# BERT文本分类系统

摘要

人工智能发展到今天，互联网上存在的各种短文本新闻数不胜数，如何对这些短文本新闻进行有效的归类，在短文本有限的特征里学习到足够多的信息，是一项富有挑战的工作。

基于此，我尝试了一种基于BERT模型的中文短文本新闻分类算法，该算法使用自然语言处理领域中的集大成者——BERT(Bidirectional Encoder Representations from Transformers)预训练模型，在句子级别表达短文本的特征，然后将获得的特征向量对下游文本分类任务进行微调，解决我们的分类问题，基于BERT模型的短文本分类算法能够很好的表示短文本的特征，也为其他自然语言处理下游研究提供了参考。

关键词：文本分类；BERT；预训练模型；

# 1 引言

文本分类的方法主要有三种：基于规则方法，基于机器学习的统计方法，基于深度学习方法。基于规则的方法使用一组预先定义的规则将文本分成不同的类别，这些方法需要对领域有深入的了解，而且系统很难维护，所以渐渐淡出了人们视线，基于机器学习的方法学习根据过去对数据的观察进行分类。使用先前标记的示例作为训练数据，机器学习算法可以学习文本片段及其标签之间的内部联系。因此，基于机器学习的方法可以检测数据中的隐藏模式，具有更高的可扩展性，并可应用于各种任务。流行的分类算法包括Naive Bayes、支持向量机、隐马尔可夫模型、梯度提升和随机森林等。

2012年，AlexNet[1]，一个基于深度学习的模型，在ImageNet竞赛中一骑绝尘，范式开始发生转变。从那时起，深度学习模型被广泛应用于计算机视觉和自然语言处理领域，改善了现有的技术水平。这些模型试图学习特征表示并以端到端的方式执行分类（或回归）。它们不仅能够发现数据中的隐藏模式，而且在应用程序之间的可转移性更强。这些模型正在成为近年来各种文本分类任务的主流框架。

我们根据他们的神经网络结构将这些自然语言处理文本分类工作分成几类，如基于循环神经网络（RNNs）的模型、卷积神经网络（CNNs）、注意力机制、Transformer[2]以及图网络等等。

最早是基于前馈神经网络的模型，将文本视为一个词袋，对于每个单词，他们使用Word2vec[3]或Glove[4]这样的嵌入模型学习一个向量表示，将向量和或者嵌入的平均值作为文本的表示，通过一个或多个前馈层，即多层感知器（MLPs），然后使用诸如Logit模型分类器、朴素贝叶斯分类器或者深度平均网络（DAN）[5]这样的分类器对最终层的表示执行分类。此后Joulin等人[6]提出了一个简单有效的文本分类器FastText。Le和Mikolov[7]提出Doc2vec，它使用一种无监督的算法来学习可变长度文本的固定长度特征表示，如句子、段落和文档。

基于RNN的模型，将文本视为一系列单词，用于捕获单词依赖关系和文本结构以进行文本分类。在RNN的许多变体中，长短期记忆（LSTM）是最流行的体系结构，它被设计用来更好地捕捉长期依赖关系。Tai等人[8]开发了一个Tree-LSTM模型，将LSTM概括为树型结构的网络类型，以学习丰富的语义表示。Zhu等人[9]还将链结构的LSTM扩展到树结构。Liu等人[10]使用多任务学习来训练RNNs利用来自多个相关任务的标记训练数据。Zhou等人[11]集成了一个双向LSTM（Bi-LSTM）模型和二维最大池化来捕获文本特征。

基于CNN的模型，训练识别文本中的模式，如关键短语，用于分类，RNN被训练来识别不同时间的模式，而CNN则学习识别不同空间的模式[12]。第一个基于CNN的文本分类模型是Kalchbrenner等人提出的[13]。后来，Kim[14]提出了一个比DCNN更简单的基于CNN的文本分类模型。字符级CNN也被用于文本分类[15]。Prusa等人[16]提出了一种使用CNN编码文本的方法，这种方法适用于字母大小，允许从原始文本保留更多的信息，以提高分类性能。Mou等人[17]提出了一个基于树的CNN来捕获句子级语义。Pang等[18]将文本匹配作为图像识别任务，并使用多层CNN识别显著的n元模式。Wang等人[19]提出了一个基于CNN的模型，它结合了短文本的显式和隐式表示来进行分类。将CNN应用于生物医学文本分类[20]的兴趣也日益增长。

注意机制是识别文本中相关词的有效工具，现已成为深度学习模型的有效工具，注意力是由我们如何对一个图像的不同区域或一个句子中的相关词进行视觉注意所激发的。在为NLP[21]开发深度学习模型时，注意力成为一个日益流行的概念和有用的工具。杨等人[22]提出了一种用于文本分类的分层注意网络。沈等[23]提出了一种有向自注意网络，用于RNN/CNN自由语言理解。Wang等人[24]将文本分类视为一个标签-词匹配问题：每个标签与词向量嵌入在同一个空间。Kim等人[25]提出了一种语义句子匹配方法，该方法使用一个紧密连接的循环和共同注意网络。与DenseNet类似，该模型的每一层都使用连接的关注特征信息以及前面所有循环层的隐藏特征。该算法能够从最底层的嵌入字到最顶层的递归层保存原始特征信息和相关特征信息。殷等[26]提出了另一种基于注意力的句子对匹配CNN模型。使每个句子的表征考虑到其成对句子。这些相互依赖的句子对表征比孤立的句子表征更有效，在多种分类任务中得到验证。

预训练模型自2018年以来，我们看到了一系列大规模的基于变压器的预训练语言模型（PLM）的兴起。与早期基于CNNs或LSTMs[27]的上下文嵌入模型相比，基于Transformer的PLM使用了更深层的网络架构（例如48层的Transformers[28]），并且预先接受了大量文本语料库的训练，通过根据上下文预测受限的单词来学习上下文文本表示。这些PLM使用特定于任务的标签进行了微调，并在许多下游NLP任务（包括文本分类）中刷新了新纪录。虽然预训练是无监督的，但是微调是监督式学习的，如前所述，openAIGPT使用从左到右的转换器来学习自然语言生成中的文本表示，而BERT使用双向转换器来理解自然语言。统一语言模型（UniLM）[29]被设计用于处理自然语言理解和生成任务。Unilm使用三种类型的语言建模任务进行预训练：单向、双向和序列到序列的预测。据报道，UniLM[30]的第二个版本在广泛的自然语言理解和生成任务方面达到了新的最高水平，明显优于以前的PLM，包括ELMo[31]、GPT-2[32]、XLNet[33]、BERT及其变种。

图神经网络，用于捕获自然语言的内部图形结构，如句法和语义分析树，自然语言文本虽然表现出顺序，但也包含内部的图形结构，如句法和语义解析树，它们定义了句子中词之间的句法/语义关系。为NLP开发的最早的基于图的模型之一是TextRank[34]。在文献[35]中，彭等人提出了一种基于层次分类感知和注意力图胶囊CNNs的文本分类模型。姚等人[36]使用了类似的图CNN（GCNN）模型进行文本分类。他们基于词的共现和文档的词关系为语料库建立了一个单一的文本图，然后学习语料库的文本图卷积网络（TextGCN）。

混合模型，将注意力、RNNs、CNNs等结合起来，捕捉句子和文档的局部和全局特征，将许多混合模型已经开发结合了LSTM和CNN架构，以捕捉句子和文档的本地和全局特征。朱等[37]提出了一种卷积LSTM（C-LSTM）网络，Zhang等人[38]提出了一个用于文档建模的依赖敏感CNN（DSCNN）。Chen等人[39]通过CNN-RNN模型实现了多标签文本分类，该模型能够同时捕获全局和局部文本语义，从而建立高阶标签相关性模型，同时具有可控的计算复杂度。Tangetal。肖等人[40]将文档视为一个字符序列，而不是文字，并提议同时使用基于字符的卷积和循环层进行文档编码。在[41]中，Kowsari等人提出了一种用于文本分类的分层深度学习方法（HDLTex）。HdLTex采用了混合深度学习模型架构的堆栈，包括MLP、RNN和CNN，在文档层次的每个级别提供专门的理解。

# 2 关键技术

## 2.1 短文本向量化

在语音中，由音频频谱序列向量组成的矩阵被用作输入，并被馈送到神经网络进行处理；在图像中，由图像的像素形成的矩阵展平成向量形成的向量序列被馈送到神经网络进行处理。自然语言处理文本数据应该如何处理？我们也用向量来表达每个单词。预处理后的短文本需要向量化。词向量技术是将单词转化变成密集向量。在自然语言处理应用中，词向量作为机器学习和深度学习模型的特征被输入。受神经网络语言模型的启发，研究人员发现了一种表达单词向量的分布式方法，即通过上下文来表达单词的意思。此外，与基于稀疏表示生成词向量的统计方法如词袋模型和神经网络生成的分布式表示词向量相比，取得了更好的结果。

然而，这种方法也有一些缺点，即不能表达多义词，众所周知，自然语言处理难就难在它的歧义性，一个词在不同的语境中代表的意思可能截然不同，而且多义词的意义会受到训练语料的影响。例如，“我喜欢吃苹果"”和“我喜欢苹果公司”这两个词有不同的含义。然而，如果大量的训练语料代表了苹果，水果的语义，那么训练词向量中的“苹果”将包含水果的语义，这将导致对模型的误解。

为了解决这个问题，出现了一种基于语境化表示生成单词向量的语言模型。这种模型在对句子进行编码时，结合了每个单词所在句子的上下文。这个基于上下文的词向量成功地解决了区分多义词的问题。BERT就是这样一种生成基于上下文表示词向量语言模型。

## 用预训练模型来特征提取

预训练模型是一个已经训练好的保存下来的神经网络，该网络之前在一个大型的数据集上进行了训练（例如图像线管的网络在ImageNet数据上进行了训练）可以将预训练模型当做特征提取器来使用，用于迁移学习，当预训练模型学习到的特征容易泛化的时候，迁移学习才能得到比较有效的使用，迁移学习是指现在一个基础通用的规模大的数据集上进行任务的训练，生成一个基础网络，然后将学习到的特征重新进行调整或迁移到另一个目标网络上，用来训练目标任务的数据集，与训练模型用于特征提取关键在于将输出层去掉，保持原有权重，将剩下的整个网络当做一个特征提取器，从而应用到新的数据集中。从早期的预训练词向量（比如 Word2vec 和 GloVe等）到句子级表示（比如 ELMO 、 OpenAI GPT等）都对自然语言处理任务有很大的促进作用。

## 2.3 BERT模型

### 2.3.1 BERT模型结构

BERT（基于Transformer的双向编码器表示）是谷歌人工智能实验室的研究人员最近发表的一[篇论文](https://arxiv.org/pdf/1810.04805.pdf)。它在机器学习社区中引起了轰动，在各种 NLP 任务中呈现最先进的结果，包括问题回答 （SQuADv.）、自然语言推理（MNLI）等。

BERT 的关键技术创新是将"Transformer"的双向训练应用于语言建模。这与以前从左到右或从左到右和从右到左联合训练的文本序列的努力形成鲜明对比。论文结果表明，与单方向语言模型相比，双向训练的语言模型对语言语境和流程有更深刻的认识。在本文中，研究人员详细介绍了一种名为遮蔽语言模型（mask language model MLM）的新技术，该技术可以使我们的模型实现双向训练。

在计算机视觉领域，研究者们不断展示了迁移学习的价值——在已知任务（例如ImageNet）上预先训练神经网络模型，然后进行微调——使用之前经过预训练的神经网络模型作为我们新的特定用途模型的基础，继而进行进一步训练。近年来，研究人员一直在证明，类似的技术在许多自然语言处理任务中也是有用的

BERT 利用Transformer，一种了解文本中单词（或子词）之间的上下文关系的注意机制。Transformer架构中包括两个独立的机制 -——读取文本输入的编码器和生成任务预测的解码器。由于 BERT 的目标是生成语言模型，因此只需要编码器机制。谷歌在一[篇论文](https://arxiv.org/pdf/1706.03762.pdf)中描述了Transformer的详细工作。

与序列模型（按顺序读取文本输入，从左到右或从右到左）相反，Transformer编码器一次读取整个单词序列。因此，它被认为是双向的，尽管我认为说它是非方向的更为准确。此特征允许模型根据单词的所处环境（单词的左侧和右侧）学习单词的上下文。

下图显示了BERT模型结构。输入是一系列字符，这些字符首先嵌入到向量中，然后在神经网络中进行处理。输出是大小 H 的矢量序列，其中每个矢量对应于具有相同索引的输入字符。

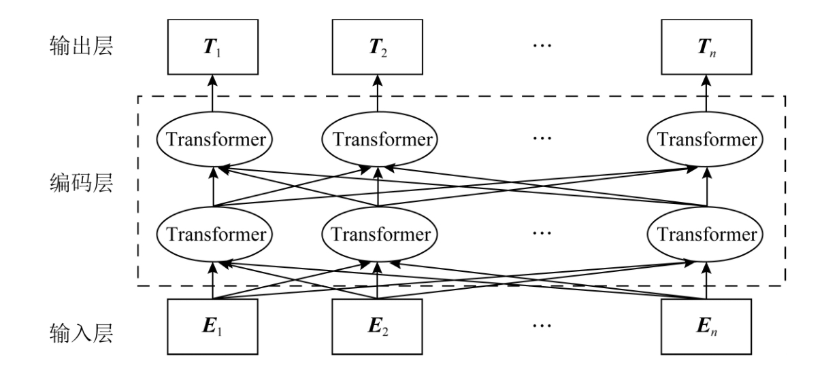


图2-2 BERT模型结构图

Fig.2-2 BERT model structure diagram

### 2.3.2 Transformer

Transformer结构如图2-3所示，编码器实现的是对原信息的理解 解码器实现的是对内部状态到目标信息的映射。每个编码器都有完全相同的内部结构，Transformer结构如图2-3所示。

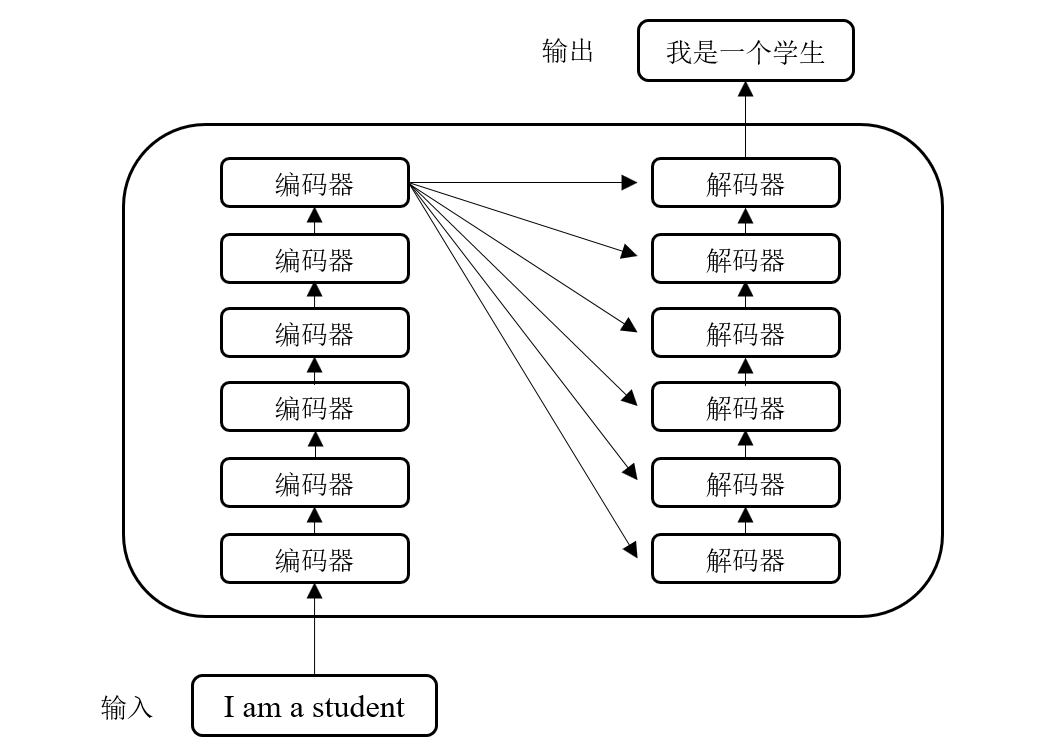


图2-3 Transformer结构图

Fig.2-3 Transformer structure diagram

编码器首先通过自注意力层，这可以帮助编码器在编码特定单词时查看输入句子中的其他单词。自注意力层的输出继续通过求和和归一化输入到前馈神经网络，然后通过求和和归一化传输到下一个编码器。

解码器同样有自注意力层与前馈神经网络层，不同的是，它们中间还有一个编码-解码注意力层，帮助解码器专注于输入句子的相关部分。

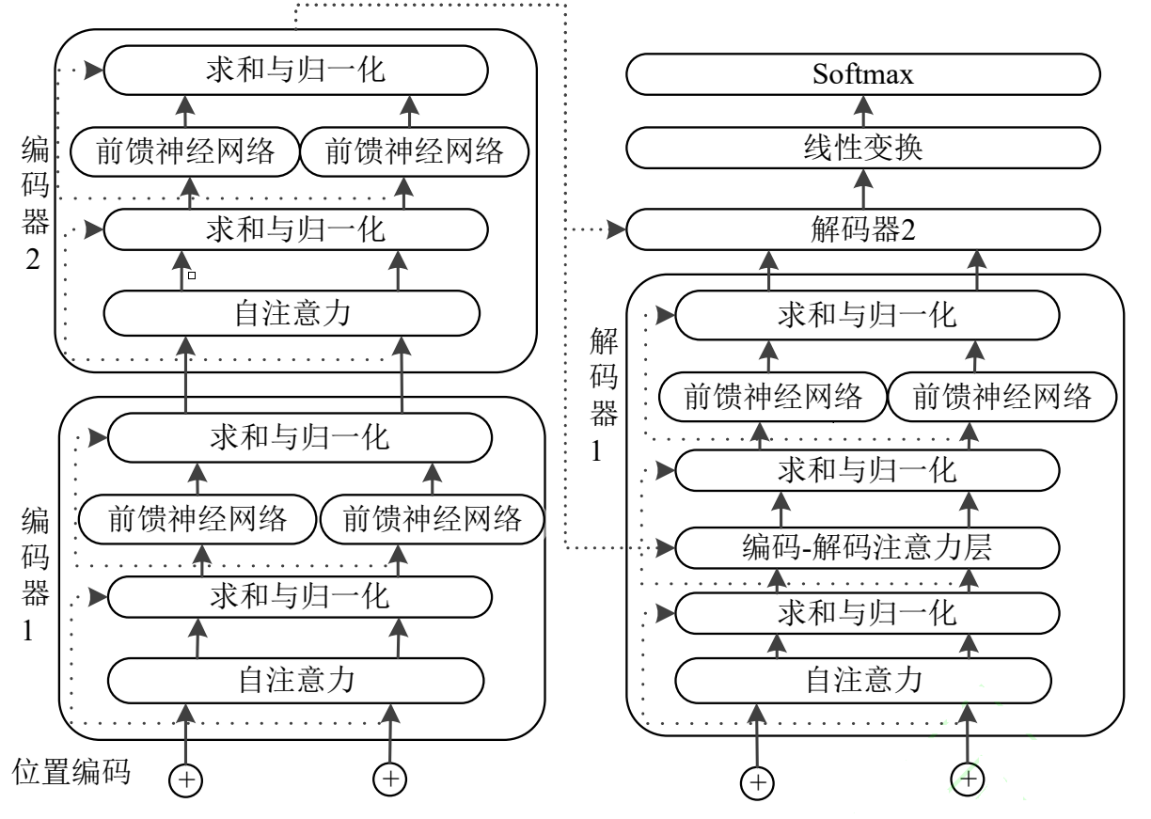


图2-4 transformer内部结构图

Fig.2-4 Internal structure diagram of transformer

运行流程：

输入“狗狗不再想要去海边它快要累垮了”，句子通过处理，输入的是一连串单一字符，问题在于，我们怎样才能让模型知道代词“它”指的是“狗”而不是“海”呢？

我们用公式来说明上述步骤，假设编码器的输入为：

（2-8）

解码器的输出为：

（2-9）

所有输入都要经历这样一个过程，即乘上三个不同的权重矩阵得到不同的向量，计做。

拿每个中的元素去对每个中的元素做attention，告诉词语间有多匹配，自注意力就是语句对自身计算注意力权重。

三个向量：查询向量 关键向量 值向量。

计算矩阵：

（2-10）

（2-11）

（2-12）

拿我们刚才那个例子，字符“它”经过词嵌入得到向量，将输入乘上三个权重矩阵产生三个向量，用去对做attention 得到一个分数，这个分数，就是注意力的值，为了平衡，我们将分数除以k向量维度的平方根，得到一个新的分数，然后通过Softmax操作传递结果，使分数规范化，即都为正数，加起来为1，再将结果乘以值向量，结合匹配度与语义信息，得到自注意力模型的输出 计算过程如下：

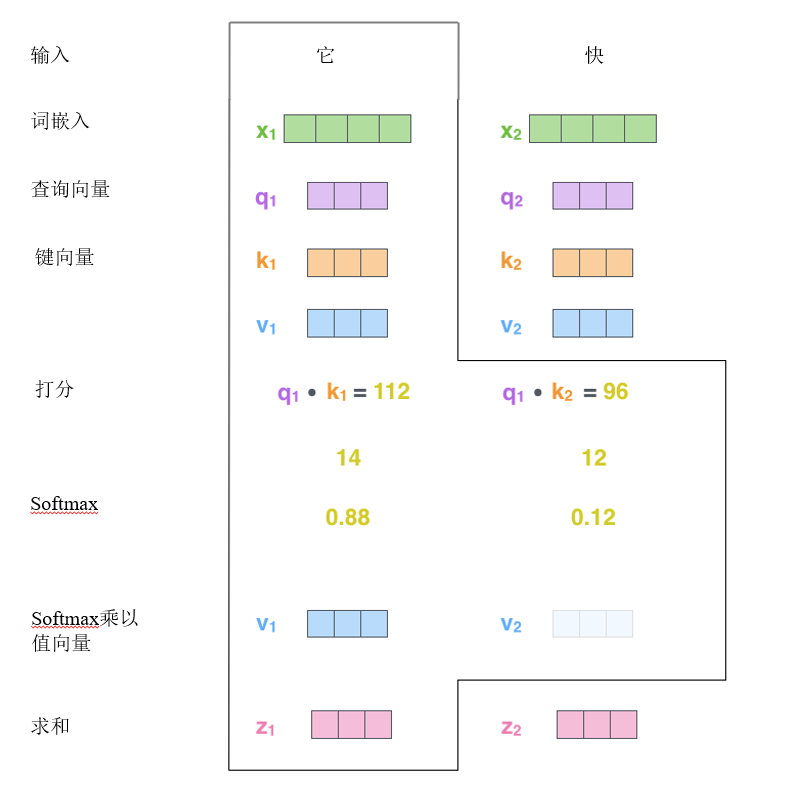


图2-5 计算过程图

Fig.2-5 Q, K, V calculation process diagram

上述过程总结为公式：

计算attention:

(2-13)

为键向量维度，我们示例句子中对“它”一字进行编码时，不同的注意力头关注对象不同，一个注意力集中在“狗”上，另一个集中在“累”上，从某种意义上说，模型对“它”的表示同时包含了“狗”的表示和“累”的一些表示：

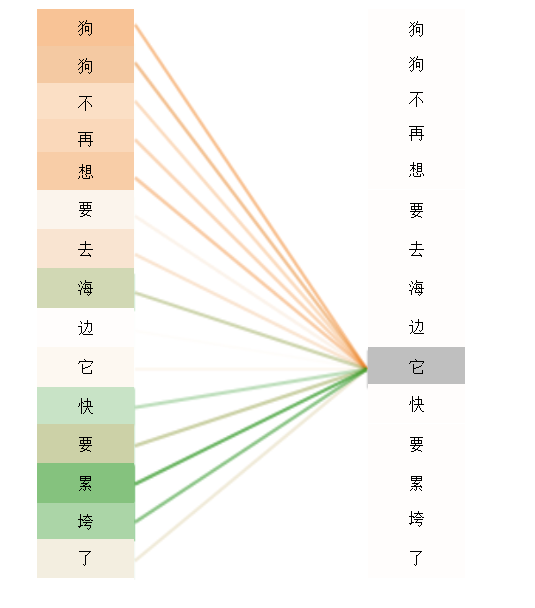


图2-6 注意力形象表示图

Fig.2-6 Attention image representation

BERT模型采用的多头注意力（Muti-head-attention)实际上就是多个self-attention的堆叠，如下图所示，多层叠加的self-attention组成了multi-head-attention，不过因为多层的缘故，最后所有的self-attention会生成多个大小相同的矩阵，处理方式是把这些矩阵拼接起来，然后通过乘上一个参数矩阵得到最后的计算结果。

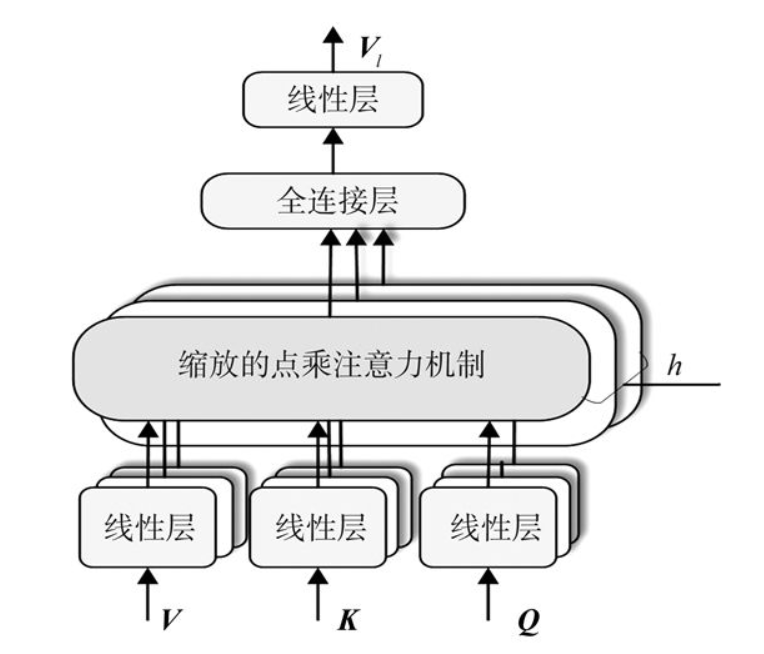


图2-7 bert模型内部图

Fig.2-7 Model internal diagram

Multi-Head Attention 通过多层的 Self-Attention 可以将输入语句映射到不同的子空间中，于是能够更好地理解到语句所包含的信息。

由公式6，对每个注意力头计算出相应的匹配程度向量，。

对于单字“它” 下图全面展示了计算流程：

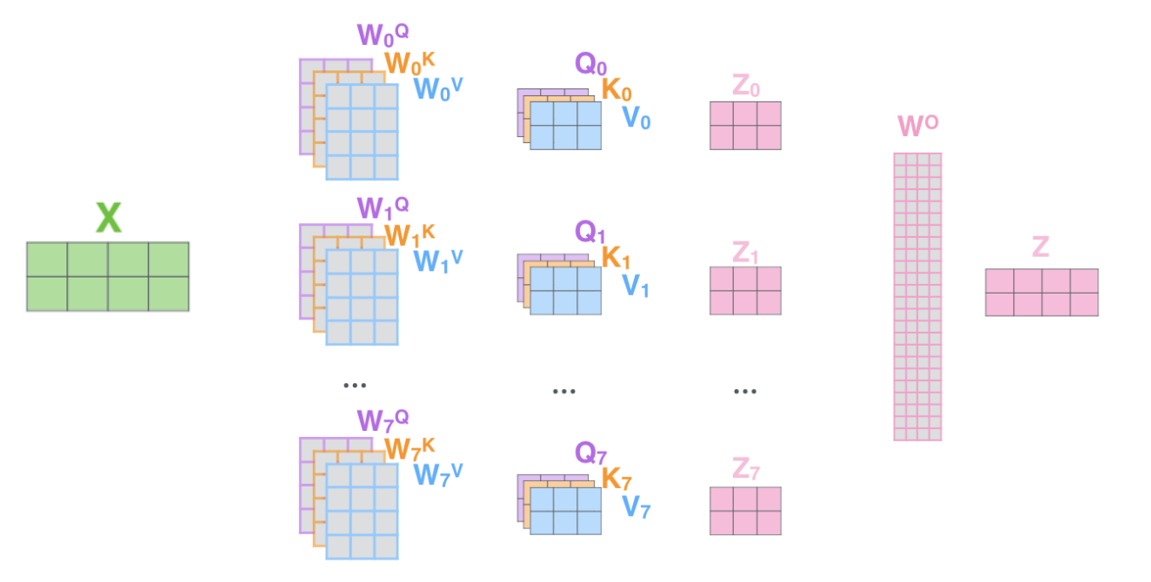


图2-8 详细计算过程图

Fig.2-8 Detailed calculation process diagram

该过程可用以下公式表示：

(2-14)

(2-15)

Tranformer作为一个特征抽取器为BERT服务时相较传统有了些许不同，多了残差连接，在multi-head attention之后是拼接和归一化，更新Z,该计算过程如下：

(2-16)

再将结果输入一个前馈神经网络层，使用relu这个激活函数，运算过程如下：

(2-17)

均为待训练权重。

# 系统的设计与实现

根据对中文短文本新闻分类系统的整个流程分析与设计，按照系统不同功能的划分，本文的系统功能结构图如下所示：

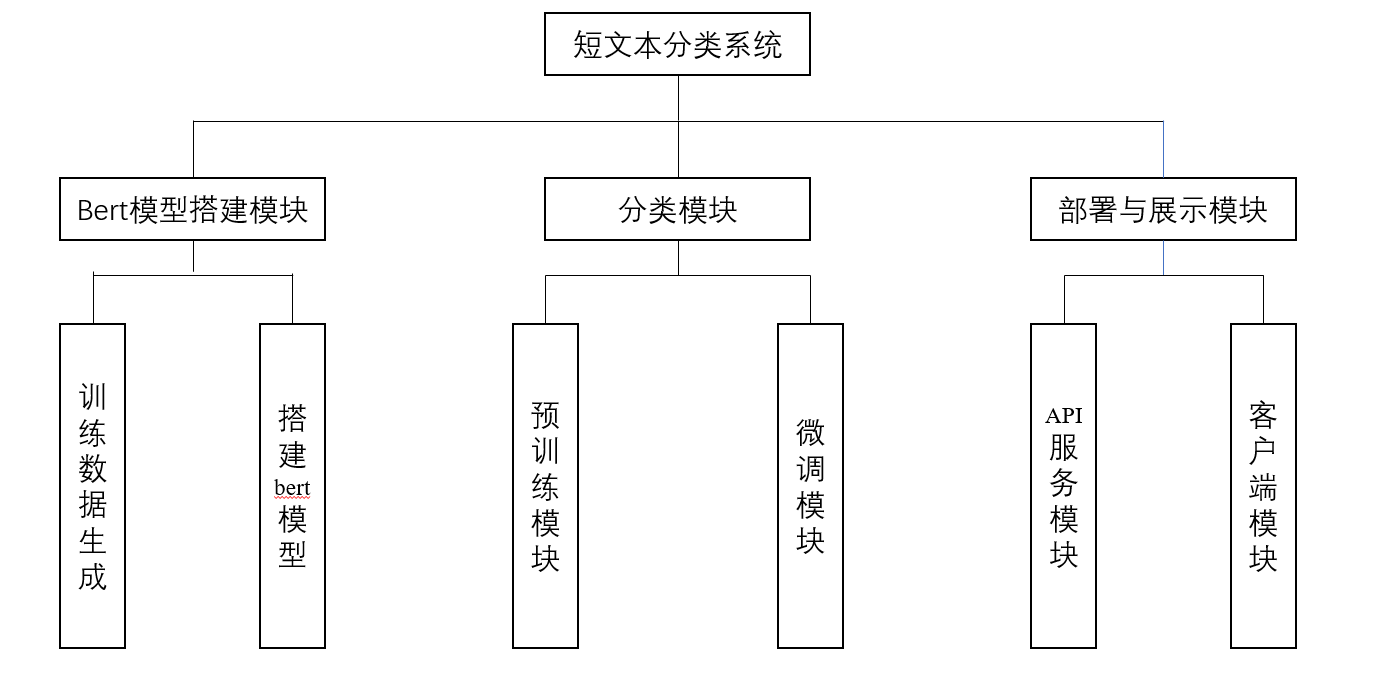


图3-1 系统功能结构图

Fig.3-1 System functional structure diagram

系统共分类为三大模块：BERT模型搭建模块、分类模块以及部署与展示模块。我们先把BERT整体模型框架搭建好，然后进行预训练和微调，最后完成模型的部署与展示。

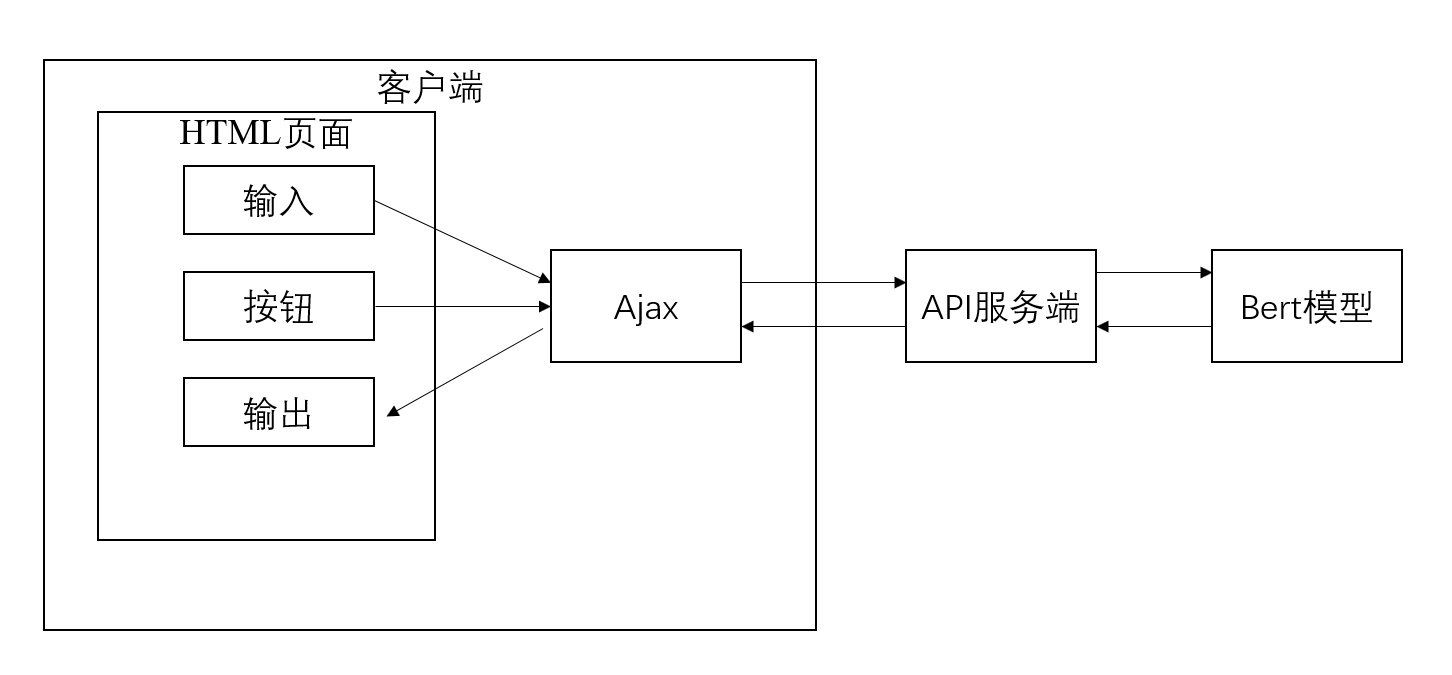


图3-2 部署与展示基本架构图

Fig.3-2 Deployment and display basic architecture diagram

最终系统展示图：

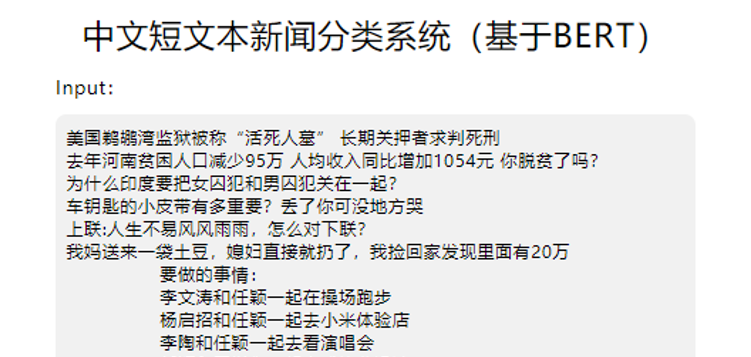
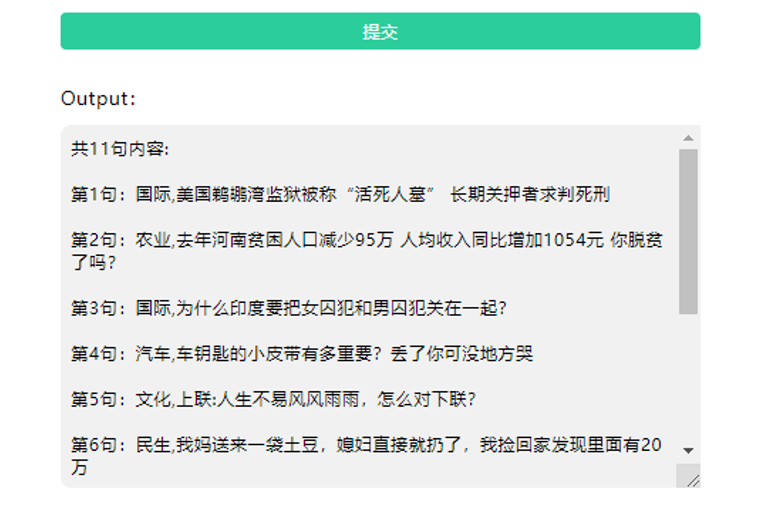
 

图3-20 分类结果图

Fig.3.3 Classification result graph

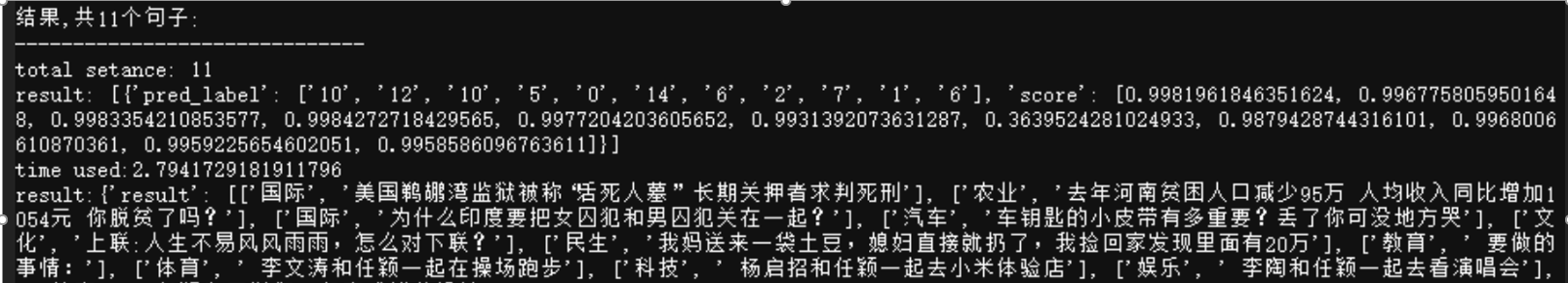


图3-21 后端显示图

Fig.3-4 Rear view display

# 4 实验分析与系统测试

## 4.1 实验相关准备与设置

### 4.1.1 实验环境

整个实验硬件环境如表4-1所示：

表4-1 硬件环境配置表

Tab.4-1 Hardware environment configuration table

|  |  |
| --- | --- |
| 名称 | 值 |
| CPU型号 | Intel(R) Xeon(R) CPU E5-2678 v3 @ 2.50GHz |
| GPU型号 | NVIDIA RTX2080Ti |
| 主板型号 | X10DRG-O+-CPU |
| 硬盘型号 | Samsung SSD 860 |

整个实验软件环境如表4-2所示：

表4-2 软件环境配置表

Tab.4-2 Software environment configuration table

|  |  |
| --- | --- |
| 名称 | 值 |
| 开发工具 | PyCharm 2020.1 x64 |
| 深度学习框架 | TensorFlow-gpu1.14 |
| 后端 | Flask |
| 前端 | HTML5 + CSS3 |

### 4.1.2 数据集

本文选用了开源的今日头条中文短文本新闻分类数据集[43]，它从今日头条中摘录了38w条中文短文本新闻组成，这些短文本新闻分为十五个类别，包括体育/财经/教育/旅游/国际/科技/房产/汽车/军事/文化/娱乐/证券/农业/电竞/民生，本文对所有类别的短文本新闻进行平均抽样，共抽取了20w条新闻短文本，具体分布如下：

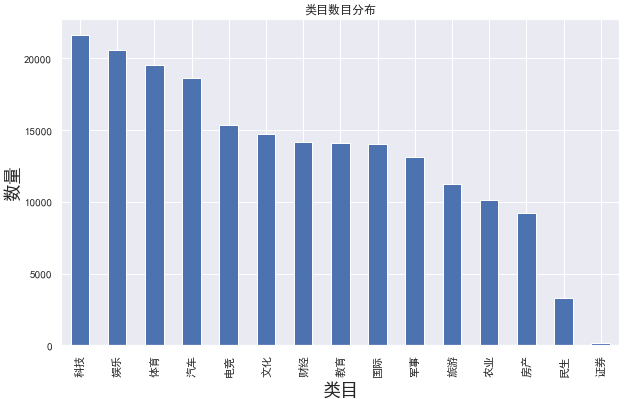


图4-1 数据集类目分布图

Fig.4-1 Data set category distribution map

其中16w新闻短文本作为训练集，2w作为测试集，2w作为校验集。

数据示例如下所示：

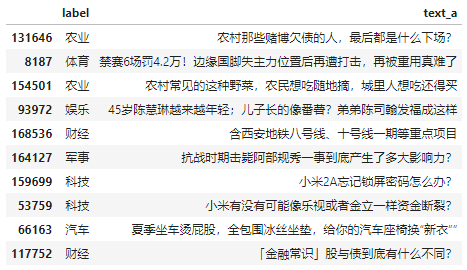


图4-2 数据集数据示例图

Fig.4-2 Data set data sample diagram

### 4.1.3 参数设置

本次实验涉及很多参数，我们采用了google发布的与训练好的中文模型“BERT-base，Chinese”该模型采用了12层的Transformer，隐藏尺寸为768，共有12个多头注意力，模型参数总大小为1.1亿（110m），优化器采用的是Adam，学习率设置为2e-5，句子最大长度设置为12.

表4-3 BERT模型训练参数表

Tab. 4-3 BERT model training parameter table

|  |  |
| --- | --- |
| 名称 | 值 |
| Transformer层数  隐藏尺寸数 | 12  768 |
| 注意力头数（multi-head-attention） | 12 |
| 模型参数总大小 | 110M |
| 优化器  学习率（learning\_rate）  句子最大长度（maax\_seq\_length）  批大小（train\_batch\_size）  迭代次数（num\_train\_epochs） | Adam  2e-5  128  32  5 |

# 5 结论

本文通过对BERT模型的分析与设计，循序渐进地完成了我们的短文本新闻分类系统，在这个过程中我复现了BERT模型并利用中文新闻数据集对它进行微调，最终把它迁移到我们的中文文本分类任务中，取得了很好的效果，证明了BERT模型的通用性。同时整个系统设计过程中我也收获良多，由远及近、由浅入深地研究了自然语言处理相关任务的方法，为以后研究打下了一定基础。

BERT屠榜后，一系列基于BERT的相关模型开始崭露头角，比如RoBERTa、ALBERT、ERNIE等等，这些BERT后续模型，已经在不断地为自然语言处理领域输送无限可能，让 nlp这颗明珠更加璀璨。我们有理由相信，接下来几年中，两阶段模型：超大规模预训练+具体任务FineTuning依然会是主流，这股浪潮势必会为更多领域带来新的突破。

随着计算力的增长，模型朝更大，更深方向发展，2019年10月，google提出了新的预训练模型T5，该模型有110亿个参数，在GLUE基准测试中取得当前最佳效果，2020年5月28日，恰逢本文写作期间，OPENAI又发布了GPT-3模型，一个参数高达1750亿，训练数据45TB的巨无霸单向语言模型，以当前的算力成本，训练一次预计花费数千万美元，如此巨大的工作量带来的好处也是显而易见的，经过这么多数据的训练，模型已掌握了很多人类语言相关知识，能够根据你的关键词写论文、编程以及做算数题等等，而且它主要聚焦于更通用的NLP模型，不需要微调便能很好的解决问题。

未来何去何从，强人工智能在我们有生之年能否诞生？我们拭目以待。