

哈尔滨工业大学（深圳）	小组号	版本号	密级
	1	V4.1	A
文档名称	《电商评论信息抽取&情感分析》项目需求分析报告		发布时间
			2021/5/23

《电商评论信息抽取&情感分析》 项目需求分析报告

课程名称： 软件设计与开发实践 B

学院（系）： 计算机科学与技术学院

专 业： 计算机科学与技术

组长	180310308 王锦成
组员 1	180110625 刘晓轲
组员 2	180110505 胡 聪

2021 年 5 月 23 日

变更历史记录

序号	变更 (+/-) 说明	作者	版本号	日期	批准
1	项目立项	王锦成、刘晓轲、胡聪	V1.0	2021/3/15	王锦成
2	完成总体框架	胡聪	V2.0	2021/4/15	王锦成
3	完善项目背景意义	胡聪	V3.0	2021/4/20	王锦成
4	完善引言部分	胡聪	V3.1	2021/5/01	王锦成
5	完成剩余部分	胡聪	V3.2	2021/5/10	王锦成
6	补充算法描述	胡聪	V4.0	2021/5/21	王锦成
7	修正语言描述问题	王锦成、刘晓轲、胡聪	V4.1	2021/5/23	王锦成

目录

变更历史记录	1
1. 引言	4
1.1 项目背景和意义	4
1.2 需求分析	4
1.2.1 命名实体识别及其常用方法	5
1.2.2 文本表示的数学方法	7
1.3 立项总结	7
1.4 参考资料	8
2. 任务概述	9
2.1 目标	9
2.2 运行环境	10
2.3 算法设计	10
2.3.1 BERT 模型	10
2.3.2 算法流程	13
3. 实验描述	16
3.1 数据集介绍	16
3.2 算法实验结果展示	17
4. 系统设计	19
4.1 系统总体设计	19
4.2 功能划分	20
4.3 性能需求	20
4.3.1 结果准确率	20
4.3.2 时间特性	20
4.4 运行需求	21
4.4.1 用户界面	21
4.4.2 运行环境	21
4.4.3 故障处理	21
4.5 其他需求	21
4.5.1 可使用性	21
4.5.2 可维护性	21

4.5.3 安全保密	22
4.5.4 可移植性	22
5. 项目总结	22
5.1 遇到的问题及解决方案	22
5.2 思考	22

1. 引言

1.1 项目背景和意义

根据国家统计局发布的 2020 年全年社会消费品零售总额的数据情况，2020 年全国社会消费品零售总额达 391981 亿元，其中网上零售额达 117601 亿元，占比约 30.0%。尤其在 2020 年新冠病毒肺炎引发的疫情影响之下，网购成为了大多数消费者的主要消费方式。

在网络消费品零售额高速增长的态势之下，从琳琅满目的商品中挑选出令人满意的商品，成为了消费者面临的困难，而电商平台的评价，成为了消费者判断商品品质的重要依据。调查机构 BrightLocal 的数据显示，超过 80% 的网民相信网络上的购物评价，因此如何从大量的网购评价中提取出关键信息，构建一套完善的商品评级体系，对于提升消费者购物体验来说十分重要。

与此同时，对于商家和平台来说，通过大量分析消费者网购评价，能够总结出所销售商品以及所提供服务中存在的不足之处，以改善产品品质和服务质量，有效提升销售额及消费者满意度。

利用自然语言处理相关技术，从大量真实的消费者评价中找出共性和关键词，判断其情感色彩，既方便消费者进行判断选择，又方便商家进行后续的商品改进，能够实现消费者和商家的双赢，促进消费品零售行业和社会经济的健康发展。

1.2 需求分析

由于在该项目开发过程中已经有完整的数据集文件，方便后续进行数据处理，本项目的开发需求主要集中在对所提供的数据集进行处理上。开发者需要将数据集中的完整语句进行处理，从中提取出语句中所描述的内容、情感色彩、描述对象等。考虑到实际应用的需要，经过处理的数据准确率预期应达到 90~95% 以上。考虑到本项目的开发周期，以及项目的实际需求，本项目预期将不会有 UI 界面，仅有最终处理的结果以及准确率等信息。

本项目的实质是自然语言处理，因此下文会对比一些自然语言处理过程中常用的模型、方法等。

1.2.1 命名实体识别及其常用方法

命名实体识别 (Named Entity Recognition, NER) 是 NLP 中一项非常基础的任务。NER 是信息提取、问答系统、句法分析、机器翻译等众多 NLP 任务的重要基础工具。

实体是指某一概念的实例，例如“人名”是一种概念，或者说是实体类型，那么一个具体的人名，则是“人名”这一概念的一个实体，或者说实体就是某一概念的具象化。而实体识别，则是指从输入的文本中，提取出想要获取的实体类型。

NER 是一种序列标注问题，NER 的过程，就是根据输入的句子，预测出其标注序列的过程，也即给定一系列的特征去预测每个词的标签。

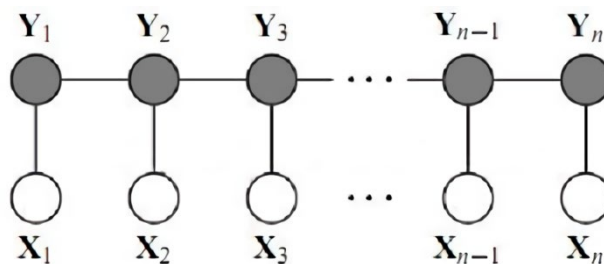


图 1 NER 模型示意图

图中的 X 是语句中每个单词所对应的特征， Y 是单词所对应的标签，目前命名实体识别常用深度学习算法进行特征的选择和提取，使用条件随机场算法 (Conditional Random Field, CRF) 进行特征处理和标签预测。在深度学习的神经网络中，由于神经网络只接受数字，不接受字符串等其他数据结构，因此，我们需要使用转换工具将输入的词句转换为词向量（词向量模型）。转换词向量后的训练过程最好能够使用足够庞大的数据集，或者使用近年较为流行的各式预训练模型。

NER 过程中的常用方法有：

(1) HMM 和 CRF 等传统机器学习算法：条件随机场 (Conditional Random Field, CRF) 是 NER 目前的主流模型。它的目标函数不仅考虑输入的状态特征函数，而

而且还包含了标签转移特征函数。CRF 的优点在于其为一个位置进行标注的过程中可以利用丰富的内部及上下文特征信息；

(2) LSTM+CRF：将 Token 从离散 One-Hot 表示映射到低维空间中成为稠密的 Embedding，随后将句子的 Embedding 序列输入到 RNN 中，用神经网络自动提取特征，Softmax 来预测每个 Token 的标签，采用 LSTM 作为特征抽取器，再接一个 CRF 层来作为输出层；

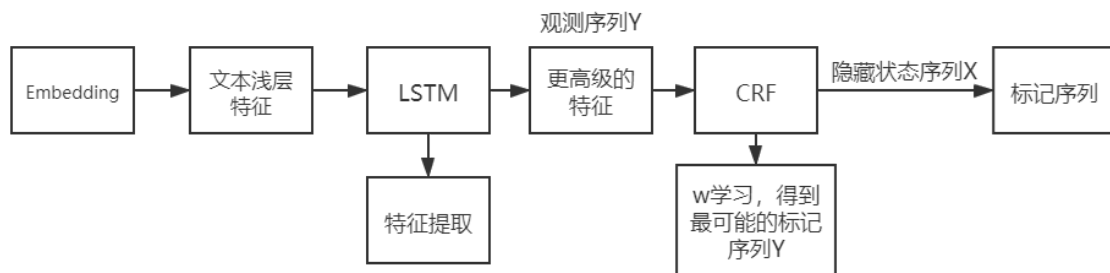


图 2 LSTM+CRF 过程示意图

(3) CNN+CRF：采用 CNN 作为特征抽取器，再接一个 CRF 层作为输出层。对于序列标注来讲，普通 CNN 有一个不足，就是卷积之后，末层神经元可能只是得到了原始输入数据中一小块的信息。而对 NER 来讲，整个输入句子中每个字都有可能对当前位置的标注产生影响，即所谓的长距离依赖问题。CNN 虽然在长序列的特征提取上有弱势，但是 CNN 模型可有并行能力，有运算速度快的优势。正常 CNN 的 Filter，都是作用在输入矩阵一片连续的区域上，不断 Sliding 做卷积。Dilated CNN 为这个 Filter 增加了一个 Dilation Width，作用在输入矩阵的时候，会跳过所有 Dilation Width 中间的输入数据；而 filter 本身的大小保持不变，这样 Filter 获取到了更广阔的输入矩阵上的数据，看上去就像是“膨胀”了一般。具体使用时，Dilated Width 会随着层数的增加而指数增加。这样随着层数的增加，参数数量是线性增加的，而 Receptive Field 却是指数增加的，可以很快覆盖到全部的输入数据。膨胀卷积的引入，使得 CNN 在 NER 任务中，能够兼顾运算速度和长序列的特征提取。IDCNN 对输入句子的每一个字生成一个 Logits，这里就和 biLSTM 模型输出 Logits 完全一样，加入 CRF 层，用 Viterbi 算法解码出标注结果；

(4) BERT+ (LSTM) +CRF：BERT 中蕴含了大量的通用知识，利用预训练好的

BERT 模型，再用少量的标注数据进行 FINETUNE 是一种快速的获得效果不错的 NER 的方法。

1.2.2 文本表示的数学方法

要将一篇文本用数学语言表示，有以下多种方法：

- (1) 基于 One-Hot、TF-IDF、TextRank 等的 bag-of-words 方法；
- (2) 主体模型方法：LSA (SVD)、pLSA、LDA；
- (3) 基于词向量的固定表征方法：Word2vec、fastText、glove；
- (4) 基于词向量的动态表征方法：ELMo、GPT、BERT；

上述四种类型的方法是自然语言处理领域较为常用的文本表示方法。文本由单词构成，对于词向量，One-Hot 最为简单，但是却存在语义鸿沟问题，而 LSA (SVD) 又存在计算复杂度较高等问题。为了解决这些问题，分布式假设方法诞生了，简单来说分布式假设是指相同上下文语境的词语具有相似含义，据此产生了 Word2vec、fastText、glove 等方法。

上述提到的三种方法得到的词向量都是固定表征的，对于歧义问题无法解决，例如“算账”一词，其有基本含义“计算账目”，同时又有引申含义“吃亏或失败后与人较量”，传统方法无法区分或很难区分同一词汇的基本义、引申义、比喻义等不同意义。为了解决这类问题，由此诞生了 ELMo、GPT、BERT 等语言模型的动态表征方法。

特征提取器：ELMo 采用 LSTM 进行提取，而 GPT 和 BERT 则采用 Transformer 进行提取。很多任务表明 Transformer 的特征提取能力强于 LSTM；

单/双向语言模型：ELMo 和 BERT 都采用双向语言模型，但 ELMo 采用的是两个单项语言模型进行方向相反的拼接，相较于 BERT 的处理能力弱。

1.3 立项总结

经过论文阅读和小组讨论，本项目决定采用 BERT 预训练模型，以此来完成本项目的任务。本小组的目标是将数据集进行训练和处理，最终从大量购物网站

评论信息中抽取出商品信息、情感色彩等有用信息。

本项目开发过程中，由刘晓轲负责主要代码的编写，王锦成和胡聪负责代码开发的辅助工作。文档主要由胡聪完成，KeyNote 和其他相关资料主要由王锦成完成，汇报工作主要由刘晓轲完成。

1.4 参考资料

- [1] Devlin J , Chang M W , Lee K , et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.
- [2] Clark K , Khandelwal U , Levy O , et al. What Does BERT Look At? An Analysis of BERT's Attention[C]// Acl Workshop Blackboxnlp: Analyzing & Interpreting Neural Networks for Nlp. 2019.
- [3] Xue K, Zhou Y, Ma Z, et al. Fine-tuning BERT for joint entity and relation extraction in Chinese medical text[C]//2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2019: 892-897.
- [4] Cai J, Li J, Li W, et al. Deep learning model used in text classification[C]//2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). IEEE, 2018: 123-126.
- [5] Khatua A, Khatua A, Cambria E. A tale of two epidemics: Contextual Word2Vec for classifying twitter streams during outbreaks[J]. Information Processing & Management, 2019, 56(1): 247-257.
- [6] Jettakul A, Thamjarat C, Liaowongphuthorn K, et al. A comparative study on various deep learning techniques for Thai NLP lexical and syntactic tasks on noisy data[C]//2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE). IEEE, 2018: 1-6.
- [7] Sun S, Cheng Y, Gan Z, et al. Patient Knowledge Distillation for

- BERT Model Compression[C]//EMNLP/IJCNLP (1). 2019.
- [8] Tenney I, Das D, Pavlick E. BERT Rediscovered the Classical NLP Pipeline[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 4593-4601.
- [9] Guo B, Zhang C, Liu J, et al. Improving text classification with weighted word embeddings via a multi-channel TextCNN model[J]. Neurocomputing, 2019, 363: 366-374.
- [10] Yadav V, Bethard S. A Survey on Recent Advances in Named Entity Recognition from Deep Learning models[C]//Proceedings of the 27th International Conference on Computational Linguistics. 2018: 2145-2158.
- [11] Akbik A, Bergmann T, Vollgraf R. Pooled contextualized embeddings for named entity recognition[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019: 724-728.
- [12] Li J, Sun A, Han J, et al. A survey on deep learning for named entity recognition[J]. IEEE Transactions on Knowledge and Data Engineering, 2020.
- [13] Sohrab M G, Miwa M. Deep exhaustive model for nested named entity recognition[C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018: 2843-2849.

2. 任务概述

2.1 目标

随着电商行业的发展，消费者对商品的品质产生了更高的追求。我们在线上购买商品时，往往会参考有过该商品历史购买记录的消费者的评论。通过分析这

些评论，一方面可以使消费者方便地选择出自己心仪的商品；另一方面也可以让商家更好地分析自己商品的优缺点，从而调整销售策略，具有非常大的实用价值和应用场景。

综合来看，本项目的任务就是给一条文本，然后在该文本中抽取四元组：商品属性特征、消费者观点、情感极性分类和属性种类分类。

评价标准：计算四元组 F1 值。

2.2 运行环境

软件环境	软件名称	版本	备注
	操作系统	Windows 10 专业版 1803 17134.950	
	应用软件	PyCharm Community Edition 2018.3.5	
	深度学习框架	Pytorch	
硬件环境	硬件名称	规格/数量	备注
	CPU	Intel(R) Xeon(R) W-2123 @3.60GHz	
	内存	32.0GB	
	硬盘	SUMSUNG 256GB	
	显卡	NVIDIA GeForce RTX 2080 Ti	

表 1 运行环境需求表

2.3 算法设计

2.3.1 BERT 模型

谷歌的研究人员提出了一种新的称为 BERT 的语言表示模型，BERT 代表来自 Transformer 的双向编码器表示（Bidirectional Encoder Representations from Transformers）。不同于最近的语言表示模型，BERT 旨在通过联合调节所有层中的左右上下文来预训练深度双向表示。因此，只需要一个额外的输出层，

就可以对预训练的 BERT 表示进行微调，从而为广泛的任务（比如回答问题和语言推断任务）创建最先进的模型，而无需对特定于任务进行大量模型结构的修改。

BERT 的概念很简单，但实验效果很强大。它刷新了 11 个 NLP 任务的当前最优结果，包括将 GLUE 基准提升至 80.4%（7.6% 的绝对改进）、将 MultiNLI 的准确率提高到 86.7%（5.6% 的绝对改进），以及将 SQuAD v1.1 的问答测试 F1 得分提高至 93.2 分（提高 1.5 分）——比人类表现还高出 2 分。

现有的两种方法可以将预训练好的语言模型表示应用到下游任务中：基于特征的和微调。基于特征的方法，如 ELMo，使用特定于任务的模型结构，其中包含预训练表示作为附加特征。微调方法，如生成预训练 Transformer (OpenAI GPT) 模型，然后引入最小的特定于任务的参数，并通过简单地微调预训练模型的参数对下游任务进行训练。在之前的工作中，两种方法在预训练任务中都具有相同的目标函数，即使用单向的语言模型来学习通用的语言表达。

BERT 的模型结构是一个多层双向 Transformer 编码器。为了方便比较，BERT 选择了与 OpenAI GPT 一样的模型大小。然而，重要的是，BERT Transformer 使用的是双向的自注意力，而 GPT Transformer 使用的是受限的自注意力，每个标记只能关注其左边的语境。我们注意到，在文献中，双向 Transformer 通常被称为“Transformer 编码器”，而只有标记左侧语境的版本由于可以用于文本生成而被重新定义为“Transformer 解码器”。BERT、OpenAI GPT 和 ELMo 之间的比较如图 3 所示。其中，预训练模型结构的不同。BERT 使用双向 Transformer。OpenAI GPT 使用从左到右的 Transformer。ELMo 使用独立训练的从左到右和从右到左的 LSTM 的连接来为下游任务生成特征。其中，只有 BERT 表示在所有层中同时受到左右语境的制约。

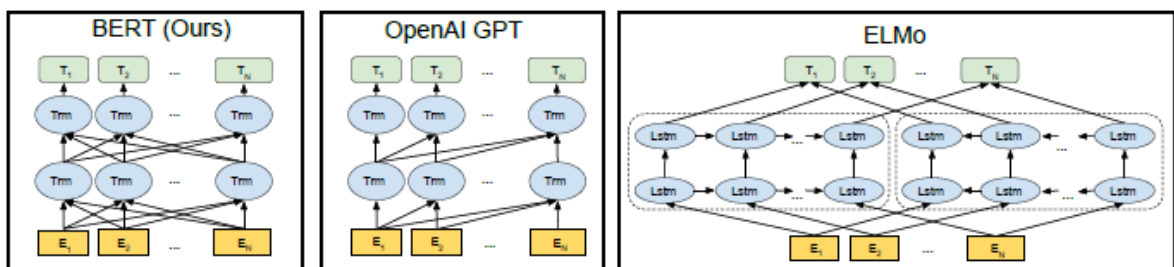


图 3 模型之间的对比

BERT 模型的输入表示能够在标记序列中清楚地表示单个文本句子或一对文本句子。在整个工作中，“句子”可以是连续的任意跨度的文本，而不是实

际语言意义上的句子。“序列”是指输入到 BERT 的标记序列，它可以是单个句子，也可以是两个句子组合在一起。通过把给定标记对应的标记嵌入、句子嵌入和位置嵌入求和来构造其输入表示。

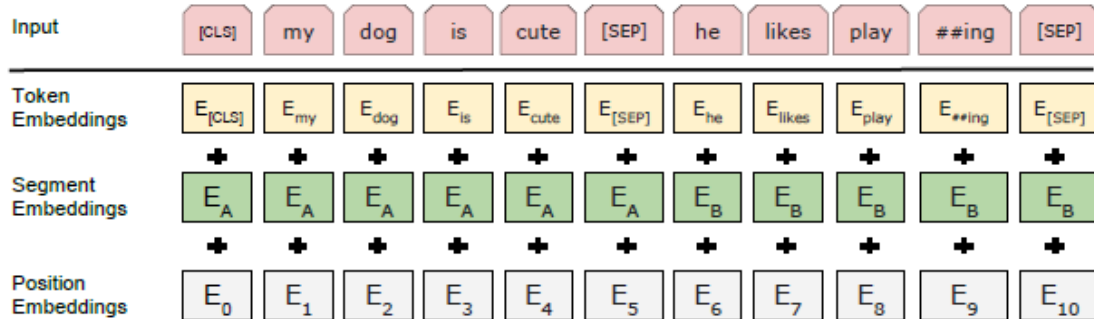


图 4 BERT 的输入表示

BERT 是传统语言模型的一种变形，而语言模型 (Language Model, LM) 做的事情就是在给定一些词汇的前提下，去估计下一个词汇出现的几率分布。给定前 t 个在字典里的词汇，语言模型要去估计第 $t+1$ 个词汇的几率分布 P 。

$$P(x^{(t+1)} | x^t, \dots, x^{(1)}),$$

其中， $x^{(t+1)}$ 可以是词向量 $V = \{w_1, \dots, w_{|V|}\}$ 中的任何单词

为什么需要训练一个语言模型？（1）无监督数据无限大，如在 ImageNet 中，需要人工标注数据，而训语言模型时网络上所有的文本都是潜在的数据集（BERT 预训练使用的数据集有 33 亿字，其中包含维基百科等）；（2）优秀的语言模型能够学会语法结构、解读语义甚至指代消解，通过特征提取能够更有效率地训练下游任务并提升其表现；（3）减少处理不同自然语言处理任务中所需要的工程建筑成本。

以往为了解决不同的 NLP 任务，我们会为该任务设计一个最适合的神经网络架构并做训练。这就是近来 NLP 领域非常流行的两阶段迁移学习，先预先训练出一个对自然语言有一定“理解”的通用模型，再将该模型拿来特征提取或是下游的监督式任务。

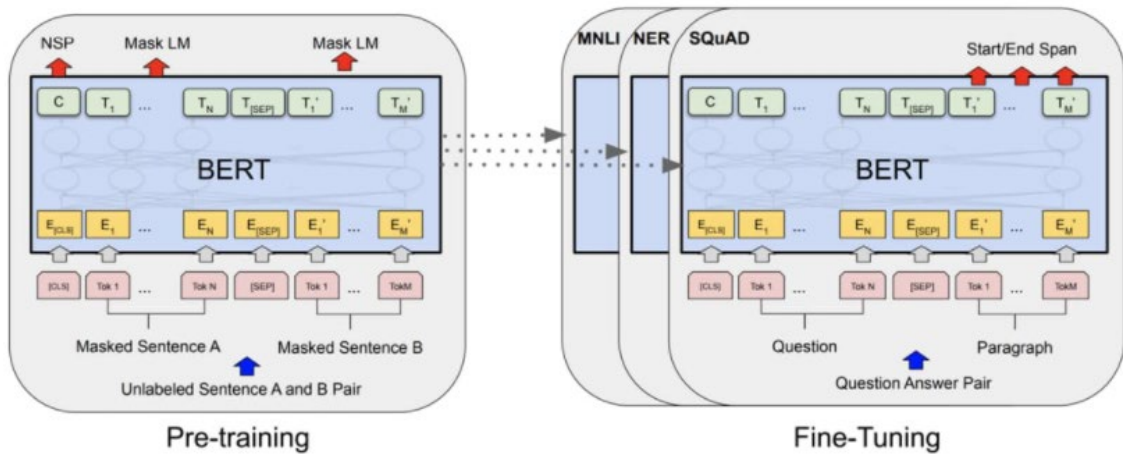


图5 两阶段迁移学习在BERT下的应用：使用预先训练好的BERT对下游任务做fine tuning

Google 在预训练 BERT 时让它同时进行两个任务：完形填空（Masked Language Model）和判断第 2 个句子在原始文本中是否跟第 1 个句子相接（Next Sentence Prediction, NSP）。

2.3.2 算法流程

（1）数据预处理：因为计算机仅能处理数值型数据，所以在进行数据处理时，需要对输入程序中的文本数据（数据集中的 Labels 和 Reviews）进行预处理，使得其能够在后续的训练过程中成为可用的输入数据。对于 XX_Labels，系统将其作为训练的对照对象，用于计算后续模型训练结果中的精确率（Precision）、召回率（Recall）和 F1 值（F-Measure）。在将数据集文件读入程序后，系统将根据 Labels 文件中的数据内容及其含义转换为词向量，并将其作为准确值。数据集中的 Reviews 数据将用于训练和测试。在使用时，首先使用分词工具对其进行按字分词，然后使用随机划分工具将其划分为训练集和测试集，最后根据后续训练需要的输入数据格式，通过 DataSet 对相关数据进行封装和装载，得到后续执行所需要的输入数据；

（2）BertEmbeddings 层：该层的输入为装载过后的文本数据，BERT 模型通过计算得到它的三种 Embeddings：Word_Embedding、Position_Embedding 和 Token_Embedding。

在获取上述的三个 Embeddings 后，将其相加可以得到句子的表示向量，将结果输入到 Layer_Normer 层和 Dropout 层进行简单的数字处理，表示返回结果。

这个处理结果将作为后续处理的输入；

	这	价	位	非	常	给	力	。
word_embedding	6821	817	855	7478	2382	5314	1213	8024
position_embedding	0	1	1	1	1	1	1	0
token_embedding	0	1	3	5	6	6	7	0

表 2 计算结果示例

(3) BertEncoder 层：该层实际上是由 12 层堆叠起来的，由将上一层的输出作为下一层的输入的 BertLayer 层组成。在具体实现中，使用循环遍历来实现堆叠。每个 BertLayer 层都由 Attention 层、中间层（Intermediate 层）和输出层（Output 层）组成。

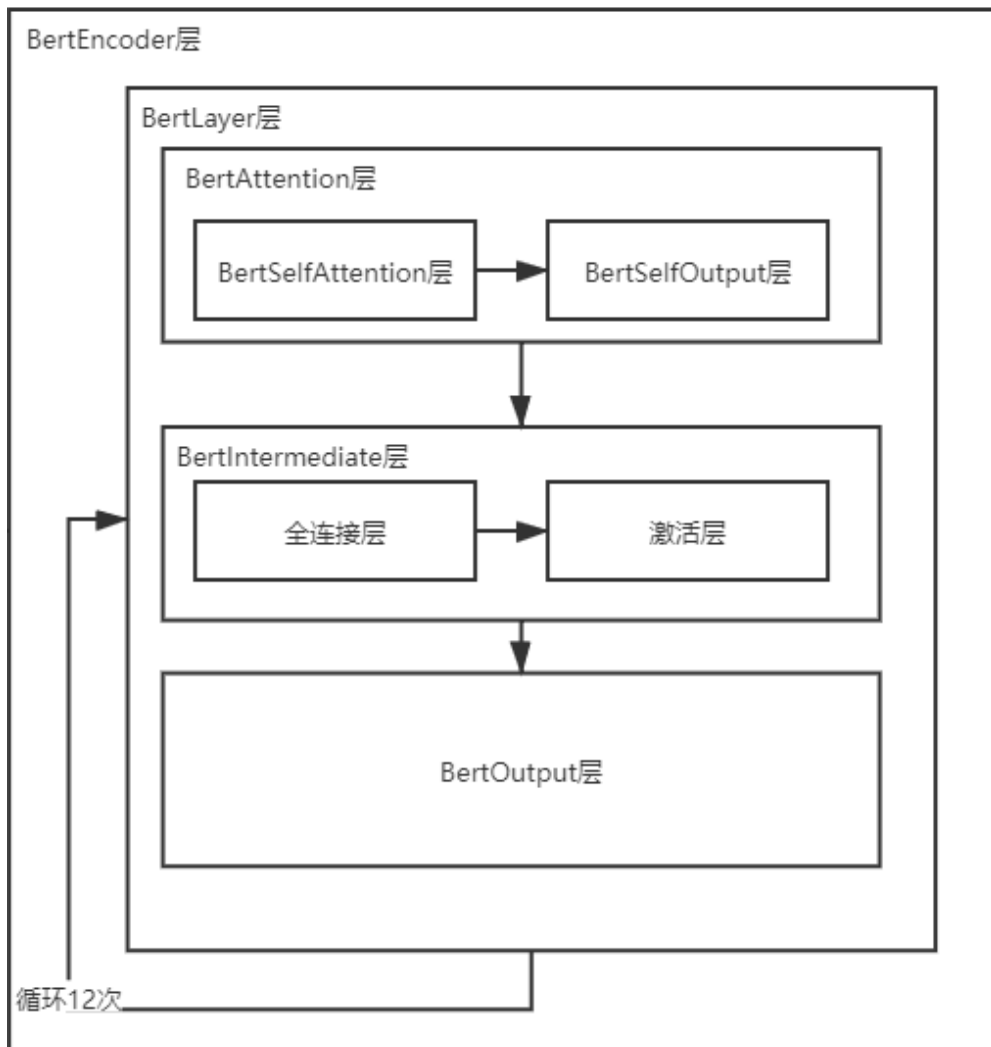


图 6 BertEncoder 层结构示意图

在 Attention 层中，数据首先进入 BertSelfAttention 层进行计算，由 BertSelfOutput 层进行输出计算。系统在这一层的训练目标是获得三个矩阵

Query、Key 和 Value。其中，Query 和 Key 用来确定权重，结果乘以 Value 来确定注意力的大小。系统通过输入的向量矩阵训练预测 Query、Key，一句话的注意力权重矩阵计算公式如下：

$$Attention = Query \times Key$$

对权重因子进行简单地数据处理后，我们用 SoftMax 对这个权重矩阵进行处理，得到结果；最后将上述处理的结果与 Value 相乘，获得最终的注意力：

$$FinalAttention = Attention \times Value$$

在具体使用中，我们使用线性分类器进行各个矩阵的预测。最终，我们的训练结果是三个向量矩阵 Query，Key 和 Value，分别与我们需要提取的这个句子的 Aspect，Opinion，Category 和 Polarities 对应 Attention 层输出的结果进入由一个全连接层和一个激活层组成的 BertIntermediate 层，输出结果。

最后，BertIntermediate 输出的结果进入 BertOutput 层。这个 BertOutput 层的结构与上文 BertSelfOutput 层相同但参数有所区别，经过这一层的处理，我们基本完成 BertLayer 的一次执行。后续，上层 BertLayer 的输出作为下层 BertLayer 的输入，进行一共 12 次的循环。

（4）整理输出和 Pooler 层：系统取出最后一层 BertLayer 的输出，经过 pooler 层，获得最终输出模型的结果。在 Pooler 层中，取出每一句的第一个单词，做全连接和激活。得到的输出可以用来分类等下游任务；

（5）最终输出的获取和评分计算：在获取 Bert_Model 的输出后，根据问题需求从中抽取序列标注和情感分析结果，使用之前构建的解码器对其进行解码，将数值数据解码重新成为中文文本数据，与之前得到的准确结果进行对比，计算出精确率（Precision）、召回率（Recall）和 F1 值（F-Measure），如果在每次训练中，当前模型结果的 F1 评分大于历史最佳评分，则保存模型，并输出最终结果。

3. 实验描述

3.1 数据集介绍

本项目的训练集由两类商品——化妆品和笔记本电脑组成，同时提供了大量笔记本电脑的评论，测试集为笔记本电脑的评论。表 2 为数据的情况分析：

数据集名称 数据数量	Train Laptop	Test Laptop	Train Laptop Corpus	Train Makeup
样本数量	871	2237	26481	13663
平均长度	26.6	21.3	44.1	21.4
平均四元组数量	2.8	—	—	1.9

表 3 数据集情况分析

在数据集中，Reviews 的数据是评论的数据，ID 是评论的编号，Reviews 是评论的内容。

```
id, Reviews
1, 看到***~买的，发货速度挺快
2, 卸妆很干净温和不刺激，用完脸滑滑的，舒服
3, 买来做水疗用的，比我在实体店买的便宜，用完再来。
4, 这刀片没法用刮不下来，不知道好评是不是刷出来的，想回复问题都不让&hellip;&hellip;
5, 中午下单下午送到，太快了，支持**。
6, 第三次购买，一如既往的好用👍👍
7, 还没试，试了再来评价，看着很小罐
8, 味道很不错，以前用过，现在又换回来了，期待一如既往的好用
9, 价格实惠，质量好，样子也简单。喜欢
10, 暂时用着可以😊
11, 这款价格经常变动越来越高了啊
```

图 7 ID 和 Reviews 之间的对应情况

其中，Labels 的数据 ID 是对应某条评论的 ID，商品属性 (aspect) 和观点

(opinion) 其情感极性 (sentiment polarity) 和属性种类 (category)。

id	AspectTerms	A_start	A_end	OpinionTerms	O_start	O_end	Categories	Polarities
1	发货速度	9	13	挺快	13	15	物流	正面
2	卸妆	0	2	很干净	2	5	功效	正面
2	_			温和	5	7	使用体验	正面
2	_			不刺激	7	10	使用体验	正面
2	_			滑滑的	14	17	使用体验	正面
2	_			舒服	18	20	使用体验	正面
3	_			便宜	16	18	价格	正面
4	刀片	1	3	没法用	3	6	其他	负面

表 4 数据集拆分

基于对数据集的分析，我们总结出数据之间的关系有：（1）一个商品属性特征对应多个消费者观点。例如：电脑性能：运行流畅，不卡顿；（2）多个商品属性特征对应一个消费者观点。例如：玩游戏办公都可以。

3.2 算法实验结果展示

图 8 为输入测试样本的观点挖掘的结果。图 9 为提供的标注好的数据集。

```

01,_, , ,很舒服, 5, 8, 使用体验, 正面
02,_, , ,很香, 0, 2, 气味, 正面
03,_, , ,太油腻了, 0, 4, 使用体验, 负面
03,_, , ,洗不干净, 7, 11, 功效, 负面
04,作用, 0, 2, 不大, 2, 4, 功效, 负面
05,_, , ,挺好的, 21, 24, 整体, 正面
06,泡沫, 19, 21, 丰富, 21, 23, 使用体验, 正面
06,_, , ,很好用, 15, 18, 整体, 正面
06,泡沫, 19, 21, 细腻, 24, 26, 使用体验, 正面
07,_, , ,还不错, 6, 9, 整体, 正面
08,_, , ,挺好的, 0, 3, 整体, 正面
08,物流, 4, 6, 挺快的, 7, 10, 物流, 正面
09,_, , ,很喜欢, 6, 9, 整体, 正面
09,_, , ,很好用, 9, 12, 整体, 正面
09,物流, 13, 15, 很快, 15, 17, 物流, 正面
09,物流, 13, 15, 给力, 17, 19, 物流, 正面
09,_, , ,五星好评, 19, 23, 整体, 正面
10,_, , ,好评, 38, 40, 整体, 正面
10,价格, 0, 2, 实惠, 2, 4, 价格, 正面
10,活动, 6, 8,_, , ,价格, 正面
10,_, , ,便宜多了, 33, 37, 价格, 正面
11,_, , ,挺好的, 0, 3, 整体, 正面
11,_, , ,不干, 19, 21, 使用体验, 正面
11,_, , ,顺滑, 22, 24, 使用体验, 正面
12,_, , ,很好用, 3, 6, 整体, 正面

```

图 8 观点挖掘结果

```
0, 刀片, 一点不锋利, 其他, 负面
0, 丿, 不好, 整体, 负面
1, 香味, 淡淡的, 气味, 正面
1, 丿, 不干, 使用体验, 负面
1, 丿, 很舒服, 使用体验, 正面
2, 香味, 很香, 气味, 正面
3, 丿, 太油腻了, 使用体验, 正面
4, 丿, 作用不大, 整体, 负面
5, 发货, 蛮快的, 物流, 正面
5, 丿, 挺好的, 整体, 正面
6, 丿, 很好用, 整体, 正面
6, 丿, 好, 整体, 正面
7, 丿, 还不错, 整体, 正面
8, 物流, 挺快的, 物流, 正面
8, 丿, 挺好的, 整体, 正面
9, 物流, 很快, 物流, 正面
9, 丿, 很喜欢, 整体, 正面
9, 丿, 很好用, 整体, 正面
10, 价格, 实惠, 价格, 正面
10, 丿, 好评, 整体, 正面
10, 活动, 丿, 价格, 正面
11, 丿, 挺好的, 整体, 正面
11, 丿, 不干, 使用体验, 正面
11, 丿, 顺滑, 使用体验, 正面
12, 效果, 很温和, 使用体验, 正面
```

图 9 标注好的数据集

其中，图 8 中的结果能够识别出评论中的各个词的含义，满足了最开始的预期，对于一句话多对多个方面进行评价的情况，也能够进行准确判断，并且将关键词等信息从中提取出来。

经过多次训练，我们得到了最终的训练结果。

```
precision:0.883459 recall:0.854312 f1:0.868353
```

图 10 最终结果

可以看到，我们得到的数据与标准答案的拟合情况较为理想。但是由于数据在词向量转换过程中给存在一定的效率损失，这使得准确率会出现较大幅度的下降情况。与此同时，由于时间有限，我们的训练过程进行的时间还不够长，如果训练时间能够满足，准确率能够得到进一步的提高，更加接近标准答案。

4.2 功能划分

本系统功能模块划分图如下：

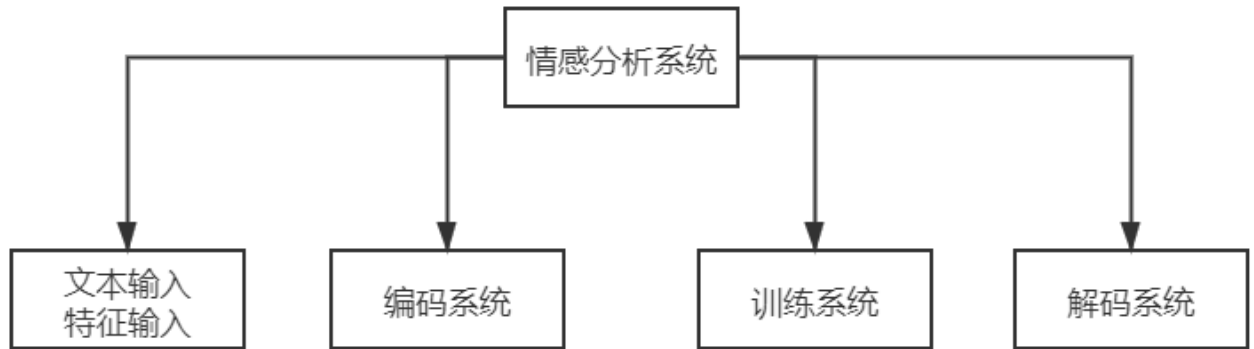


图 13 系统功能模块划分图

文本输入、特征输入：对输入的文本进行 embedding，提取特征；

编码系统：用 BERT 的预训练模型；

训练系统：采用 Bert 的 Attention 机制，进行模型训练。

解码系统：使用训练好的模型进行解码，完成整个信息抽取的任务。

4.3 性能需求

4.3.1 结果准确率

通过本项目训练的数据集最终结果应该达到 90%以上。在时间和硬件性能允许的条件下，预期应继续提高准确率。

4.3.2 时间特性

由于本项目没有用户可操作性，对于时间相关特性没有特别需求。考虑到实际情况，对于数据集的单次训练时间应该控制在 3~5 分钟左右。

4.4 运行需求

4.4.1 用户界面

考虑到项目的实际情况，用户只需要最终的数据结果以及准确率相关数据，因此系统中并没有提供一个图形化的用户界面。

4.4.2 运行环境

本系统运行在 Windows 10 (20H2) 操作系统中，在没有高性能 GPU 的设备上能够正常运行，但如果用户拥有高性能 GPU，则获取最终结果的速度将得到明显提升。

4.4.3 故障处理

一般情况下，用户在进行正确的环境配置后，运行本系统时不会产生故障，但在开发设计本系统时，应该考虑到用户可能会进行的非法操作，对其进行相应的约束，同时能避免系统在用户进行非法操作后进入故障状态。

4.5 其他需求

4.5.1 可使用性

本系统应保证用户能够使用该系统实现预期功能，同时应该在不同的系统环境下都具有较高的运行兼容性。

4.5.2 可维护性

系统开发时应该首先对代码结构进行相应的规划，其次在开发过程中应该保持良好的代码注释习惯、变量命名习惯等，以确保在后续的二次开发中后续开发者能够维护和修改本项目代码。

4.5.3 安全保密

由于该系统并不会用于实际生产环境中，本项目安全保密级别较低。

4.5.4 可移植性

本系统应可以方便快捷地被移植到 Linux 和 macOS 等不同的操作系统平台中。

5. 项目总结

5.1 遇到的问题及解决方案

本项目在开发中遇到了以下问题：（1）组员对于 Python 基础知识和相关框架掌握不太熟练，上手 Python 项目需要大量的时间；（2）组员对于自然语言处理相关知识无基础，或基础知识掌握不深；（3）本学期课程安排导致组员共同开发的时间不足，很多时候进度没有办法进行统一；（4）项目开发进度与预期不符，速度偏慢。

对于以上问题，我们设计了如下解决方案进行处理：（1）采取自学和互相帮助沟通交流的方法，让组员迅速提高编程基础，在开发阶段开始前能够熟练掌握 Python 知识；（2）通过阅读论文和互联网上相关实际案例，增强对于自然语言处理相关知识的理解，通过实例了解系统开发的大致流程；（3）合理安排会议时间，小组成员共同努力进行项目的开发；（4）取消了原定的用户界面开发，使得组员能够将时间更多聚焦于项目的核心功能上。

5.2 思考

《软件设计开发与实践 B》是一门综合性较强的课程，在应用本科前三年学习到的计算机相关知识的同时，又加深了我们对于计算机学科的进一步理解，对

于一个完整的软件系统开发流程有了更加深入的认识。设计开始阶段必须明确设计的目的与需求分析,在进行总体设计时应该兼顾全局,能够意识到后续开发过程中可能出现的需求调整。系统的架构是十分重要的,需要对系统设计、模块设计进行准确地定义,才能在后续的编码阶段更加轻松和方便。

此外,本次课程设计也提高了我们的团队合作精神,在软件系统的开发过程中需要集思广益,结合更多人的创新想法,在开发过程中,如果组员出现了不同的意见,我们会进行集中讨论,商量出更加合理的问题解决方案。

自然语言处理是近期热度较高的问题,深度学习为自然语言处理带来了巨大的技术突破,它的广泛应用极大地改变了人们的日常生活。当深度学习和其他认知科学、语言学结合时,或许可以发挥出更大的威力,解决语义理解问题,带来真正的“智能”。尽管深度学习在自然语言处理的各个任务中取得了巨大成功,但若大规模投入使用,仍然有许多研究难点需要克服。但是,随着深度学习的不断研究深入,在不久的将来,自然语言处理领域将会取得更多研究成果和发展。