妈爱我 -> 1, 1, 1, 1, 0

The **cat** is walking in the bedroom.

The **dog** is walking in the bedroom. ?

基于频率

- 大规模语料
- 短上下文
- 词孤立

概述——研究发展

词典与规则

- 基于关键词
- Bag of words & Syntactic Rules（Turney 2002）
- SentiWordNet(2006)

机器学习

- 基于频率
- SVMs（Moraes，2013）
- Naive Bayes分类器（Tan，2009）

深度学习

- 基于上下文
- RNTN（Socher，2013）
- CNN，Dynamic CNN（Kim，2014）
- Attention mechanisms（Wang，2016）

- 20世纪90年代末国外情感分析起步

- 2000年以来，在线的主观信息快速增长，情感分析研究逐渐活跃
- 2008，大数据时代，SA进入上升期

- 词向量出现，深度学习方法开始普及
- 2013 年，Google 发布了Word2Vec

研究发展-深度学习

特征提取：

稠密的word embedding

Word2vec

基于上下文

捕捉语境信息

&

压缩数据规模

可计算、可训练

- 数字指标提高
- 业务应用不佳

神经网络：

CNN、RNN、Attention、LSTM

- 篇章级研究

概述——研究发展

词典与规则

- 基于关键词
- Bag of words & Syntactic Rules（Turney 2002）
- SentiWordNet(2006)

机器学习

- 基于频率
- SVMs（Moraes，2013）
- Naive Bayes分类器（Tan，2009）

深度学习

- 基于上下文
- RNTN（Socher，2013）
- CNN，Dynamic CNN（Kim，2014）
- Attention mechanisms（Wang，2016）

预训练+微调

- 基于大规模语料训练
- ELMo（Peters，2018）
- BERT（Devlin，2019）

- 20世纪90年代末国外情感分析起步

- 2000年以来，在线的主观信息快速增长，情感分析研究逐渐活跃
- 2008，大数据时代，SA进入上升期

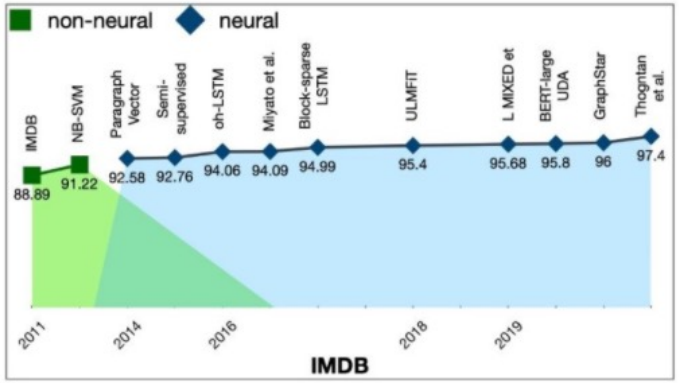
- 词向量出现，深度学习方法开始普及
- 2013年，Google发布了Word2Vec

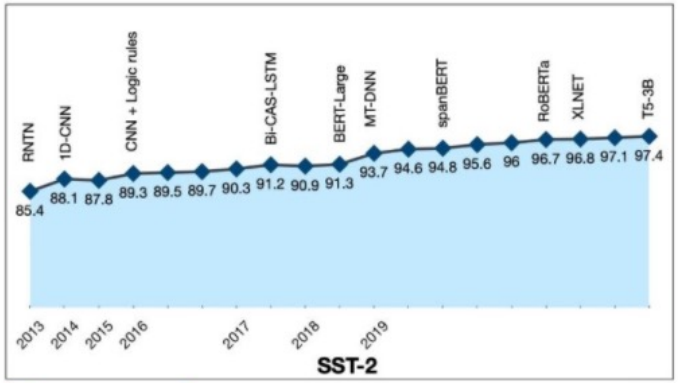
研究发展-预训练模型

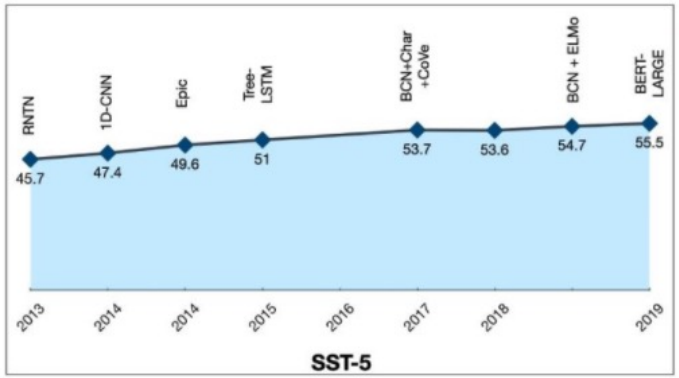
预训练模型——迁移学习　　Bert、ELMo

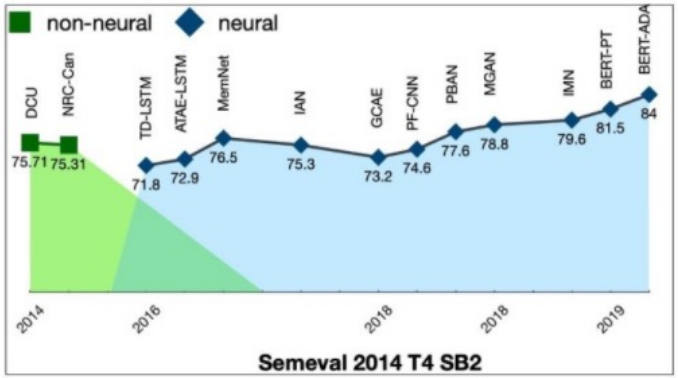
将大规模语料中的知识学习到模型中

基于大规模语料训练





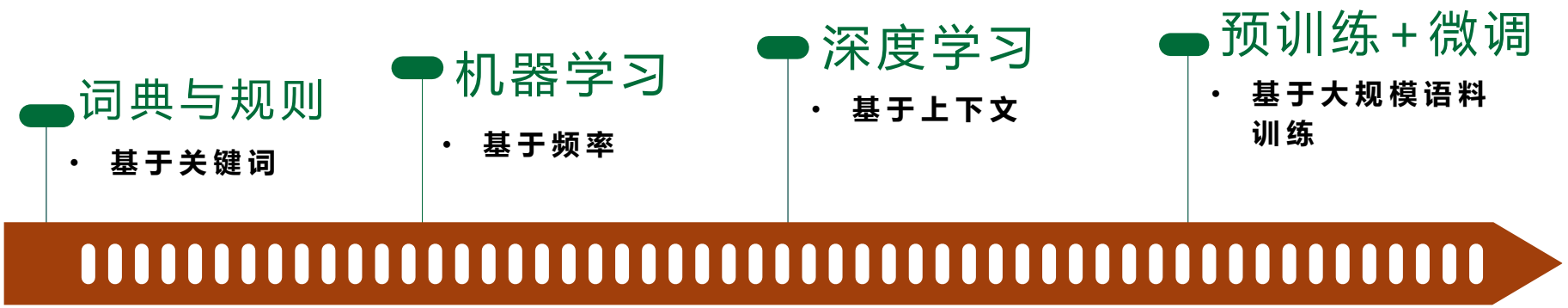




- 模型初始化
- 应用方面提升

- 属性级研究
- 多模态融合

研究发展-趋势

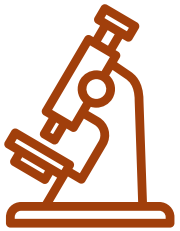


语义表示： 符号表示 ⟶ 分布表示

学习模式： 浅层学习 ⟶ 深度学习

语言知识： 人工构建 ⟶ 自动构建

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

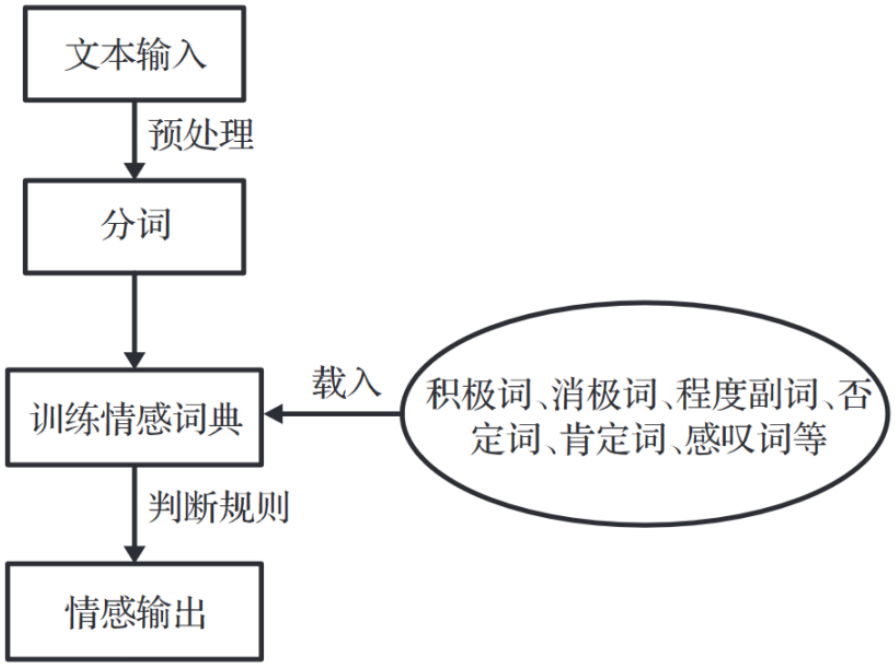


经典模型

彭书蕾

经典模型——基于情感词典与规则

利用情感词典获取文档中情感词的情感值，再通过判断规则确定文档的整体情感倾向。——基于关键词



经典模型——基于情感词典与规则

作者	基础词典	其他词典	规则	数据来源	情感极性分类效果		
					P/%	*R*/%	*F*1/%
董丽丽等[25]	HowNet	网络情感词、未登录情感词、否定词、程度副词、关联词	无	ZOL中的笔记本电脑评论	75.44	81.21	78.22
Asghar等[26]	SentiWordNet	表情符号、修饰语、否定词、领域术语	无	酒店评论数据	82.50	83.50	82.99
Han等[27]	SentiWordNet	领域情感词	无	IMDB数据集	76.96	76.81	76.87
李晨等[28]	HowNet、NTUSD、哈工大同义词词林	转折归总词、程度副词、否定词	无	新闻、博客和论坛数据	76.00	81.00	78.42
胡召亚等[29]	大连理工情感词汇本体库	表情符号	句型规则、句间关系规则	公开的微博情感分析语料	70.70	68.30	69.40
吴杰胜等[30]	HowNet、NTUSD、大连理工情感词汇本体库	领域情感词、否定词、双重否定词、程度副词、关系连词、表情符	句型规则、句间关系规则	与“短视频整顿”话题相关的微博文本	82.10	82.70	83.40
王志涛等[31]	HowNet、NTUSD	新词、修饰词表、表情符词表	句型规则、句间关系规则、表情符规则、词语多元组规则	新浪微博文本数据	68.30	67.10	67.70

经典模型——基于情感词典与规则

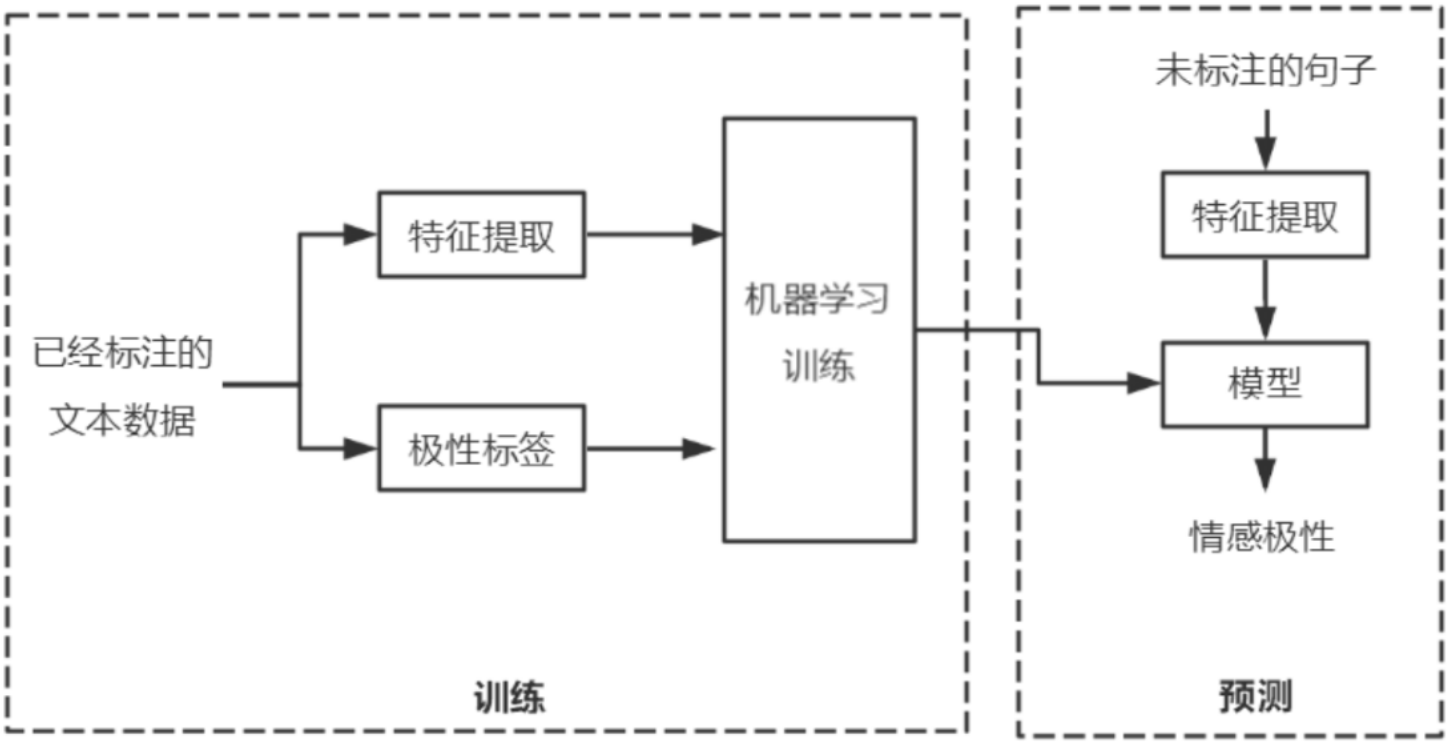
- 准确反映文本非结构化特征
- 方法简单，易于分析理解
- 在缺乏大量训练数据的情况下，也能达到一定的效果
- 当情感词覆盖率与准确率较高时，效果较好

- 基本依赖情感词典的构建
- 对于新词的识别效果较差，只能不断扩充情感词典
- 跨领域和跨语言效果不理想
- 无法考虑上下文语义关系

经典模型——基于传统机器学习

将情感分析问题视作一个文本分类任务
本质——基于词频

基于机器学习的情感分类方法主要分为三类：**有监督、 半监督**和**无监督**的方法。



有监督方法

朴素贝叶斯

朴素贝叶斯分类法是基于贝叶斯定理和特征条件独立假设的分类方法，它通过特征计算分类的概率，选取概率大的情况，是基于概率论的一种机器学习分类（监督学习）方法，被广泛应用于情感分类领域的分类器。

最大熵

最大熵分类器属于指数模型类的概率分类器。基于最大熵原理，并且从适合训练数据的所有模型中，选择具有最大熵的模型。近年部分学者基于最大熵构建情感分析模型，对文本情感进行了分析。

KNN

KNN分类算法的思路是：如果一个样本在特征空间中的k个最相似的样本中的大多数属于某一个类别，则该样本也属于这个类别。

支持向量机SVM

支持向量机通过寻求结构化风险最小以提高学习机泛化能力，实现经验风险和置信范围的最小化，从而达到在统计样本量较少的情况下，亦能获得良好统计规律的目的。

经典模型——基于传统机器学习

作者	模型	算法特点	数据来源	情感极性分类效果		
				P/%	*R*/%	*F*1/%
谢丽星等[38]	SVM	用层次结构,将情感分析过程分为两大策略、4种方法	新浪中的影视、名人和产品领域	67.28	–	–
刘宝芹等[39]	NB	建立三层树状情绪分类结构	不同话题的微博文本	70.60	65.30	67.80
Wawre等[40]	NB	对于大规模训练集,朴素贝叶斯方法更好	IMDB数据集	66.77	62.00	64.29
Kaur等[42]	KNN	将*N*-gram用于特征提取,特征提取与分类技术相结合	电子商务网站的评论	82.00	81.50	81.75
徐建忠等[43]	SVM	设计特征向量,采用有监督的机器学习算法进行分类	航天事件相关的微博文本	80.30	78.50	79.40
李锐等[44]	SVM	对词向量进行加权,解决文本特征稀疏的问题	公开的微博情感分析语料	89.35	89.35	89.35
Rathor等[45]	SVM	SVM的学习精度高	公开的Amazon评论数据集	81.20	–	–

基于SVM的文本情感分析方法被认为是最好的情感分析方法，泛化错误率低，计算开销不大，而且对于训练样本较小的文本可以得到很好的情感分析效果，对高维数据的处理效果良好，能够得到较低的错误率，但该方法对参数调节和核函数的选择敏感。

观点对比

半监督方法

通过对未标记的文本进行特征提取可以有效地改善文本情感分类结果，这种方法可以有效解决带有标记的数据集稀缺的问题。

无监督方法

根据文本间的相似性对未标记的文本进行分类，这种方法在情感分析中使用较少。

经典模型——基于传统机器学习

更简单，更精准，扩展性和可重复性更好，能取得更高的分类准确度，

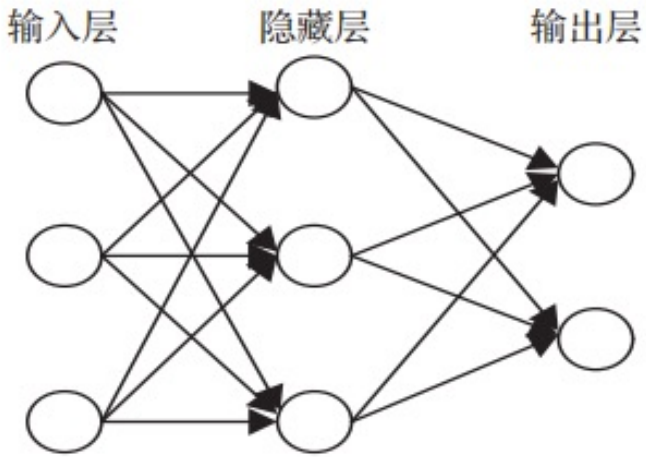
但是分类精确度依赖于高质量的标注训练集，大规模高质量的训练数据需要高人工成本，人为主观的数据标注结果也会影响分类效果。

不能充分利用**上下文文本**的语境信息，因此其分类准确性有一定的影响。

经典模型——基于深度学习

深度学习是人工神经网络在使用多层网络进行任务学习中的应用，随着深度学习在图像和语音处理方面取得重大进展，它在情感分析领域也开始被广泛应用。

能够有效解决基于传统情感分析方法中忽略上下文语义的问题。



单一神经网络

典型的神经网络学习方法有：卷积神经网络（CNN）、递归神经网络（RNN）、长短时记忆（LSTM）网络等。

混合神经网络

除了对单一神经网络的方法的研究之外，有不少学者在考虑了不同方法的优点后将这些方法进行组合和改进，并将其用于情感分析方面。

注意力机制

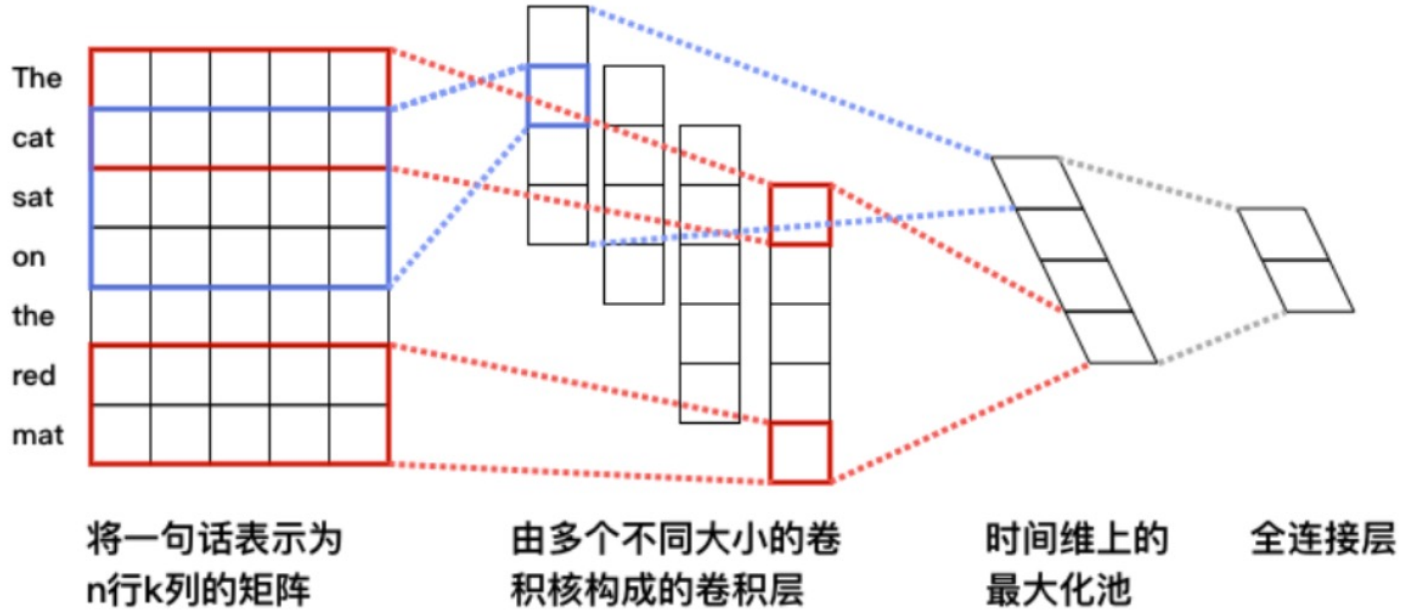
在深度学习的方法中加入注意力机制，用于情感分析任务的研究，能够更好地捕获上下文相关信息，提取语义信息，防止重要信息的丢失，可以有效提高文本情感分类的准确率。

预训练

预训练模型是指用数据集已经训练好的模型。通过对预训练模型的微调，可以实现较好的情感分类结果，最新的预训练模型有：ELMo、BERT、XL-NET、ALBERT等。

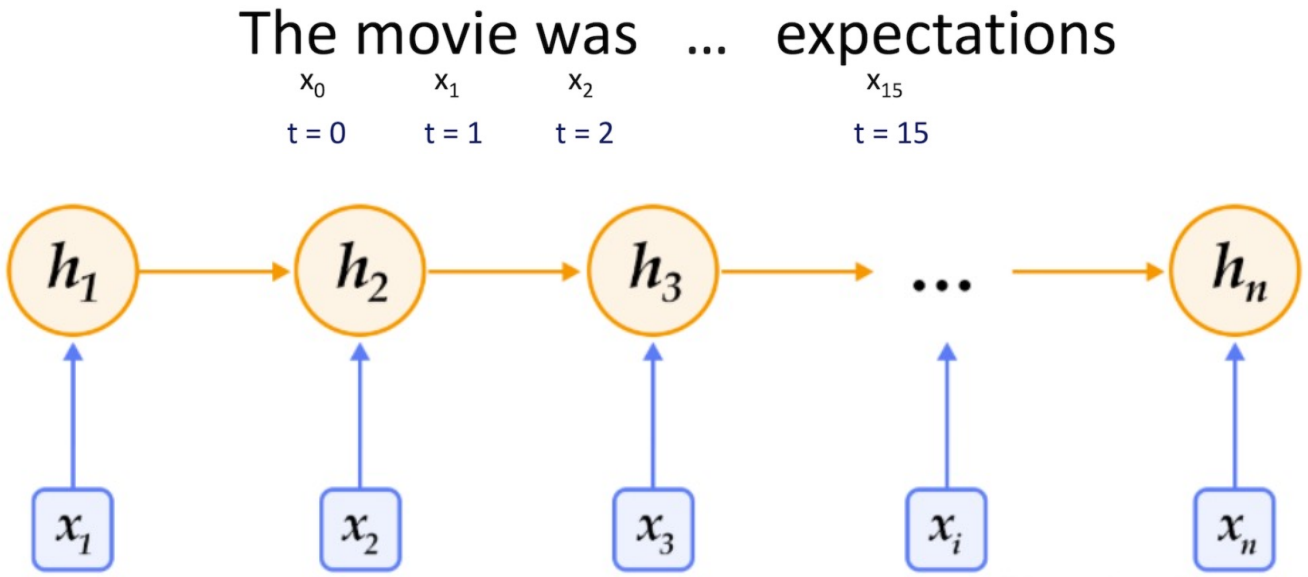
CNN-卷积神经网络

整体模型主要分为三个部分：输入层、卷积+池化层、全连接+softmax层。



RNN-循环神经网络

每个单词的出现都依赖于它的前一个单词和后一个单词。将词映射为其词向量表示，然后再作为循环神经网络每一时刻的输入Xt。

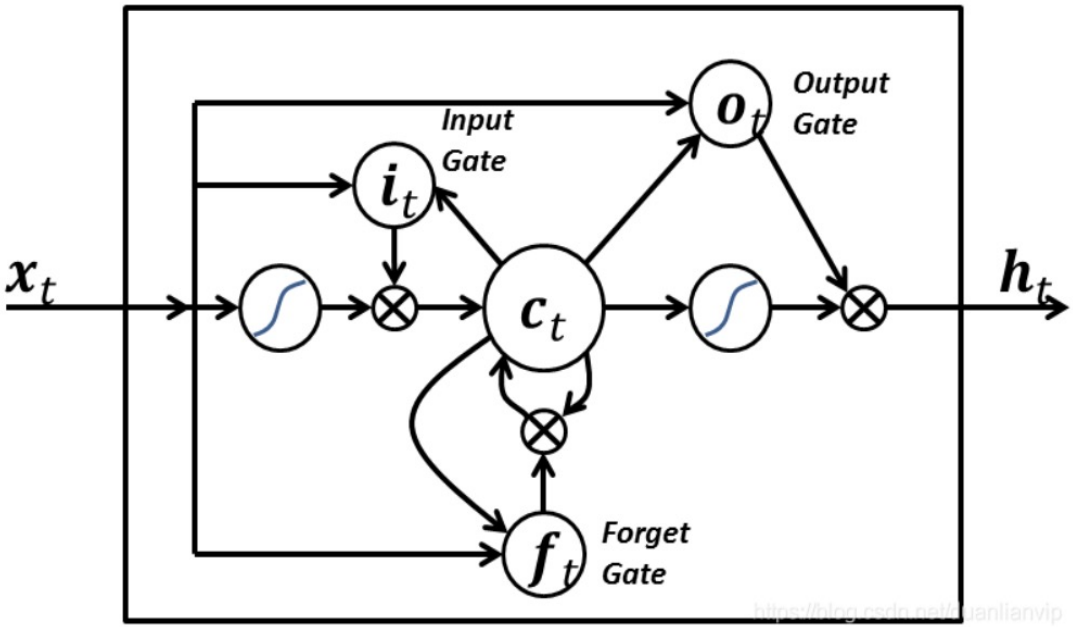


LSTM-长短期记忆网络

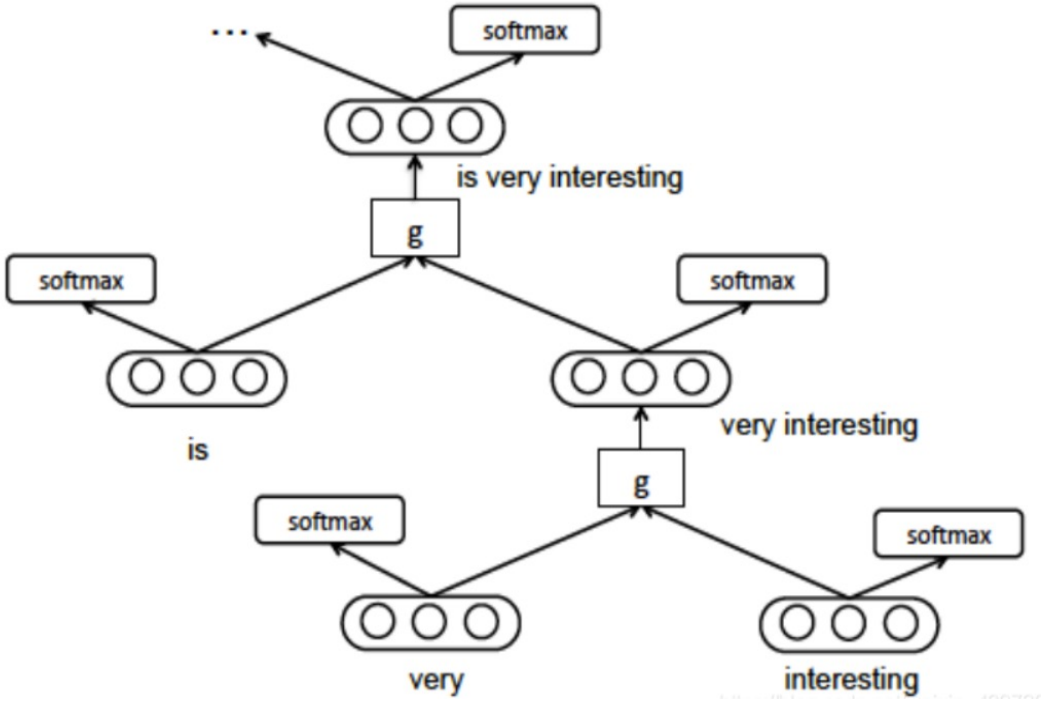
长短期记忆网络单元，是改进的RNN单元。

引入了记忆控制单元和三个门：输入门、遗忘门、输出门。实现了保留较长序列中的重要信息，忽略不重要信息。

从抽象的角度看，LSTM保存了文本中长期的依赖信息，解决RNN中梯度消失的问题。



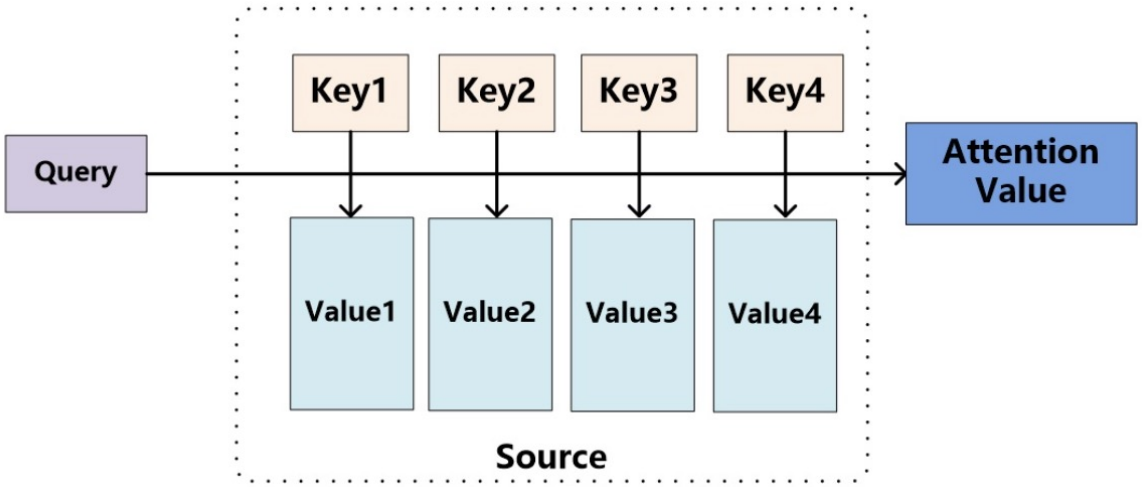
RecNN-递归神经网络



递归神经网络可以看作是循环神经网络的概括。

词嵌入是将单词表示成低维的稠密的实数向量。如何用稠密的向量表示短语，这是使用词向量的一个难题。在成分分析中，我们一般使用递归神经网络来解决这个问题。

Attention



一些特定的情感词，我们往往需要特别注意，因为它们是很重要的情感词，往往决定了评论者的情感。权重更高。

$$\mathbf{Attention}(\mathbf{Query}, \mathbf{Source}) = \sum_{i=1}^{L_x} Similarity(Query, Key_i) * Value_i$$

注意力增强的双向LSTM

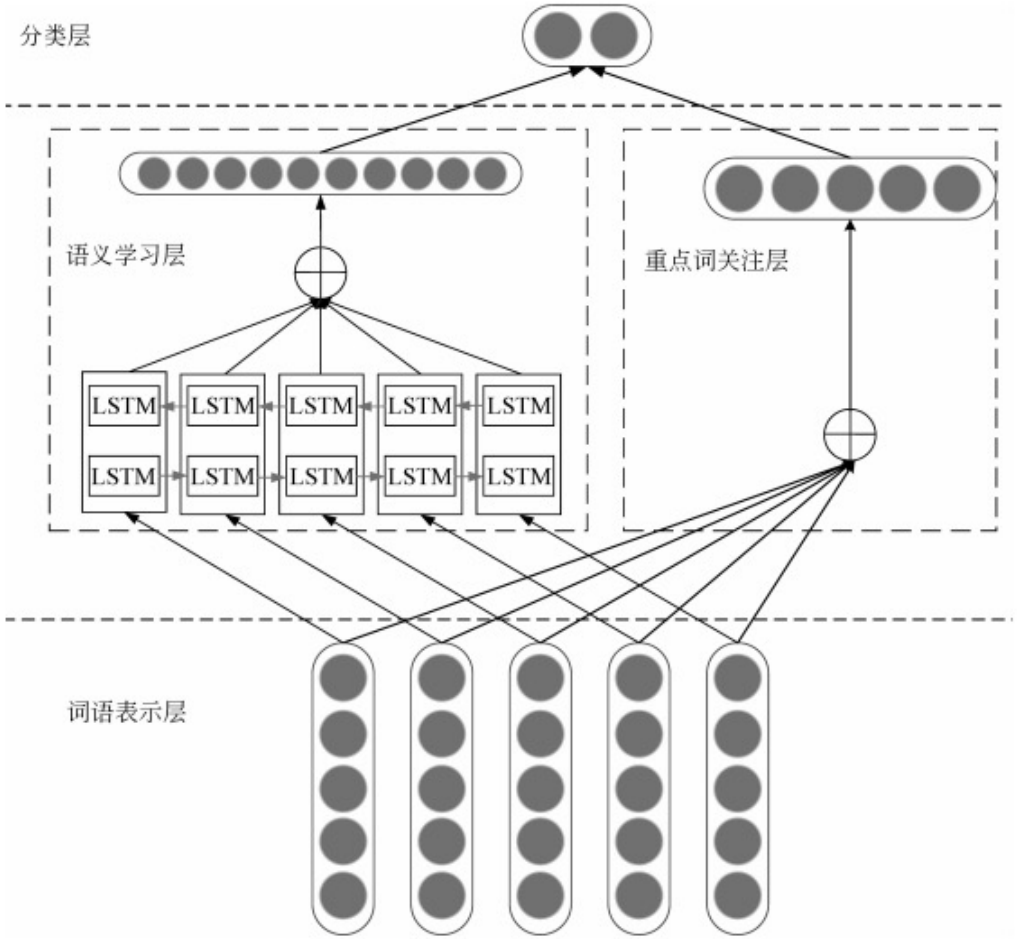


图 1　词向量注意力机制的 BiLSTM 模型结构

关鹏飞,李宝安,吕学强,周建设. 注意力增强的双向LSTM情感分析[J]. 中文信息学报, 2019, 33(2): 105-111.

无需人工标签，可以从海量的语料中可以学习到通用的语言表示，并显著提升下游的任务。

ELMo

双向的LSTM语言模型，由一个前向和一个后向语言模型构成，目标函数就是取这两个方向语言模型的最大似然值。

利用LSTM网络结构生成词语的表征，**每一个词只对应一个词向量。**

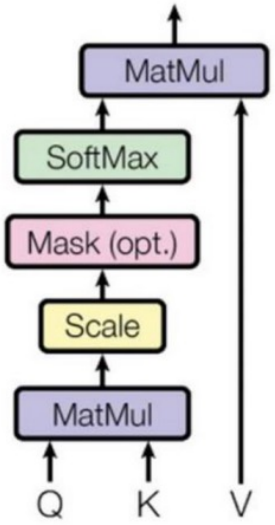
根据具体输入从该语言模型中可以得到**上下文依赖的当前词表示。**

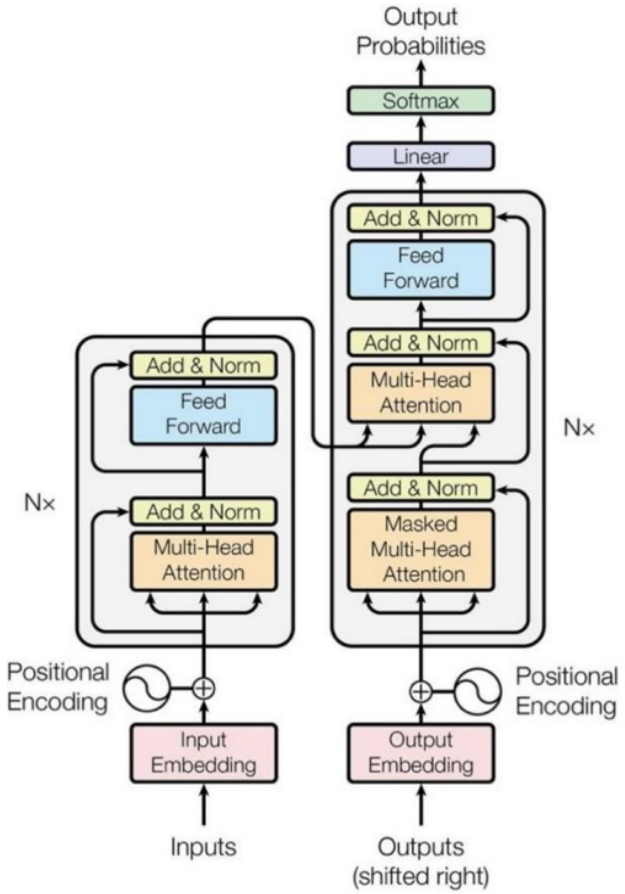
BERT

将双向的transformer机制用于语言模型，充分考虑到单词的上下文语义信息。在模型的输入方面BERT使用了WordPiece embedding 作为词向量，并加入了位置向量和句子切分向量。

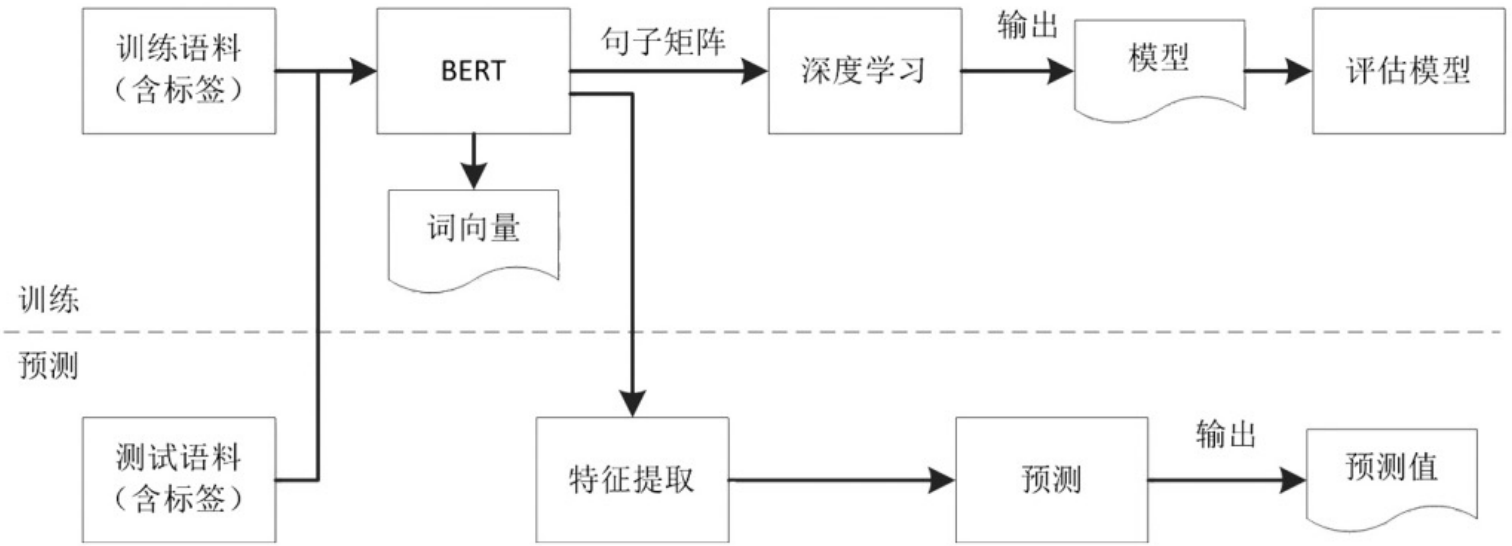
Transformer

预训练语言模型可以有效地捕获文本中的语法和语义。
Transformer 可以并行处理句子中出现的每个词，训练速度更快。
词向量化是 Transformer 的核心之一。
自注意力机制(self-attention) 。





BERT是一个预训练的语言表征模型。它不再像以往一样采用传统的单向语言模型或者把两个单向语言模型进行浅层拼接的方法进行预训练，而是采用新的**masked language model（MLM），**以致能生成**深度的双向**语言表征。



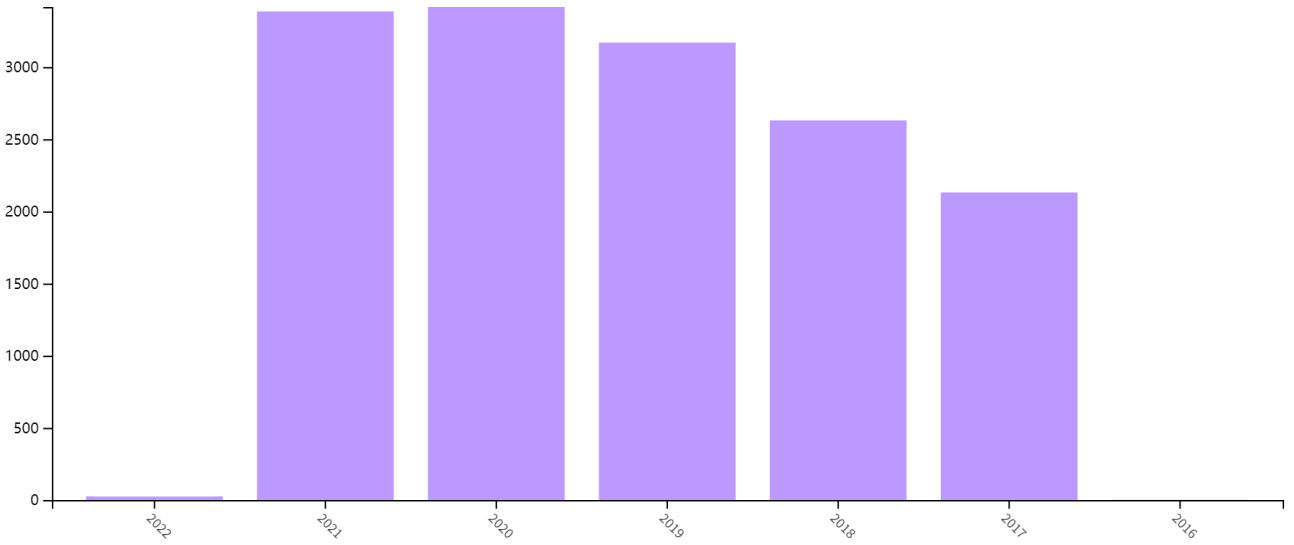
模型	验证集			测试集		
	dev acc	*F*1	每轮用时/s	test acc	*F*1	每轮用时/s
Attention-based RNN	0.737 8	0.798 0	25.487 1	0.714 4	0.805 6	11.394 7
BERT-base	0.782 3	0.831 5	23.541 2	0.732 0	0.837 6	10.113 6
BERT-base+transfer	0.799 2	0.836 5	31.318 6	0.752 3	0.842 3	16.117 2

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

前沿进展

黄彦宸

“Sentiment Analysis”相关文献数量

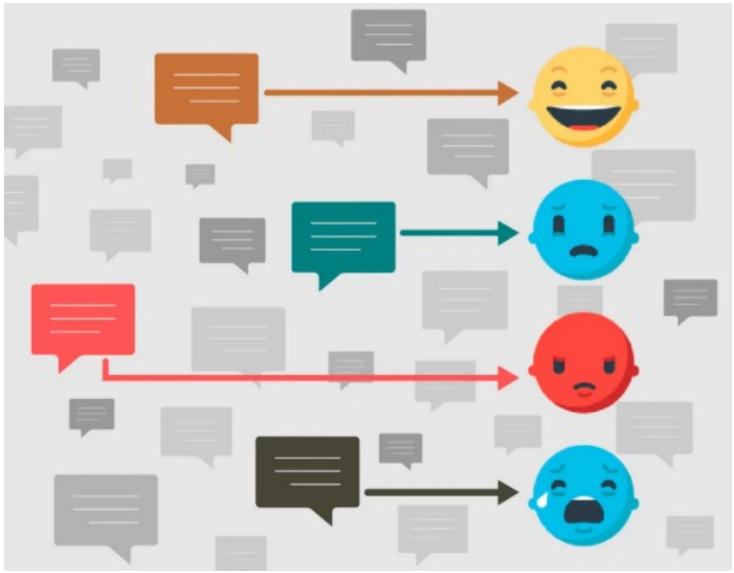


*数据源自web of science

NMNLP	主会议	24篇
	Findings	12篇
ACL	主会议	15篇
	Findings	12篇

*数据源自知乎用户“刘聪NLP”


热门研究方向
基础任务
多模态
对话
细粒度
情感信息抽取



句子级情感分析

&

篇章级情感分析

方面级情感分析

Aspect-based Sentiment Analysis

基础任务

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

1. A Neural Group-wise Sentiment Analysis Model with Data Sparsity Awareness. Zhou et.al. AAAI2021

本文尝试引入文本之外的信息来协助情感分类，通过引入用户信息来考虑每个人的偏好和语言习惯，并提出了一种神经群体情感分析模型解决数据稀疏性的问题。

2. Segmentation of Tweets with URLs and its Applications to Sentiment Analysis. Aljebreen et.al. AAAI2021

在社交媒体平台中传播信息的一种重要方式是在用户帖子中包含指向外部来源的 URL，本文研究了带有URL推文的结构，提出算法解决了带有URL推文的分割问题。

方面级情感分析

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

1. Context-Guided BERT for Targeted Aspect-Based Sentiment Analysis. Wu et.al. AAAI2021

本文通过构建上下文引导的BERT模型(CGBERT)，将上下文感知自注意网络(Yang等人2019)扩展到(T)ABSA任务,以及提出了一种新的准注意上下文引导BERT模型(QACG-BERT)。

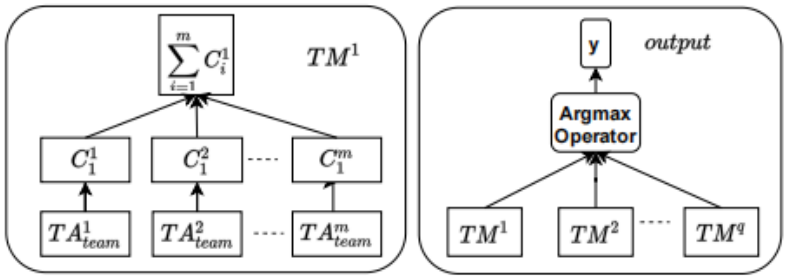
2. An Adaptive Hybrid Framework for Cross-domain Aspect-based Sentiment Analysis. Zhou et.al. AAAI2021

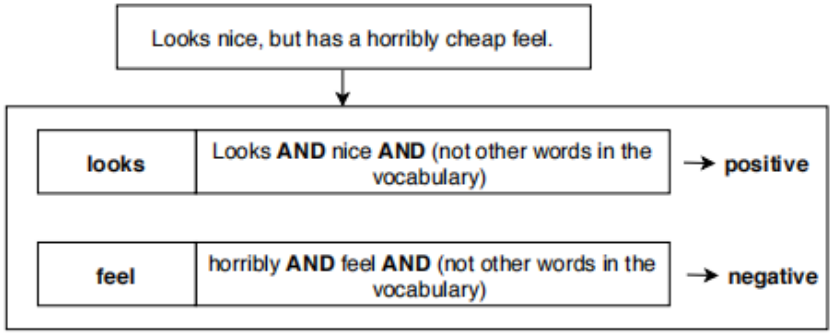
本文提出了一个自适应混合框架，将半监督学习和对抗训练集成在同一个网络中，解决了对抗训练中的任务分类器无法利用目标域数据中方面和情感相关信息的问题。

3. Human-Level Interpretable Learning for Aspect-Based Sentiment Analysis. Yadav et.al. AAAI2021

本文提出了一种人类可解释的学习方法以增强可解释性。

可解释性


$\sum_{i=1}^{m} C_i^1$
TM^1
C_1^1
C_1^2
C_1^m
TA_{team}^1
TA_{team}^2
TA_{team}^m
y
output
Argmax Operator
TM^1
TM^2
TM^q


Looks nice, but has a horribly cheap feel.
looks
Looks AND nice AND (not other words in the vocabulary)
positive
feel
horribly AND feel AND (not other words in the vocabulary)
negative

情感信息联合抽取

北京理工大學 BEIJING INSTITUTE OF TECHNOLOGY

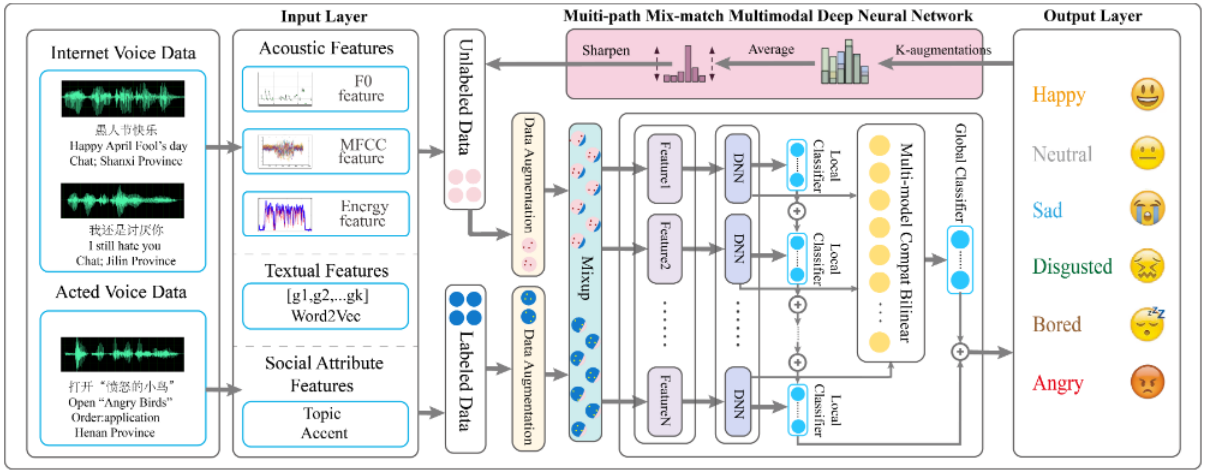
1. Question-Driven Span Labeling Model for Aspect–Opinion Pair Extraction

2. Bidirectional Machine Reading Comprehension for Aspect Sentiment Triplet Extraction

3. A Joint Training Dual-MRC Framework for Aspect Based Sentiment Analysis

4. A Simple and Effective Self-Supervised Contrastive Learning Framework for Aspect Detection

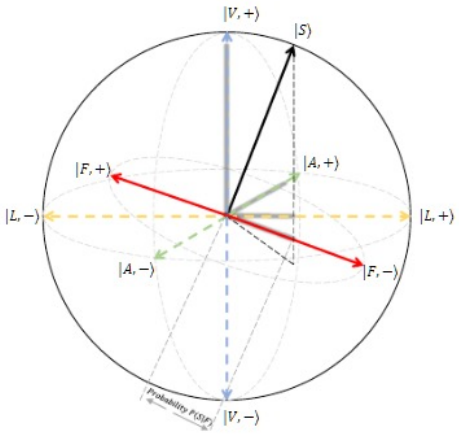
多模态情感分析



*资料源自*Inferring Emotion from Large-scale Internet Voice Data: A Semi-supervised Curriculum Augmentation based Deep Learning Approach*

北京理工大學 BEIJING INSTITUTE OF TECHNOLOGY

基于量子认知的全新多模态融合策略

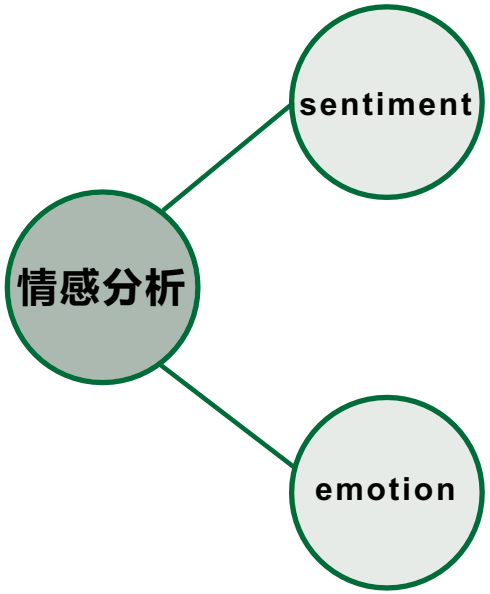


	CMU-MOSI		CMU-MOSEI	
Approach	Acc_2	$F1$	Acc_2	$F1$
Linguistic Only	77.1	72.3	81.5	87.8
Visual Only	54.7	48.4	71.1	83.0
Acoustic Only	56.1	60.0	71.2	83.1
Proposed Framework	**84.6** (↑ 7.5)	**84.5** (↑12.2)	**84.9** (↑ 3.4)	**91.1** (↑3.3)

	CMU-MOSI		CMU-MOSEI	
Approach	Acc_2	$F1$	Acc_2	$F1$
Framework$_{\{Linguistic,Visual\}}$	78.2	74.3	82.1	88.4
Framework$_{\{Linguistic,Acoustic\}}$	79.6	75.1	82.7	89.2
Framework$_{\{Visual,Acoustic\}}$	55.1	55.2	70.8	82.7
Proposed Framework	**84.6** (↑ 5.0)	**84.5** (↑9.4)	**84.9** (↑ 2.2)	**91.1** (↑1.9)

*资料源自*Quantum Cognitively Motivated Decision Fusion for Video Sentiment Analysis*

对话情感任务



北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

对话情感任务

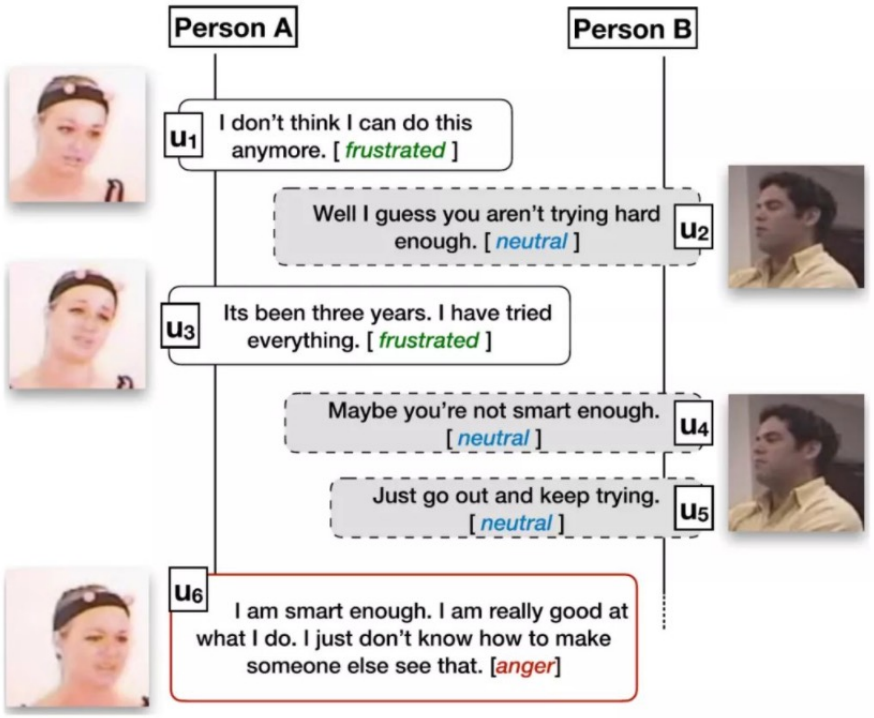
识别情绪

生成情绪

其他回答任务

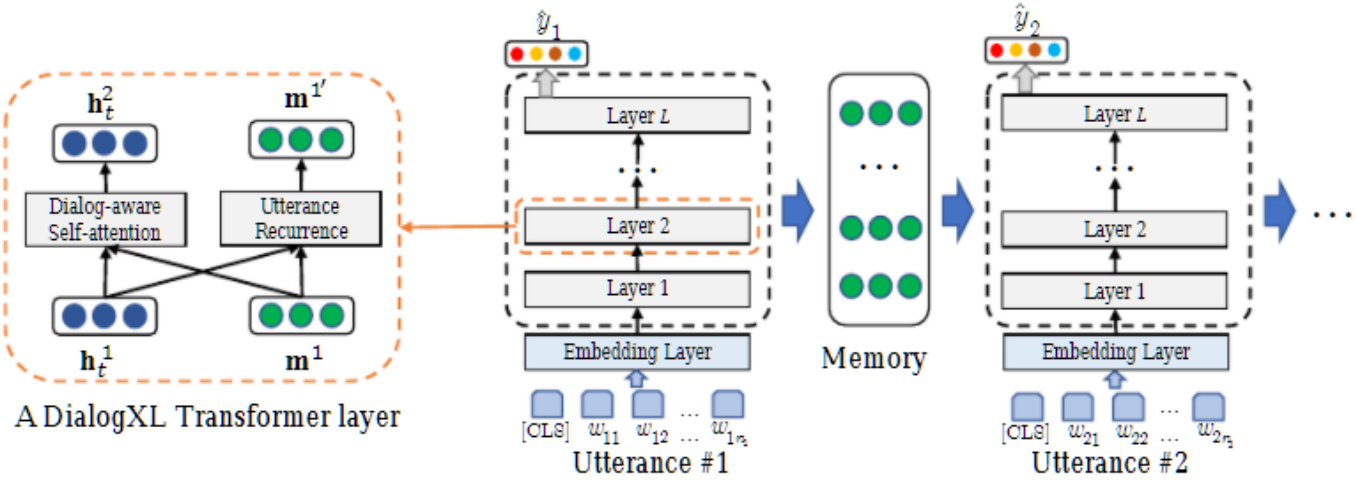
对话交替出现

……



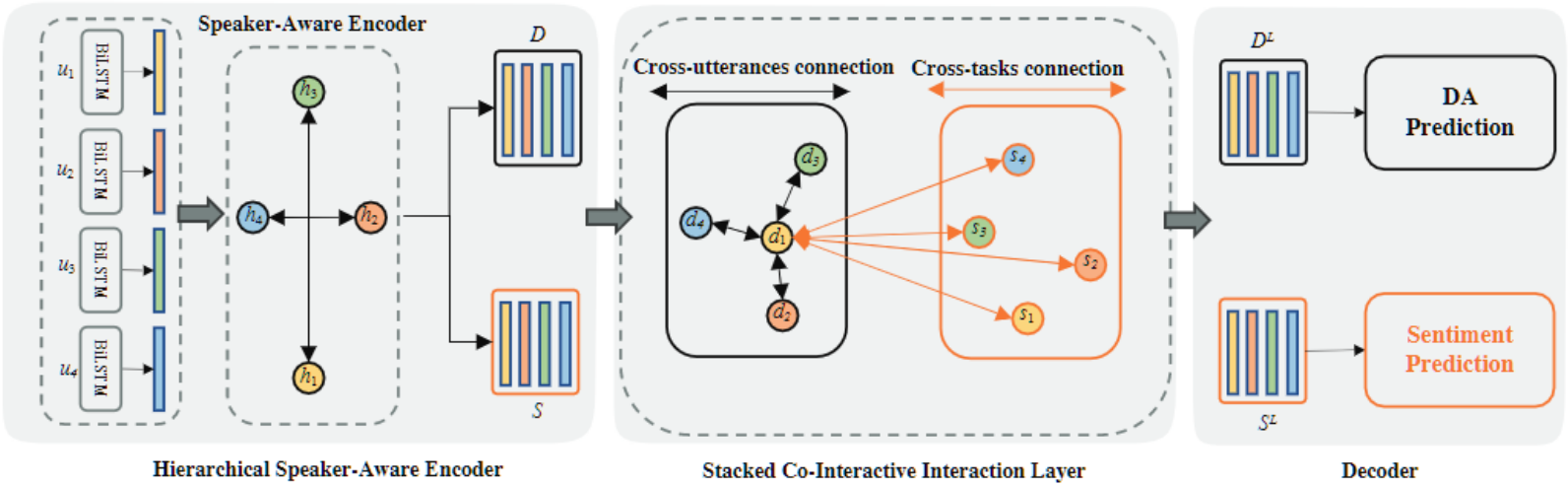
对话情感任务

XLNet 》 DialogXL



*资料源自*DialogXL: All-in-One XLNet for Multi-Party Conversation Emotion Recognition*

对话情感任务

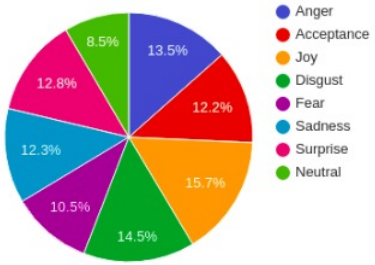


*资料源自*Co-GAT: A Co-Interactive Graph Attention Network for Joint Dialog Act Recognition and Sentiment Classification*

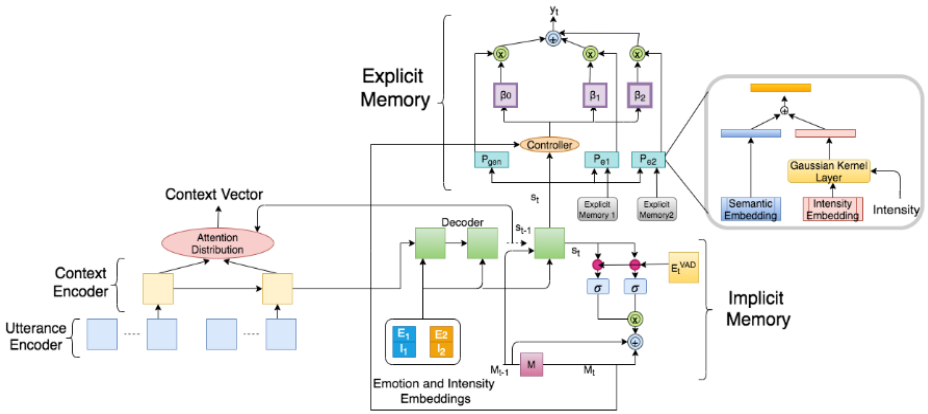
对话情感任务

MEIMD

多重情感和强度意识多方对话数据集



	Conversations	Emotions
1	It's amazing, I am thrilled you got promoted I have loads of work now and am afraid to complete it. Stop sulking, I am sure you will manage it.	Surprise (0.3), Joy (0.9) Disgust(0.3), Fear(0.6) Anger(0.3), Acceptance(0.6)
2	I am sorry this could be an infection or cancer. I am afraid but I know you could help me.	Sadness(0.6), Fear(0.3) Acceptance(0.3), fear(0.6)



*资料源自*More the Merrier: Towards Multi-Emotion and Intensity Controllable Response Generation*

中文情感分析

2021年5月，美团NLP中心开源了迄今规模最大的基于真实场景的中文属性级情感分析数据集ASAP，该数据集相关论文被自然语言处理顶会NAACL2021录用，同时该数据集加入中文开源数据计划千言，将与其他开源数据集一起推动中文信息处理技术的进步。

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

demo展示

管树言

demo-预览

词典方法

传统机器学习

深度学习方法

调用百度api

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

数据集介绍：

	label	review
0	1	很快，好吃，味道足，量大
1	1	没有送水没有送水没有送水
2	1	非常快，态度好。
3	1	方便，快捷，味道可口，快递给力
4	1	菜味道很棒！送餐很及时！
...	...	...
10980	0	以前几乎天天吃，现在调料什么都不放，
10981	0	昨天订凉皮两份，什么调料都没有放，就放了点麻油，特别难吃，丢了一份，再也不想吃了
10982	0	凉皮太辣,吃不下都
10983	0	本来迟到了还自己点！！！
10984	0	肉夹馍不错，羊肉泡馍酱肉包很一般。凉面没想象中好吃。送餐倒是很快。

[10985 rows x 2 columns]

waimai_10k某外卖平台收集的用户评价，正向4000 条，负向约 8000 条

数据集链接：
https://raw.githubusercontent.com/SophonPlus/ChineseNlpCorpus/master/datasets/waimai_10k/waimai_10k.csv

demo-简介

北京理工大學 BEIJING INSTITUTE OF TECHNOLOGY

	label	review
0	1	更博了，爆照了，帅的呀，就是越来越爱你！生快傻缺[爱你][爱你][爱你]
1	1	@张晓鹏jonathan 土耳其的事要认真对待[哈哈]，否则直接开除。@丁丁看世界 很是细心...
2	1	姑娘都羡慕你呢…还有招财猫高兴……//@爱在蔓延-JC:[哈哈]小学徒一枚，等着明天见您呢/...
3	1	美~~~~~[爱你]
4	1	梦想有多大，舞台就有多大![鼓掌]
...	...	...
119983	0	一公里不到，县医院那个天桥下右拐200米就到了！//@谢礼恒：我靠。这个太霸道了！离224...
119984	0	今天真冷啊，难道又要穿棉袄了[晕]？今年的春天真的是百变莫测啊[抓狂]
119985	0	最近几天就没停止过！！！[伤心]
119986	0	//@毒药女流氓:[怒] 很惨！
119987	0	呢??@杰?Kelena ？！[抓狂]?搞乜鬼？？！！想知？入去GOtrip睇睇： htt...

[119988 rows x 2 columns]

10 万多条，带情感标注 新浪微博，正负向评论约各 5 万条

数据集链接：
https%3A//github.com/SophonPlus/ChineseNlpCorpus/raw/master/datasets/online_shopping_10_cats/online_shopping_10_cats.zip

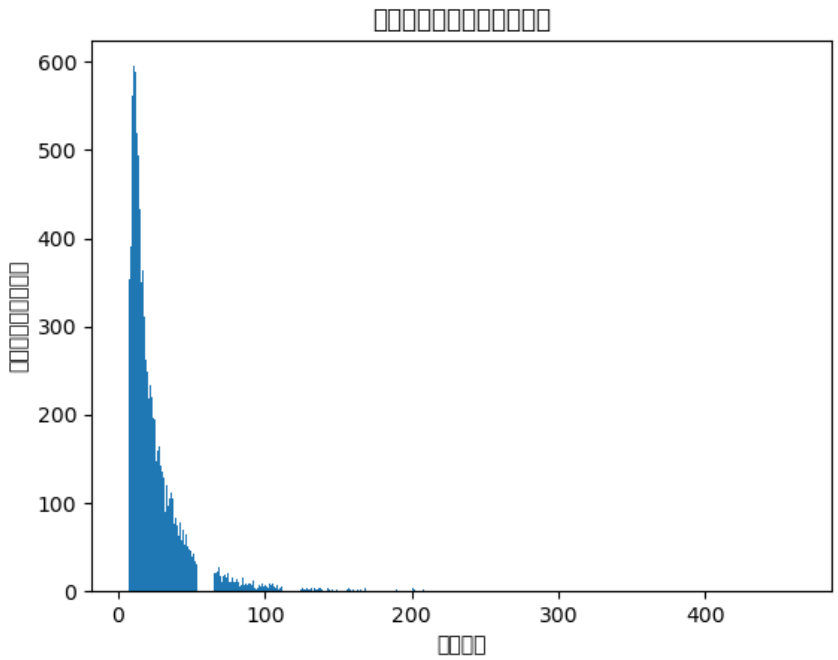
demo-简介

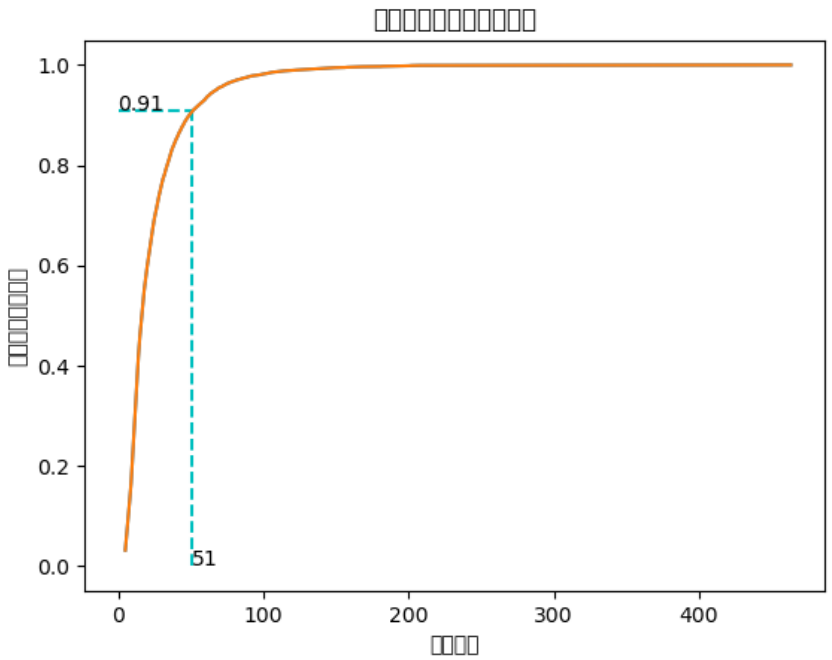
北京理工大學 BEIJING INSTITUTE OF TECHNOLOGY

```
df = pd.read_csv('data/weibo_senti_100k.csv')
print(df.groupby('label')['label'].count())

df['length'] = df['review'].apply(lambda x: len(x))
len_df = df.groupby('length').count()
sent_length = len_df.index.tolist()
sent_freq = len_df['review'].tolist()

# 绘制句子长度及出现频数统计图
plt.bar(sent_length, sent_freq)
plt.title(u"句子长度及出现频数统计图")
plt.xlabel(u"句子长度")
plt.ylabel(u"句子长度出现的频数")
plt.savefig("./句子长度及出现频数统计图.png")
plt.close()

# 绘制句子长度累积分布函数(CDF)
sent_pentage_list = [(count/sum(sent_freq)) for count in accumulate(sent_freq)]

# 绘制CDF
plt.plot(sent_length, sent_pentage_list)

# 寻找分位点为quantile的句子长度
quantile = 0.91
#print(list(sent_pentage_list))
for length, per in zip(sent_length, sent_pentage_list):
    if round(per, 2) == quantile:
        index = length
        break
print("\n分位点为%s的句子长度:%d." % (quantile, index))

# 绘制句子长度累积分布函数图
plt.plot(sent_length, sent_pentage_list)
plt.hlines(quantile, 0, index, colors="c", linestyles="dashed")
plt.vlines(index, 0, quantile, colors="c", linestyles="dashed")
plt.text(0, quantile, str(quantile))
plt.text(index, 0, str(index))
```

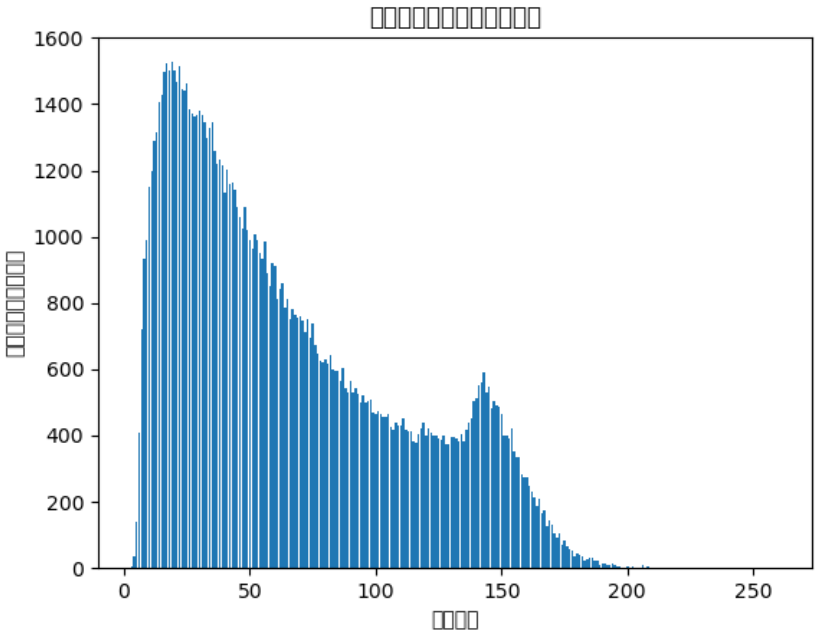
通过相关代码，显示数据的长度以及分布信息：

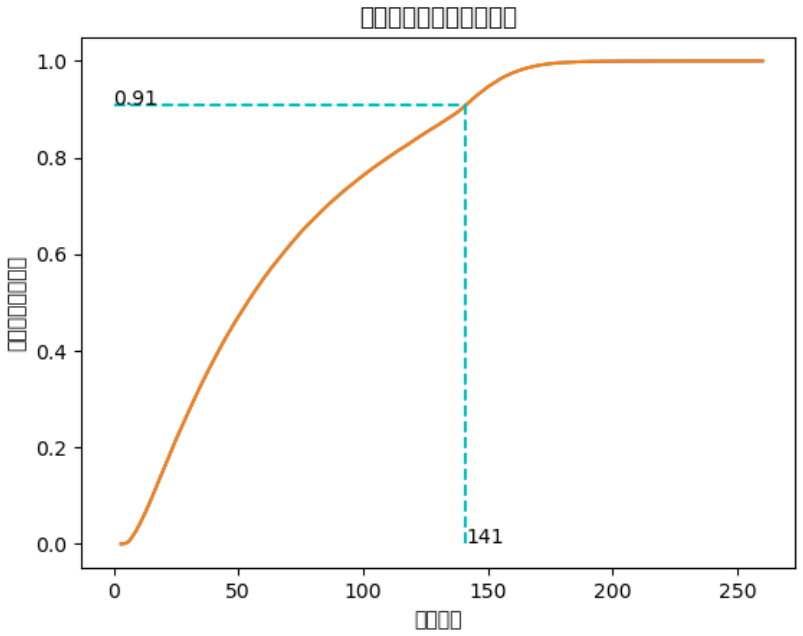
demo-简介


600
500
400
300
200
100
0
0
100
200
300
400


1.0
0.8
0.6
0.4
0.2
0.0
0.91
51
0
100
200
300
400

demo-简介


1600
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250


1.0
0.8
0.6
0.4
0.2
0.0
0.91
141
0
50
100
150
200
250

词典方法

demo-情感词典


```
最尼玛 -6.70400012637
扰民 -6.49756445867
fuck... -6.32963390433
RNM -6.21861284426
wcnmlgb -5.96710044003
2.5: -5.90459648251
Fxxk -5.87247473641
MLP -5.87247473641
吃哑巴亏 -5.77120419579
IAQI -5.77107837123
MLGBD -5.69408191501
NNND -5.66228462641
MLGB. -5.60457743583
成甘 -5.60457743583
最桑 -5.60457743583
真无语 -5.60457743583
ＴＭ -5.60457743583
次奥次奥次奥 -5.59258287133
cnmd -5.54446545761
MBD -5.50280109843
NNDX -5.48173951768
水蛭 -5.48173951768
美素丽 -5.48173951768
草尼 -5.48173951768
凌迟 -5.46005372985
尼玛尼玛尼玛 -5.42622557462
冠周炎 -5.41616190446
加塞儿 -5.41616190446
日尼玛 -5.41616190446
```

demo-情感词典

情感分析用词语集(beta 版):http://www.keenage.com/html/c_bulletin_2007.htm
NTUSD-简体中文情感极性词典:http://www.datatang.com/data/11837
程度副词及强度和否定词表:http://www.datatang.com/data/44198
现有情感词典汇总:http://www.datatang.com/data/46922

demo-情感词典

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

```
# 计算得分
def sentiment_score(sentence):
    # 1.对文档分词
    seg_list = seg_word(sentence)
    # 2.将分词结果转换成字典，找出情感词、否定词和程度副词
    sen_word, not_word, degree_word = classify_words(seg_list)
    # 3.计算得分
    score = score_sentiment(sen_word, not_word, degree_word, seg_list)
    return score
```

```
        # 若是程度副词
        if i in degree_word.keys():
            W*=float(degree_word[i])
        # 若是否定词
        elif i in not_word.keys():
            # print(i)
            W*=-1
        elif i in sen_word.keys():
            score+=float(W)*float(sen_word[i])
            W=1
    return score
```

demo-情感分析

```
print("宝你真的好美我好爱你", sentiment_score("宝你真的好美我好爱你"))
print('滚一边去，烦人',
      sentiment_score('滚一边去，烦人'))
print('宝和你聊天真开心，记得想我', sentiment_score('宝和你聊天真开心，记得想我'))
print('你有完没完，都说了我不想说话', sentiment_score('你有完没完，都说了我不想说话'))
print('那宝你记得多喝水，呜呜呜，想我了你就找我' ,
        sentiment_score('那宝你记得多喝水，呜呜呜，想我了你就找我'))
print('我们还是分手吧，我不喜欢你了',sentiment_score('我们还是分手吧，我不喜欢你了'))
```

结果：

```
Building prefix dict from the default dictionary ...
Loading model from cache /var/folders/lk/q1qry7w577z714b2f9nfv
Loading model cost 0.746 seconds.
Prefix dict has been built successfully.
宝你真的好美我好爱你 3.591915784882
滚一边去，烦人 -5.199494420063
宝和你聊天真开心，记得想我 4.4018854257393
你有完没完，都说了我不想说话 -1.939586542621
那宝你记得多喝水，呜呜呜，想我了你就找我 -4.177168074092
我们还是分手吧，我不喜欢你了 -2.3566957262
(dl) mac@ShuyandeMacBook-Air Emotional Analysis %
```

demo-情感分析

```
现在测试了   700  个例子
现在测试了   800  个例子
现在测试了   900  个例子
现在测试了   1000  个例子
现在测试了   1100  个例子
现在测试了   1200  个例子
现在测试了   1300  个例子
现在测试了   1400  个例子
现在测试了   1500  个例子
现在测试了   1600  个例子
现在测试了   1700  个例子
现在测试了   1800  个例子
现在测试了   1900  个例子
现在测试了   2000  个例子
accuracy = 0.4025
```

情感词典对于我们的数据集效果极差

```
现在测试了   800  个例子
现在测试了   900  个例子
现在测试了   1000  个例子
现在测试了   1100  个例子
现在测试了   1200  个例子
现在测试了   1300  个例子
现在测试了   1400  个例子
现在测试了   1500  个例子
现在测试了   1600  个例子
现在测试了   1700  个例子
现在测试了   1800  个例子
现在测试了   1900  个例子
现在测试了   2000  个例子
accuracy = 0.254
```

demo-情感分析

```
(dl) PS C:\Users\94257\Desktop\Emotional Analysis>  c:; cd '
\94257\.vscode\extensions\ms-python.python-2022.2.1924087327
tion_dirct.py'
Building prefix dict from the default dictionary ...
Dumping model to file cache C:\Users\94257\AppData\Local\Tem
Loading model cost 0.715 seconds.
Prefix dict has been built successfully.
测试结束,数据如下：
accuracy:  0.7065638674723901
precision:  0.6424465394343527
recall:  0.9315662248895558
训练加测试耗时：   2811.80201125144 96
(dl) PS C:\Users\94257\Desktop\Emotional Analysis>
```

传统机器学习方法

demo-朴素贝叶斯分类器

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

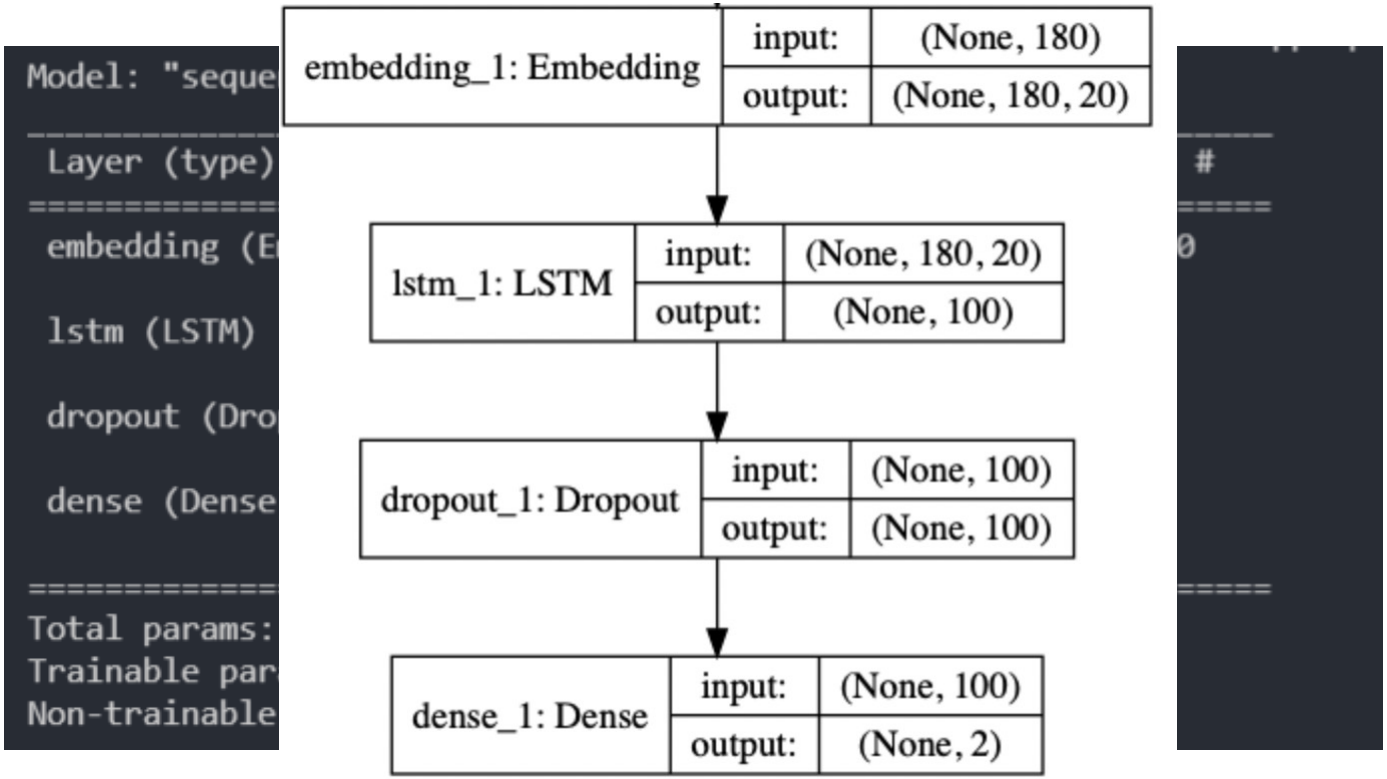
```
def _train(self, train
    """…
    print("BayesClassi
    # get the frequenc
    total_pos_data, to
    total_pos_length,
    total_word = set()
    for i, doc in enum
        if train_data_
            for word i
                if bes
                    to
                    to
                    to
        else:
            for word i
                if bes
                    to
                    to
                    to
    self._pos_p = tota
    self._neg_p = tota
    # get each word's
    for word in total_
        self._pos_word
            total_pos_
        self._neg_word
            total_neg_
    print("BayesClassi
```

```
BayesClassifier is training ......
BayesClassifier trains over!
测试结束,数据如下:
accuracy:  0.8111888111888111
precision:  0.855188141391106
recall:  0.7492507492507493
训练加测试耗时:  0.19643759727478027
(dl) PS C:\Users\94257\Desktop\Emotional Analysis>

(dl) PS C:\Users\94257\Desktop\Emotional Analysis>  c:; cd 'c:
\94257\.vscode\extensions\ms-python.python-2022.2.1924087327\p
in_bayes.py'
BayesClassifier is training ......
BayesClassifier trains over!
测试结束,数据如下:
accuracy:  0.8911439883309022
precision:  0.8962925428595557
recall:  0.8846378261231974
训练加测试耗时:  3.579425096511841
(dl) PS C:\Users\94257\Desktop\Emotional Analysis>
```

深度学习方法

参考网址：https://www.cnblogs.com/jclian91/p/10886031.html

demo-lstm



参考网址：https://www.cnblogs.com/jclian91/p/10886031.html

demo-lstm

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

```
# 创建深度学习模型， Embedding + LSTM + Softmax.
def create_LSTM(n_units, input_shape, output_dim, filepath):
    x, y, output_dictionary, vocab_size, label_size, inverse_word_dictionary = load_data(
        filepath)
    model = Sequential()
    model.add(
        Embedding(input_dim=vocab_size + 1,
                  output_dim=output_dim,
                  input_length=input_shape,
                  mask_zero=True))
    model.add(LSTM(n_units, input_shape=(x.shape[0], x.shape[1])))
    model.add(Dropout(0.2))
    model.add(Dense(label_size, activation='softmax'))
    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    #plot_model(model, to_file='./model_lstm.png', show_shapes=True)
    model.summary()

    return model
```

参考网址：https://www.cnblogs.com/jclian91/p/10886031.html

demo-lstm

```
_________________________________________________________________
Epoch 1/5
275/275 [==============================] - 24s 76ms/step - loss: 0.4175 - accuracy: 0.8188
Epoch 2/5
275/275 [==============================] - 22s 79ms/step - loss: 0.2810 - accuracy: 0.8944
Epoch 3/5
275/275 [==============================] - 21s 78ms/step - loss: 0.2575 - accuracy: 0.9028
Epoch 4/5
275/275 [==============================] - 21s 75ms/step - loss: 0.2473 - accuracy: 0.9100
Epoch 5/5
275/275 [==============================] - 21s 75ms/step - loss: 0.2333 - accuracy: 0.9140
2022-03-12 19:12:15.203403: W tensorflow/python/util/util.cc:368] Sets are not currently co
 so consider avoiding using them.
WARNING:absl:Found untraced functions such as lstm_cell_layer_call_fn, lstm_cell_layer_call
 of 2). These functions will not be directly callable after loading.
WARNING:absl:<keras.layers.recurrent.LSTMCell object at 0x0000022DBCF410D0> has the same na
aming <class 'keras.layers.recurrent.LSTMCell'> to avoid naming conflicts when loading with
ible, pass the object in the `custom_objects` parameter of the load function.
训练完成!
accuracy:  0.8862084660901229
precision:  0.8688118811881188
recall:  0.8297872340425532
训练加测试耗时:  208.84450793266296
```

参考网址：https://www.cnblogs.com/jclian91/p/10886031.html

demo-lstm

北京理工大學 BEIJING INSTITUTE OF TECHNOLOGY

```
_________________________________________________________________
Epoch 1/5
3000/3000 [==============================] - 247s 81ms/step - loss: 0.1314 - accuracy: 0.9580
Epoch 2/5
3000/3000 [==============================] - 231s 77ms/step - loss: 0.1091 - accuracy: 0.9705
Epoch 3/5
3000/3000 [==============================] - 233s 78ms/step - loss: 0.0762 - accuracy: 0.9792
Epoch 4/5
3000/3000 [==============================] - 237s 79ms/step - loss: 0.0724 - accuracy: 0.9789
Epoch 5/5
3000/3000 [==============================] - 236s 79ms/step - loss: 0.0640 - accuracy: 0.9818
2022-03-12 14:57:36.809096: W tensorflow/python/util/util.cc:368] Sets are not currently considered
iding using them.
WARNING:absl:Found untraced functions such as lstm_cell_layer_call_fn, lstm_cell_layer_call_and_ret
nctions will not be directly callable after loading.
WARNING:absl:<keras.layers.recurrent.LSTMCell object at 0x000002070F6E41F0> has the same name 'LSTM
ras.layers.recurrent.LSTMCell'> to avoid naming conflicts when loading with `tf.keras.models.load_m
stom_objects` parameter of the load function.
训练完成!
accuracy:  0.9835402950245854
precision:  0.9677472437057759
recall:  0.9997450063748407
训练加测试耗时：  2204.8468425273895
(dl) PS C:\Users\94257\Desktop\lstm>
```

参考网址：https://www.cnblogs.com/jclian91/p/10886031.html

调用百度api简单测试：

demo-简单api测试

demo-简单api测试

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

```
已经测试了  1987  个例子
已经测试了  1988  个例子
已经测试了  1989  个例子
已经测试了  1990  个例子
已经测试了  1991  个例子
已经测试了  1992  个例子
已经测试了  1993  个例子
已经测试了  1994  个例子
已经测试了  1995  个例子
已经测试了  1996  个例子
已经测试了  1997  个例子
已经测试了  1998  个例子
已经测试了  1999  个例子
已经测试了  2000  个例子
已经测试了  2001  个例子
测试结束,数据如下:
accuracy:  0.9090454772613693
precision:  0.9090909090909091
recall:  0.9090909090909091
(dl) PS C:\Users\94257\Desktop\baidu_ea>
```

demo-简单api测试

```
请输入句子: 如此美妙的开局！为我欢呼！为我喝彩！！
{'log_id': 5668309738670136908, 'text': '如此美妙的开局！为我欢呼！为我喝彩！！', 'items': [{'posi
tive_prob': 0.999915, 'confidence': 0.99981, 'negative_prob': 8.54914e-05, 'sentiment': 2}]}
请输入句子: 呵呵
{'log_id': 2053135803097318284, 'text': '呵呵', 'items': [{'positive_prob': 0.867607, 'confidence'
: 0.705793, 'negative_prob': 0.132393, 'sentiment': 2}]}
请输入句子: 呵呵，无语
{'log_id': 30875105646998509 56, 'text': '呵呵，无语', 'items': [{'positive_prob': 0.000136096, 'co
nfidence': 0.999697, 'negative_prob': 0.999864, 'sentiment': 0}]}
请输入句子: 呵呵，行
{'log_id': 8177071534251383852, 'text': '呵呵，行', 'items': [{'positive_prob': 0.983016, 'confide
nce': 0.962258, 'negative_prob': 0.0169838, 'sentiment': 2}]}
请输入句子: 你要这么说我也没有办法，随便你好了
{'log_id': 12488065588806263 48, 'text': '你要这么说我也没有办法，随便你好了', 'items': [{'positive
_prob': 0.00279276, 'confidence': 0.993794, 'negative_prob': 0.997207, 'sentiment': 0}]}
请输入句子: 呵呵，我没事，我很开心，别管我
{'log_id': 3442875268590333964, 'text': '呵呵，我没事，我很开心，别管我', 'items': [{'positive_pro
b': 0.999476, 'confidence': 0.998835, 'negative_prob': 0.000524215, 'sentiment': 2}]}
请输入句子:
```

可以发现即使是比较先进的情感分析方法，也难以处理”阴阳怪气“之类的句子（就像男朋友们永远不能理解女朋友们的情绪）

测试总览

waimai_10k	方法	accuracy	precision	recall	time
	词典方法	0.4025			
	朴素贝叶斯	0.8112	0.8551	0.7493	0.19
	lstm	0.8862	0.8688	0.8298	208.8
weibo_100k	方法	accuracy	precision	recall	time
	词典方法	0.7065	0.6424	0.9316	2811.8
	朴素贝叶斯	0.8911	0.8963	0.8846	3.58
	lstm	0.9835	0.9677	0.9997	2204.8

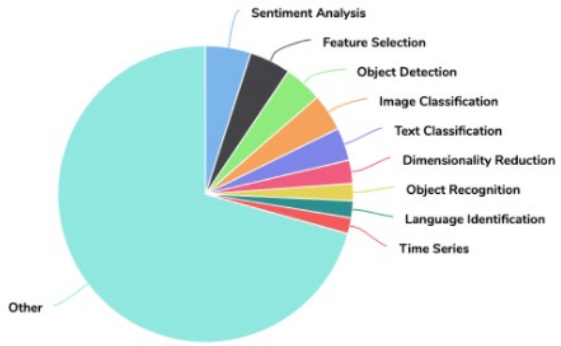
北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

Demo展示

赵旻基

SVM的TASK比例图

Tasks



TASK	PAPERS	SHARE
Sentiment Analysis	43	4.98%
Feature Selection	39	4.52%
Object Detection	36	4.17%
Image Classification	35	4.06%
Text Classification	31	3.59%
Dimensionality Reduction	22	2.55%
Object Recognition	16	1.85%
Language Identification	16	1.85%
Time Series	15	1.74%

SVM

	0.1	1	2	3	4	5	6	7	8	9	...	390	391	392	
0	0.700702	0.349477	0.180986	-0.135617	1.570739	-0.376783	0.177400	0.446059	0.249010	1.801949	...	0.430363	-0.310983	0.701291	-0.186
1	1.275913	0.828773	0.084739	-0.855393	0.459169	0.736649	2.354228	0.497984	0.403887	-1.222761	...	0.135723	-0.190565	1.179150	1.092
2	0.925918	-0.974763	-0.436517	1.068116	0.936932	-1.258674	0.322137	0.477935	-0.424870	0.796654	...	0.300132	-0.944129	0.723659	0.027
3	1.355095	1.497001	-0.059189	0.474190	2.415718	-0.358356	-0.093822	0.052588	-0.904779	-0.526579	...	-1.176677	0.722658	-1.437083	0.341
4	3.420084	-1.160131	0.603855	0.945782	3.761744	-0.860223	1.593925	0.230503	0.643815	-1.052390	...	-1.282169	-1.903865	-0.902999	0.127
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
193300	0.976102	0.416557	2.839105	1.193314	1.198483	-1.441390	0.767203	-1.257607	0.681888	0.023343	...	-0.897009	1.022446	-0.957538	0.215
193301	1.081901	0.385892	0.917241	-0.115451	0.667608	-0.040435	0.382614	0.493456	-0.249297	-0.610998	...	-0.321916	1.008280	0.286167	-0.389
193302	-0.282047	-0.327763	-0.374072	0.444585	0.205536	0.422316	0.577292	-0.395113	0.474245	-0.472313	...	-0.485845	-0.282582	-0.106732	-0.139
193303	2.090122	0.650179	-1.419914	-0.440948	1.423308	0.770389	1.578408	0.478326	0.571018	-0.855511	...	-0.692061	-0.193603	0.027580	1.769
193304	1.388794	1.252615	0.036460	1.466238	-0.002151	-0.405842	-0.939644	-1.201276	0.035427	-0.628891	...	1.271376	0.798437	-0.219180	0.618

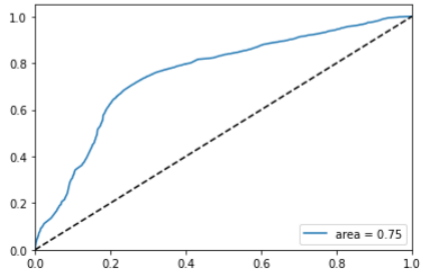
193305 rows × 400 columns

SVM

北京理工大學 BEIJING INSTITUTE OF TECHNOLOGY

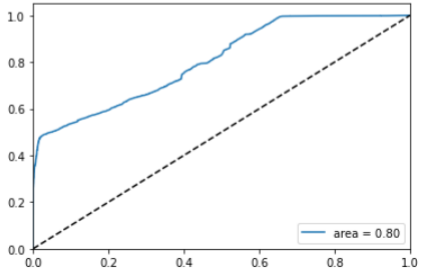
外卖-10k

```
In [101]: print( "Test Accuracy : ",clf.score(x_pca,y))
          Test Accuracy :  0.7244683481701286

In [102]: pred_probas = clf.predict_proba(x_pca)[:,1] #score

          fpr,tpr,_ = metrics.roc_curve(y, pred_probas)
          roc_auc = metrics.auc(fpr,tpr)
          plt.plot(fpr, tpr, label = 'area = %.2f' % roc_auc)
          plt.plot([0, 1], [0, 1], 'k--')
          plt.xlim([0.0, 1.0])
          plt.ylim([0.0, 1.05])
          plt.legend(loc = 'lower right')
          plt.show()
```

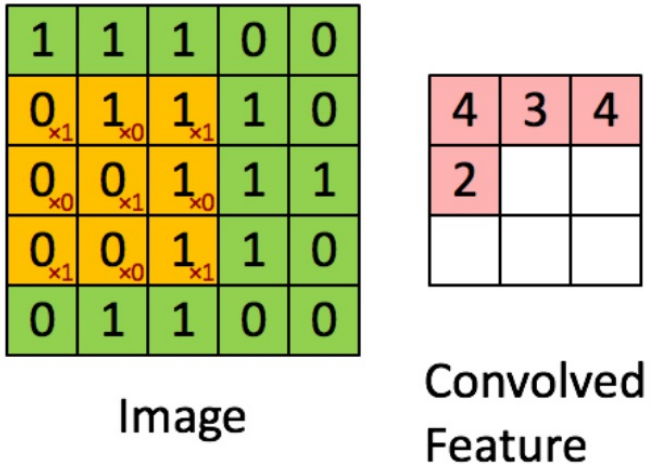


微博-100k

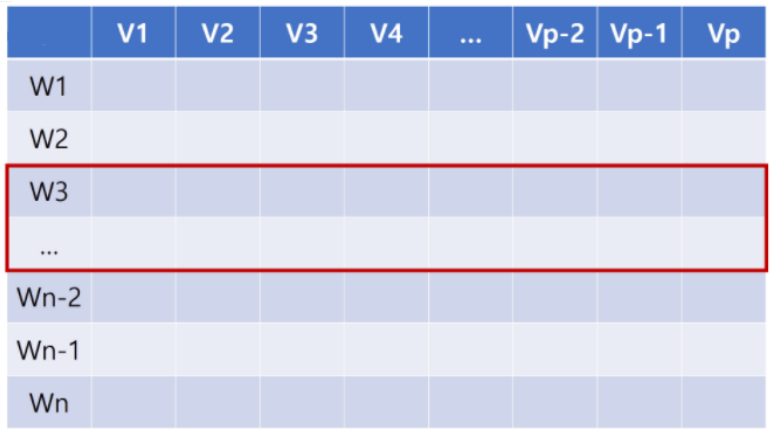
```
In [27]: print( "Test Accuracy : ",clf.score(x_pca,y))
         Test Accuracy :  0.96574

In [28]: pred_probas = clf.predict_proba(x_pca)[:,1] #score

         fpr,tpr,_ = metrics.roc_curve(y, pred_probas)
         roc_auc = metrics.auc(fpr,tpr)
         plt.plot(fpr, tpr, label = 'area = %.2f' % roc_auc)
         plt.plot([0, 1], [0, 1], 'k--')
         plt.xlim([0.0, 1.0])
         plt.ylim([0.0, 1.05])
         plt.legend(loc = 'lower right')
         plt.show()
```



Text CNN

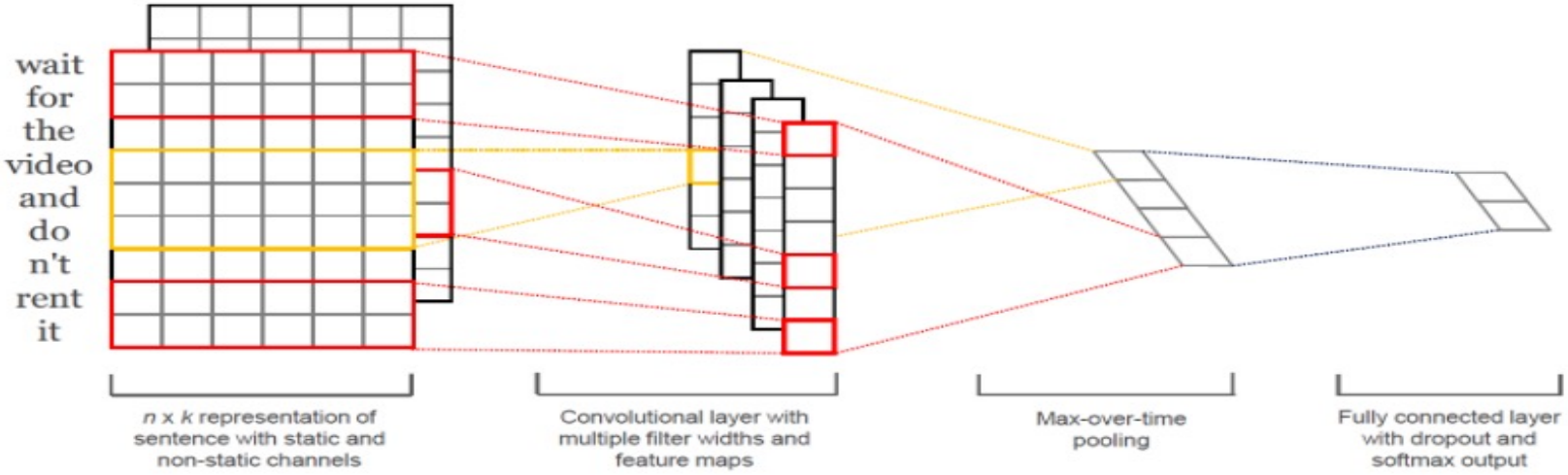


提取局部分图像信息



提取文本的局部信息
- 词出现序列 -
context信息 等

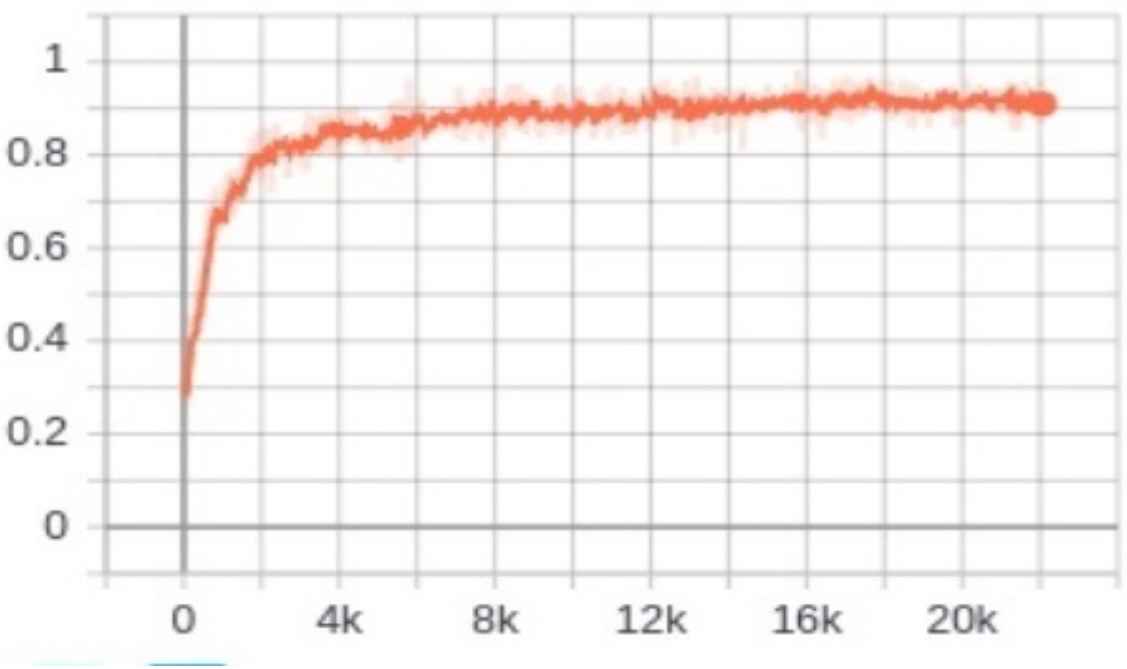
Text CNN



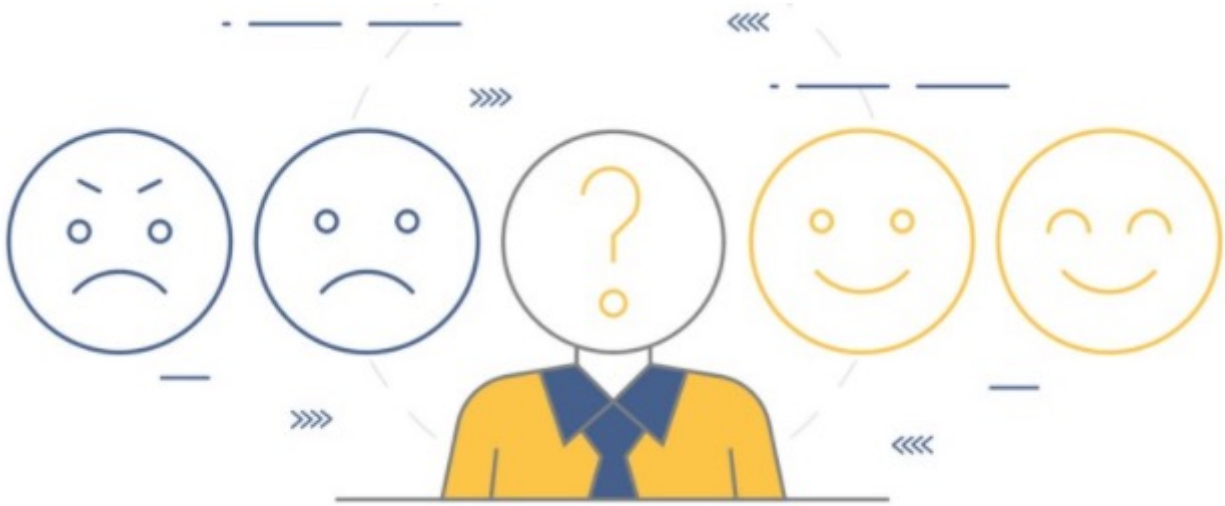
一个文章里n个词，每个词有p维的向量。这里每次运行filter时看两个词。然后把他调整能产生多种N-gram模型

Text CNN

Accuracy / Data



总结



德以明理　学以精工

谢谢