舆情文本多分类与敏感性检测技术研究与实现

文本多分类技术研究与实现

Introduction

随着大数据时代的到来，数据的体量越来越大，内容也鱼龙混杂。如何从海量的数据中筛选出有价值的可以满足需求的特定数据，并使其发挥出本身的价值，成为了各行各业，尤其是互联网企业正在努力解决的关键问题。在这样的背景下，文本分类成为了解决上述问题的一项重要技术。

文本分类是指计算机将载有信息的文本映射到预先给定的某一类别或某几类别主题的过程。有效高效的文本分类方法,可以正确归纳文本主题，总结文章内容，充分利用数据的价值，进而开发出有针对性夫的个性化用户推荐系统,使其可以根据用户的个人兴趣来定位并推荐相关的文本资料，从而实现商业需求。这样的优势使得文本分类任务愈发受到各行各业，尤其是掌握大量数据的新闻媒体，电商等行业的青睐。

我们可以根据实际应用场景，将文本分类这一课题继续细分。根据类别的数目，可以分为二分类和多类分类；根据每条文本所属的类别个数，可分为单标签文本分类和多标签文本分类。本文主要研究的是基于BERT模型的单标签和多标签的文本分类。

最初的文本分类仅仅通过分析本文内容，抓取文本关键词，通过堆和树等数据结构进行简单的分类操作。比如基于规则的特征匹配，例如篮球映射到体育，电影映射到娱乐等。这样的方法虽然容易理解，但是存在诸多问题。首先，这个方法依赖专家系统，不同类别需要专门构建特征规则，费时费力。其次，运行效率低且准确率低，不能满足实际应用中对于文本分类准确率的要求。

近年来，随着人工智能的蓬勃兴起，文本分类这个课题越来越受到学者，从业人员和企业的关注。最初的解决方案是机器学习方法，如朴素贝叶斯分类，决策树，支持向量机（SVM）等。基本原理是首先，专家们利用领域专业知识，创建合适的输入特征，然后把训练数据输入模型，通过算法拟合出最优的，即错误最小化的模型作为输出，最后把文本作为输入，寻找文本中的某些特征，根据这些特征通过训练好的模型将文本分为不同的类别。这样的机器学习方法，在准确率上远高于人工分类的方法，但是仍有不足之处。首先，如何创建合适的输入特征，即特征工程，是机器学习方法的最大瓶颈。特征工程这一步骤需要大量的专业知识以及不断地尝试，取得突破极为困难。其次，受限于计算能力，训练时使用的数据量有限，导致模型结构比较简单，存在欠拟合的问题。假如使用更大量的数据，又会出现过拟合的问题。

为了解决以上的问题，计算机科学家使用更多层的神经网络，使神经网络自动通过每一层产生适当的特征，从而避免了特征工程的复杂，创造出许多深度学习方法如卷积神经网络（CNN），递归神经网络（RNN），长短期记忆模型（LSTM）等。这些方法主要利用了文本，图像，语音等数据的连续，稠密的特点，利用其局部相关性，自动获取文本的特征。本文采用的BERT（Bidirectional Encoder Representation from Transformers）模型，就是基于深度学习方法，利用多层的双向transformer的网络结构实现了自动获取文本特征的能力。

BERT框架把原来的机器学习过程分成两个阶段：预训练（pre-training）和微调（fine-tuning）。首先，利用Google拥有的强大数据计算能力和海量的训练数据，训练出一个预训练模型，然后使用者可以根据自身的需求对预训练模型进行微调。这样既保证了模型具有极高的准确率，又对于使用者的计算能力没有太高的要求，从而在自然语言处理的许多任务中表现出色。而这种提供一个供其它任务迁移学习的模型，根据任务微调或者固定之后作为特征提取器的二阶段训练模式也成为深度学习的潮流。

本文就是使用预训练+微调的模式，把BERT模型应用在文本分类任务中，在实现BERT模型具有的高准确率的基础上，把模型运用到实际的应用中，使模型发挥自身的价值，并结合第二部分介绍的敏感词检测功能，实现一个完整的高效的业务流程。

Related work

自2018年BERT模型提出以来，BERT模型被广泛应用在不同的自然语言处理任务中，均取得了出色的成绩。但是学者没有停止对BERT的研究，提出了许多针对不同自然语言处理任务的改进。其中比较出色的包括MT-DNN，XLNet，RoBERTa和SpanBERT。

MT-DNN（Multi-Task Deep Neural Networks for Natural Language Understanding）模型是一种多任务深度神经网络，用于跨多种自然语言理解任务的学习表示。多任务学习( MTL )的灵感来自于人的学习活动。在人类的学习活动中，人们经常应用从以前的任务中学到的知识来帮助学习新的任务。多任务学习的优点有两个，一是弥补了有些任务的数据不足问题；二是得益于正则化效果，这种效果使得学习到的表示具有更强的泛化能力，防止模型过拟合。

MT-DNN模型的思想是，多任务训练和预训练语言模型是互补的技术，可以结合起来提高文本特征表示的学习效果，进而提高各种自然语言处理任务性能的表现。因此，MT-DNN模型的训练也分为两个阶段，即预训练和微调。但是，MT-DNN在微调阶段用多个特定于任务的层进行多任务学习。主要有四种类型，其具体任务全部出自GLUE（General Language Understanding Evaluation，通用语言理解评估）：（一）单句分类：CoLA（Corpus of Linguistic Acceptability）任务是预测英语句子是否合乎语法，SST-2（Stanford Sentiment Treebank）任务预测电影评论是正向还是负向。（二）文本相似度：这是一个回归任务。对于给定的句子对，模型计算二者之间的相似度。在GLUE中即为STS-B（Semantic Textual Similarity Bench-mark）这个任务。（三）成对文本分类（文本蕴含）：对于给定的句子对，推理两个句子之间的关系。RTE（Recognizing Textual Entailment）和MNLI（Multi-Genre Natural Language Inference）是语言推理任务，推理句子之间是否存在蕴含关系、矛盾的关系或者中立关系。QQP（Quora Question Pairs）和MRPC（Microsoft Research Paraphrase Corpus）是预测句子是否语义等价。（四）相关性排序：给定一个问题和一系列候选答案，模型根据问题对所有候选答案进行排序。QNLI（Stanford Question Answering）任务是预测候选答案中是否包含对问题的正确答案。这个任务重排了候选答案，将正确答案排在更前。

MT-DNN的体系结构主要可以概括为：底层（即BERT的预训练阶段）在所有任务中共享，顶层（即上述四种类型的任务）代表特定任务的输出。输入是一个句子或一对组合在一起的句子对。训练模型时把所有数据合并在一起，每个batch只有单一任务的数据，同时会带有一个task-type的标志，这样模型就可以分辨不同的任务，进行不同的loss计算。相比于交替训练（先训练任务A再训练任务B），这个训练方法可以有效避免偏向某个任务，近似地优化所有多任务的和。

从训练结果上来看，MT-DNN模型在GLUE中除了WNLI之外的全部任务都超越了已有的模型，尤其是在数据集较少的时候，MT-DNN相较于BERT有着极大的提升。这样的结果可以证明，预训练模型+多任务学习的方式，确实可以提升了模型的表现，而且可以有效解决数据集不足的问题，使得模型具有更好的泛化能力。但是MT-DNN模型也有有待提高的地方，比如作者提到的更有效的训练方法，更深度的权重共享等。

XLNet也是学者对BERT模型的一种改进。XLNet首先提出了自回归语言模型（Autoregressive LM）以及自编码语言模型（Autoencoder LM）的分析策略。自回归语言模型就是指根据上文内容预测下一个可能跟随的单词，就是常说的自左向右（或从右向左）的语言模型任务训练出的语言模型，自回归语言模型的优势是擅长生成式自然语言处理任务。 因为在生成上下文时，通常是单向的，自回归语言模型很自然地适用于此类 NLP 任务。但是，它只能使用前向上下文或后向上下文，这意味着它不能同时使用前向和后向上下文。这样的缺点使得预测的准确率仍有提高的空间。而自编码语言模型可以同时关注上下文，因为自编码语言模型的目的是从损坏的输入重建原始数据，方法是在输入中随机用伪码标记代替一部分单词，然后预根据上下文单词来预测这些被替换掉的单词，从而得到原句。BERT模型就是一个很典型的自编码语言模型，因为BERT模型预训练时的任务之一就是Masked Language Model。这样的训练方法可以兼顾前文和后文，能比较自然地融入双向语言模型，相较于自回归语言模型是一个进步。但是自编码语言模型也有缺点。此训练方法在输入侧引入伪码标记，但是Fine-tuning阶段是看不到伪码标记的，导致预训练阶段和微调阶段不一致的问题，从而引入了一些人为误差。而且该方法假设被伪码代替的词之间彼此独立。而在实际数据中，被代替的词之间可能会存在相关性，我们想要语言模型学习这种相关性来对这些被代替的词进行更准确的预测。

XLNet试图找到一种方法，融合自回归和自编码两种语言模型，集合两者的优点，避免两者的缺陷。XLNet采用的是自回归语言模型，为了解决上述的缺点，作者提出了排列语言模型（Permutation Language Modeling）。传统的自回归模型是把序列的值按顺序进行建模，所以是单向的。XLNet采用的方法是最大化所有可能的序列的因式分解顺序的期望对数似然，即随机排列组合句子中的单词，在随机排列组合后的各种可能里，再选择一部分作为模型预训练的输入。假如我们有一个序列abcd，如果我们要预测字母c，先对该序列进行因式分解，最终会有24种排列方式，假如抽取cabd的排列方式，因为c的左边没有其他的值，所以该情况无需做对应的计算；假如抽取adcb的排列方式，模型就会保留a，d的信息，从而使得上下文信息都能被保留，解决了传统自回归语言模型的缺点。而且这种方法避免了采用伪码标记，也解决了BERT模型的缺点。

但是这种方法在实现的过程中也有困难。虽然排列语言模型能满足目前的目标，但是对于普通的transformer结构来说存在问题。传统的Transformer结构中，某个单词的内容和位置向量在输入到模型前就已经加在一起了，后续的隐向量同时具有内容和位置的信息。但是，我们希望在预测下一个词时只能提供位置信息，不能提供内容相关的信息。因此模型需要同时做两件事，首先它需要预测自己到底是哪个字符，其次还要能预测后面的字符是哪个。为了解决这个问题，论文中提出来新的机制，来实现目标位置感知——双流自注意力机制（Two-Stream Self-Attention）。该机制包括两部分，一个是内容流自注意力，就是标准的Transformer的计算过程，同时编码了上下文和自身的内容和位置信息；二是查询流自注意力，包含上下文的内容信息和目标的位置信息，但是不包括目标的内容信息。两个流共同计算，进行模型的训练。

同时，XLNet模型还集成了transformer-xl的两个最重要的技术点，即片段循环机制和相对位置编码。片段循环机制主要是为了解决超长序列的依赖问题，首先取第一个段进行计算，然后把得到的结果的隐藏层的值进行缓存，第二个段计算的过程中，把缓存的值拼接起来再进行计算。该机制不但能保留长依赖关系还能加快训练，因为每一个前置片段不需要再重新计算。但是BERT模型中采用的绝对位置编码，没法区分之前存储的信息到底是哪一个片段里的，这就导致了一些位置信息的损失。因此XLNet采用相对位置编码，在计算attention的时候根据当前的位置和要参考的位置的相对距离来”实时”体现在attention的计算中。

在实验中，XLNet的表现也相较BERT有明显提升，尤其是在较长的文本上的表现。排列语言模型的机制对于自回归和自编码两种模式的融合，片段循环机制的引入，更大的数据规模，这些因素共同提升了XLNet在自然语言处理领域的表现。

RoBERTa（a Robustly Optimized BERT Pretraining Approach）是BERT 的改进版。与BERT使用相同的网络结构，但是从以下四个方面对BERT进行了提升：

首先是更大的数据规模，更大的batch size，更长的训练时间。RoBERTa 在训练过程中使用了更大的bacth size。尝试过从 256 到 8000 不等的bacth size。训练数据包括160GB 纯文本，相较于最初的BERT使用16GB 的数据集和英语维基百科进行训练，RoBERTa显然会具有更好的性能。

第二，RoBERTa使用动态掩码（dynamic masking）的方式进行预训练。原始Bert模型对每一个输入序列随机选择15%的单词替换成掩码，为了消除与下游任务的不匹配，还对这15%的单词进行分类（1）80%替换成掩码；（2）10%不变；（3）10%替换成其他词。但整个预训练过程，这15%的单词一旦被选择就不再改变，每个epoch都是重复，后续的每个训练步都采用相同的掩码。这就叫做静态掩码。作者为了研究掩码对于模型性能的影响，设计了两种改进方式进行了对照。第一种方式是在预处理的时候将数据集用不同的掩码拷贝 10 次，等价于原始的数据集采用10种静态 mask 来训练。第二种方式是不在预处理的时候执行掩码标记，而是在每次向模型提供输入时动态生成掩码。经过实验发现，第一种方法与原始BERT的静态掩码方式效果相同，而第二种动态掩码的方式可以提升模型的性能。所以在RoBERTa中，作者采用动态掩码的方式进行预训练。

第三，RoBERTa研究了原始BERT预训练过程中，NSP（Next Sentence Prediction，下一句预测）任务的效果。NSP任务的目的是原为了捕捉句子之间的关系，方法是输入连续的多个句子组成的文本A和B，判断这两组句子是否是连续的。在训练集的数据中，50%的B是A的下一组句子，50%的B是随机抽取的。近来有学者对于NSP任务提出质疑，认为这个任务不是必要的。RoBERTa的作者对此设计了不同的训练方法进行了实验：（1）原始BERT的NSP任务（2）用单个句子代替原来的多个句子，且句子长度不超过512。 由于这些输入明显少于512 个单词，因此增加batch size的大小，以使单词总数保持与原始BERT的NSP任务相似。（3）舍弃NSP任务，直接用来自同一个文档或者不同文档的连续多个句子作为输入，单词总数不超过 512 。一次输入可能会跨越文档边界，如果跨文档，则在上一个文档末尾添加文档边界标记。（4）舍弃NSP任务，直接用来自同一个文档的连续多个句子作为输入，单词总数不超过 512 。由于在文档末尾附近采样的输入长度短于 512个单词，因此在此情况下动态增加batch size大小以达到与方法（3）相似的单词总数。经过实验，方法（4）的性能表现最好。但是由于方法（4）需要动态调整batch size，为了避免不必要的麻烦。RoBERTa的作者采用了实验结果最接近方法(4)的方法（3）。这两种方法的表现都明显优于原始BERT的方法。

第四，RoBERTa采用了byte-level的字节对编码（BPE）的方式处理自然语言语料库中常见的大量词汇。字节对编码不依赖于完整的单词，而是依赖于子词(sub-word)单元，这些子词单元是通过对训练语料库进行统计分析而提取的。原始BERT使用的是 character-level 的大小为30K的字节对编码，而RoBERTa使用 bytes 而不是 unicode 字符作为子词的基本单位，可以编码任何输入文本而不会引入未知标记。词表大小也从30K提升到50K。这样的改进为原始BERT模型的预训练增加了1500万以上的参数，有效的提高了模型的表现。

RoBERTa设计实验评估了许多设计训练时的策略，并通过更高级的测试环境实现了对原始BERT模型的改进，而且在测试集上的表现甚至超越了XLNet，证明了设计训练策略的重要性。

SpanBERT也是针对BERT在预训练阶段的训练策略做出的改进。与RoBERTa相似的是，SpanBERT在训练时也舍弃了NSP任务，SpanBERT的作者给出了如下的解释：（1）相比起两句拼接，一句长句，模型可以获得更长上下文（类似 XLNet 的一部分效果）（2）在 NSP 的负例情况下，基于另一个文档的句子来预测词，会给 Masked Language Modeling 任务带来很大噪音。

SpanBERT的创新之处在于：

第一，提出了更好的掩码方案。对于原始BERT模型，训练时会随机选取整句中的最小输入单元来进行遮盖。这样会让本来应该有强相关的一些连在一起的字词，在训练时是割裂开来的。这样会导致训练结果的表现下降。为了解决这个问题，有学者尝试将这样具有较强相关性的单词连接起来一起用掩码标记（BERT WWM 模型）。还有学者更进一步，直接将由几个词组成的实体都遮盖掉（如ERNIE模型）。SpanBERT的做法是，根据几何分布，先随机选择一段（span）的长度，之后再根据均匀分布随机选择这一段的起始位置，最后按照长度遮盖。经过作者的实验，这种random span的方法是优于以上两种改进方法的。

第二，SpanBERT在训练阶段新加入了一项任务：Span Boundary Objective。在训练时取 Span 前后边界但不在Span内的两个词，然后用这两个词向量加上 Span 中被遮盖掉词的位置向量，来预测原词。这项新任务的加入，大大提高了random span的掩码方法的表现。

在以上创新点的基础上，作者主要对抽取式问答，指代消解，关系抽取，还有GLUE 做了实验。发现SpanBERT的表现普遍强于 BERT，而且SpanBERT 尤其在抽取式问答上表现好，这与它新加入的SBO预训练目标关系较大。

除了以上提到的已有模型，还有学者对于BERT模型在不同任务下的表现进行了研究。比较有代表性的如：（1）BERT模型在Argument Reasoning Comprehension 任务中的表现。ARC 任务是在给定的前提下，对于某个陈述，相反的两个依据哪个能支持前面提到陈述的推理。作者进行了实验，得出了结论并提出了合理疑问，认为BERT模型并没有真正学习到做出推测所需要的语义信息，只是利用了某些出现频率很高而且对于推测有很大影响的词，比如not，are等。作者又设计实验，创建了对抗数据集（Adversarial Dataset），重新进行实验，发现BERT模型在ARC任务上的表现并不好。（2）BERT模型在 Natural Language Inference（自然语言推理，NLI） 任务中的表现。作者首先假设在 NLI 中表现好的模型可能利用了三种启发式的特征，在任务的前提中就给了模型一些提示，有如下三种：语义重叠，对应的 推理是前提的子序列，对应的推理是前提的子串。基于这个假设，作者也做了实验并观察到，MNLI 训练集中许多数据点都存在这样的特征，且对应的选项是正确的数量远多于不正确。针对这种情况，作者构造了 HANS 数据集，均衡两种类型样本的分布，并且标记了每条数据的前提是否涉及上述几种特征。实验时模型在 MNLI 数据集微调，在 HANS 数据集评测，结果推理正确的数据点中模型都表现不错，而在推理错误类型中模型表现欠佳。这一实验结果支持了作者的假设：模型过度利用了这些启发式特征信息。因此作者在原来的训练集中加入了一定的 HANS 数据，构造了 MNL + 数据集，让模型在该数据集微调，最终获得了比原来更优秀的结果。

虽然BERT在许多多面仍有提高的空间，但BERT作为这些新模型的基础，仍具有可观的研究价值和提升潜力。接下来我们将通过BERT方法，实现在真实平台上可用，有效，高效的文本多分类模型，并研究BERT模型在文本分类任务中的表现。

BERT

本文研究的是基于原始BERT模型的文本分类任务。BERT的全称是Bidirectional Encoder Representation from Transformers，即双向Transformer的Encoder。

Attention

要介绍Transformer结构，首先要介绍注意力Attention这个机制。注意力的灵感来自于人处理信息时采取的一种手段，具体表现为我们如何对信息的不同区域关注度会有所区别，通常对于感兴趣或有更大价值的部分往往会分配大量的注意力。在深度学习中，注意力可以大致理解为对于某一个向量关注的程度如何，我们使用注意力向量来估计关注的部分和其他元素之间的关系强弱，并将不同部分的值的和用注意力向量加权得到的结果作为目标的近似值。

在自然语言理解领域内，注意力这一机制大放异彩。最初的自然语言理解任务中，解决方案大多为Sequence To Sequence模型。目的是将一个输入序列转换为一个新的目标序列，并且输入序列和目标序列的长度可以是不固定的。为了实现这一目的，模型主要做两件事情：（1）将输入序列数据的信息压缩到一个固定长度的上下文向量（context vector）中，得到的表示向量可以较好的包含整个输入的信息。这个阶段也叫做Encoder。

（2）使用上下文向量对模型进行初始化，然后输出转换后的向量。这个阶段也叫做Decoder。

这种传统的模型可以完成机器翻译、生成问答对话、句法分析等自然语言处理任务，但这个机制有一个关键的不足：使用固定距离的上下文向量，会导致其不能记住较长的句子。一旦完成了对于某个输入的处理，它就会忘记之前已经学到的部分。

Transformer

敏感词检测技术研究与实现

Introduction & Related Work

随着信息时代的发展，涉及政治，色情，暴力等因素的敏感词汇也层出不穷。在许多对外公共场合下，有些内容是要经过审查才能显示的。理论上讲，只要涉及用户输入的地方，都需要进行文本校验，以保证敏感词汇不被展示在网络上。所以我们需要一个有效高效的敏感词检测手段，在涉及文本输入的时候进行快速的检测。既不影响用户的使用体验，又能准确地过滤非法信息。在网络审查初期，都是通过人工审核，这种审核方式虽然准确，但与网络上文本产生的速度相比，其效率就显得过于低了。因此，自动化的敏感词检测方法的需求越来越强烈。

最普通的敏感词检测，就是用遍历的方法，从敏感词库中逐条读取敏感词，在待检测文本中进行搜索匹配。实现起来非常容易，但是问题很大：当待检测文本比较长，敏感词汇比较多的时候，这种方法效率很低，严重影响用户体验。

现在被广泛应用的敏感词检测方法有两种。第一种是本文采用的敏感词检测方法：DFA，即确定有限状态机，将字符比较转化为了状态转移。第二种方法是Aho-Corasick自动机算法。该算法是DFA的一种实现。这两个算法有两个特点，一个是扫描文本时完全不需要回溯，一次遍历全部检测；另一个是时间复杂度为O(n)，时间复杂度与关键字的数目和长度无关，因此具有很高的效率，不影响用户的使用体验。

DFA

DFA(Deterministic Finite Automaton)，即确定有限状态机，基本工作原理是通过事件和当前的状态得到下一个状态。在敏感词检测的任务中，我们可以把字符看做状态，把字符间的前后联系视作事件，这样就把字符的比较转化为状态的转移。具体的算法原理为：

首先以敏感词库中的词汇建立一个确定性的树形有限状态机。比如，假设敏感词库里有abc，abd，ef三个敏感词，我们就可以建立a->b->c , a->b->d ,e->f的树形结构。然后以待检测文本作为该有限状态机的输入，使状态机进行状态的转换，当到达某些特定的状态时，说明发生文本匹配，即在待检测文本中检测到了敏感词汇。假设待检测文本为abcdefg，首先检查DFA中是否有a这样一个起始状态，若存在，则检查第二个字符b是否是a的下一个状态，若是，再检查下一个字符c是否是b的下一个状态，结果为是。由于c已经是此确定有限状态机这条分支的最后一个状态，所以我们可以判定待检测文本中存在敏感词abc。检测完毕。

具体代码实现

敏感词检测功能的代码分为三个部分：

add()和parse()函数，负责建立状态机，以多维数组的形式储存敏感词库中的敏感词。

Filter()函数，负责根据待检测文本，以循环的方式查询已建立好的多维数组，得出结论是否匹配。假如匹配，则用长度与敏感词长度相同的连续占位符代替原有的敏感词，并把处理好的文本返回。

主函数main()，负责处理输入输出，调用parse()和filter()，并输出整个敏感词检测的流程所用的时间。

实验结果

不足之处