Improving Text Classifier Performance through Human-in-the-Loop: Enhancing Learning from Explanations

Abstract

文本分类利用自然语言处理(NLP)技术对预训练的文本进行分析，并为其分配相应的标签。特别是在洪水、地震等危机情境中，文本分类器对识别关键信息并将其有效地转发给相关机构，例如从社交媒体获取的信息，具有重要价值。然而，文本分类的效果大多依赖于丰富的训练数据，这种数据在许多情境中可能难以获取[1]。依赖大量非代表性标注数据会导致项目启动的延迟，并可能对模型的准确性造成影响。特别是，在紧急情境下，识别行动相关的信息（如伤亡或失踪）尤为困难。

为了解决这些问题，本论文提出人在循环（HITL）系统集成到分类器的训练中。结合自然语言解释的表示工程技术——ExpBERT，使用了经过MultiNLI自然语言推断数据集微调的BERT来从解释中学习[2]。这种优化后的嵌入表达被用作高神经网络分类器的输入，进一步提高性能。论文的核心任务在于通过结合少量但代表性的样本结合标签以及由人类或OpenAI提供的解释，来提高基于ExpBERT的文本分类器的精度[3]。

本论文深入探讨了主动学习在基于ExpBERT的文本分类任务中的应用。在迭代过程中，采用基于不确定性和多样性等acquisition functions，选择代表性的未标注实例供注释者处理。原始的小样本数据和新标注的数据一同被用于模型的再训练。结果表明，此方法在少数迭代中即可达到与使用大量训练数据的模型相似的性能。为了验证这一点，将在CrisisNLP dataset上对使用不同主动学习策略的模型性能进行了比较，同时也评估了结合Monte Carlo Dropout(MCD)的主动学习系统的效果。

本文的主要结论是：

人在循环中框架结合文本分类器能够使模型在只使用少量的标注数据和经过少量的迭代后，达到或甚至超过使用全部训练数据的模型性能。

与仅标注标签、低质量解释或少量解释相比，标注一定数量的高质量解释能够显著提高模型的性能。

在数据抽样方法上，基于语义的多样性抽样和集合MCD的Bayesian Active Learning by Disagreement(BALD)的主动学习策略可以在模型训练后，让模型能够达到较高的平均性能。

1 Introduction

本章首先介绍了基于ExpBERT文本分类器结合人在循环中的研究背景（1.1节），根据背景在1.2节产成了实验的动机，强调了此次研究的重要性。除此以外，将对工作进行概括并对比传统方式的局限。最后简要概括了主要目标以及挑战（1.3节）。

* 1. 背景

针对社交平台的紧急时间响应系统的重点主要集中在创建更好的文本分类算法来从少量数据中学习。然而，获得有用的注释数据集可能被证明是困难的[4]。针对文本分类器的预训练模型ExpBERT不仅接收分类标签，还接收对应的解释。解释包括哪些关键词导致了这种分类。ExpBERT使用这些信息融入到模型的训练中，使模型可以从更深层次的语义信息中学习来提高模型的泛化能力。因此，使用具有代表性的数据并了解关键词句后进行注释才可以引导模型精确分类。

但是，一般的社交平台数据的吞吐量无法完成对上万条数据注释。因此，许多能够低成本实现高信息价值数据抽样的主动学习形式在分类项目中得到了广泛的应用[5,6]。主动学习是解决这些问题的一种有效方法，它选择了少量具有高信息量的未注释样本，这些样本供在循环中的人类专家来注释[7,8]。得到有效注释后的新数据决定了下次迭代模型的性能，因此，使用不同的acquisition function查询信息最丰富的新实例可能是主动学习中最流行的方法。所以，查询策略自然成为了主动学习领域的研究重心。

同时，神经网络文本分类器往往不适应早期的不确定性抽样[9, 10]。因为，神经网络权重参数是固定的数值，导致了模型对正确以及错误数据的预测过于自信。然而，通过使用 dropout，可以在一定程度上引入模型的不确定性。在训练过程中，dropout 层会随机 "关闭" 一部分输入单元，这种随机性使得网络权重不再是确定的数值，从而模仿了权重的不确定性。

* 1. 动机

在上述背景下，本节研究重点关注从准确性，代表性以及健全性这三个方面阐述实验动机。

* + 1. 准确性

由推特所代表的社交网络的统一特征是庞大的数据量结合互联网语法。最终目标是在捕捉紧急需求时，需要对紧急消息类别进行精准分类同时派遣相关部门来解决需求。然而，传统的文本分类系统是经过大量监督学习完成的性能提升，成本较高反应时间较慢。具有不同信息量的文本有着随机分布，而信息量的不同会极大地影响模型的准确性，捕获和识别重要应急信息的结果大多较差。因此，使用主动学习模型来应用少量数据并提高准确性是很重要的。

* + 1. 代表性

在大规模的无标记数据中提取丰富的特征表示的能力可以减少紧急事件多分类主动学习问题中的大部分偏差。通过主动选择样本进行标记，模型可以在每次迭代中选择最有价值和最有代表性的样本，以提高模型的性能。因此选择有效的acquisition function去提取具有代表性的数据对性能的影响是最大的，也是最值得改进的。

* + 1. 健全性

原始模型在处理文本分类任务时，往往表现出过度自信的倾向。在模型输入未知内容时，导致其作出错误且不可靠的预测。而在许多实际应用中，如灾难响应、医疗判断等，这种错误的预测可能会带来严重的后果。因此，为模型引入一定的不确定性，从而减少其过度自信的倾向，成为了一个重要的研究课题。另一方面，我们可以在不引入过多计算负担的情况下，提高模型的健全性。这对于神经网络在需要高度准确和可靠预测的场景中，具有非常高的价值。

* 1. 实验方法

结合背景以及动机，此论文最终设计了一个基于池的人在循环中的主动学习文本分类系统。同传统文本分类模型不同的是，我们用少量的标注数据去训练，用训练后的模型结合不同的acquisition functions筛选的用例。然后，使用预设注释或者人类或者OepnAI注释者在循环中对提取的样例进行观察并分析关键特征，对提取的样例给予一个标签以及统一的解释。这些新用例将加入到原数据中，重复上面循环，直到性能达到全数据（训练集占比接近１，标注了标签以及９个默认的解释）性能。

为了观察性能在不同查询策略下的表现，实现中将采用基于Least Confidence的不确定性抽样，sentiment diversity sampling以及作为基准的随机采样方式。具体的acquisition function将在第2章进行细节描述。除此以外，在每次循环中，拼接了解释的文本都将使用预训练模型去预训练生成文本的深层次语义表示，并将预训练后的结果作为神经网络文本分类器的输入。

最后，将对传统神经网络模型进行结构上的改变，使用dropout 这种方式使得模型对于输入的小变化更加稳健，进而可以提高模型的泛化性能。此外，dropout 也可以减轻模型过分自信的问题，因为它引入了噪声和随机性，使得模型的预测结果不再过于确定。尽管 dropout 并不能直接量化不确定性，但其提供的随机性和噪声可以增加模型的健壮性，并在一定程度上缓解模型过分自信的问题。由于原始Bayesian Active Learning by Disagreement （BALD）算法中求熵的计算成本过高，所以升级了BALD算法，利用Monte Carlo Dropout(MCD)的同时使用Least Confidence来选择模型预测概率最均匀（即没有预测概率特别高的类别）的样本。

1.3.1目标

基于上述实验描述，我们的目标总结如下。

•开发一个基于池的人在循环中框架，并探索其有效性。

•构建注释器仿真流程，设计循环中止标准（例如达到全数据下模型性能后再运行一次）。

•生成基线（无主动学习下全数据的模型性能或无标注的随机抽样），研究主动学习是否可以提高文本分类模型的性能。比较不同查询算法抽样的影响。

•探索了解释质量以及数量对本系统的影响。例如，提供具有噪声的解释们或者引入随机性来代替精确的解释。

1.3.2 挑战

这个项目的主要挑战是不同的查询策略是否有很大的差别，以及如何改进原始ExpBERT文本多分类器的初始化框架以及训练框架来达到计算量的减少。

•值得考虑的是，主动学习文本分类系统存在多种查询策略，查询策略中不同算法的应用极大地影响了主动学习框架的性能。查询策略的选择需要从时间复杂性和代表性两方面来考虑。

•其次是带有dropout机制的神经网络在训练算法执行中以及利用MC dropout主动学习时的时间消耗可能会引起计算量的增加，因此如何在每次循环中减少计算量来提高反应效率是一个挑战。

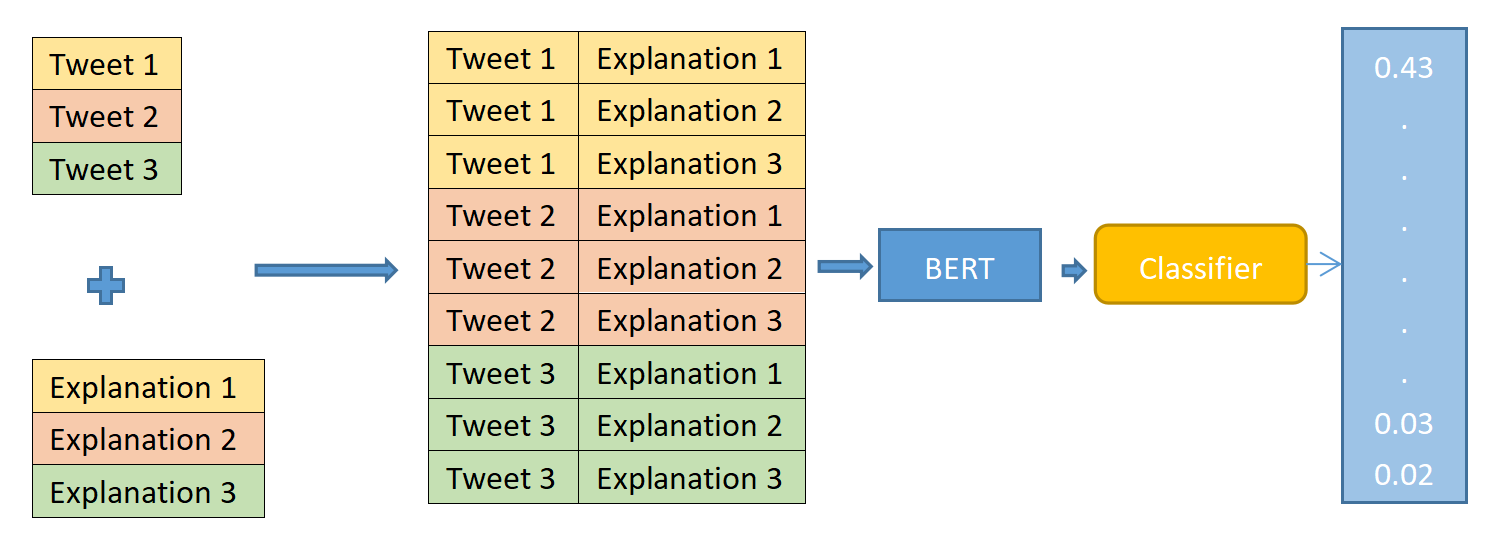
•最后对于框架设计，如何适当地设置采样的比例，迭代的次数以及每次迭代提供解释的数量等需要研究来达到满意的性能。

2 背景

本章介绍了实现利用主动学习提升基于ExpBERT文本分类器性能的关键技术，首先介绍了实验使用的预训练模型（2.1 section)，这个模型将应用在数据集的与训练中。人在循环中（2.2 section）结合基于主动学习及其应用的acquisition function(2.3 section和2.4节)，是本实验的重点实现部分，通过这项技术可以完成性能的提升。2.5节描述了多模型整合方式，以及2.6节中利用Openai模型模拟人在循环中的技术。最后探讨将贝叶斯理论应用在主动学习的可行性。

2.1 Representation Engineering with Natural Language Explanations (ExpBERT)

Representation Engineering with Natural Language Explanations (ExpBERT)模型提出了一种增强语言模型知识融合能力的方法。其将人类或者在本实验中的注释者提供的固定解释与推文相结合，从这些解释中学习以改善模型的性能[2]。图 1 直观地展示了如何将带有解释的样本与 BERT 模型相结合。解释在这里起到了关键的作用，解释的质量远比解释的数量对ExpBERT的性能影响更大，通过运用高质量解释可以引导模型的学习。



首先，将每条推文与一组预先编写的解释完全连接起来，生成一个3×3的集合。这些解释与特定推文无关，每个推文都会连接相同数量的解释。预处理后的文本和解释被输入BERT被用来生成对每个解释进行“解释”的特征[new 1]。分类器可以对输入和解释的表示进行分类训练~~，该分类器使用多层感知机（MLP）模型，输入是输入的表示w(x) 和解释的表示z(x)。~~

~~f(x) = MLP (w(x), z(x))~~

在MultiNLI自然语言推理数据集上微调后的BERT模型将为每个输入样本生成一个特征向量，代表长度为786的整个输入。然后将推文和解释的特征向量连接起来，形成一个大小为768 \* E的模型，其中E是解释的数量，并将其作为分类器模型训练和预测的输入数据。本论文将使用该模型作为基础预训练模型，同时使用Natural Language Inference（NLI）技术将嵌入量减少，并将其集成和初始化到"人在回路"系统中，因为该模型可以处理带有解释的实例向量，优化后的嵌入表示作为分类模型的输入可以提高模型的性能。

2.2 HITL

传统的自然语言处理流程不是为了充分利用人类反馈而设计的。与此不同，人在回路（human-in-the-loop）作为交互系统的关键组成部分，可以通过模拟人在循环中的角色来揭示模型的缺陷，这些缺陷在真实世界测试之前可能并不显而易见[11]。Godbole等人（2004）[12]采用支持向量机（SVM）主动学习，扩展了一个文本分类器。他们巧妙地将人类在特征工程、术语选择、文档标签等方面的输入纳入模型中，以做出统计上合理的判断。这种新型的人与机器学习算法之间的互动方式被称为人在回路（human-in-the-loop）机器学习[13]。在此方法中，根据不同的需求和场景，人和机器之间不同类型的协作，可以确定不同的HITL机器学习方案(Mosqueira-Rey et al. 2023)[14]。

主动学习（AL） (Settles 2009)[15]：主动学习的关键是系统保持着控制模型学习的过程，虽然人类作为媒介来参与对未标记的数据的注释，但是人类无法根据偏好选择无标签数据。AL将作为优化框架应用在本论文中，在2.3节中将详细介绍主动学习的应用。

交互式机器学习（IML）（Amershi et al. 2014）[16]是一种人类与系统保持紧密交互的方法。在此背景下，主动学习（AL）和交互式机器学习（IML）之间存在一些关键区别。Dudley和Kristensson（2018）[17]指出，虽然AL和IML都着重于由用户选择新的数据点进行标记，它们的驱动方式却不同。具体来说，在AL中，选择是由模型本身推动的，而在IML中，选择则是由用户推动的。由于IML是在AL的基础上发展起来的，因此它们有一些共同的缺点。然而，IML还具有一些特有的问题，特别是与人机交互技术（HCI）的复杂性有关。这些特有的问题需要更多专门的研究来解决（Michael et al. (2020)）[18]。

机器教学（MT）（Ramos et al. 2020）[19]是一种通过人类教师的指导来训练机器学习模型的方法。具体来说，人类教师会界定他们打算传递给模型的知识，强调教师在学习过程中的主动参与和指导。Devidze et al. (2020)[20]指出，与主动学习相比，MT更依赖于教师的专业知识，同时在样本选择和处理复杂任务方面的灵活性较差。因此，根据具体的需求选择适当的学习方法显得尤为重要。

2.3 Active Learning

作为HITL中最热门的学习方案，主动学习系统通过向未标记的实例提出问题并由专家（例如人工标注者）进行标记，试图克服标记瓶颈[21]。简单来说，主动学习致力于寻找最具信息价值的未标记原始数据，并将其交给注释者进行标记，这一过程更贴近实际从源数据中提取信息的场景。注释者标记这些源数据，将已经标记的实例加入到模型的训练过程。这样，主动学习能够通过使用更少的标注数据来实现全量数据下的性能表现，从而克服了标签瓶颈问题。在此项目中实例数量庞大、噪声较多、全量训练任务沉重，且准确率不高，主动学习方法就显得尤为适用。因此主动学习适用于此项目去专注于那些可能增加准确率的实例，减少无相关实例，提高紧急反应系统学习效率以及准确率。

2.3.1 The AL process and scenarios

主动学习的进程如图一所示，图一为基于池的主动学习的进程。首先，利用已标记的实例集L，对机器学习模型进行初始化训练。然后，模型对未标记的样本集进行打分，根据查询策略选择具有代表性的未标记样本，提供给人类标注者。人工标注者对所选样本进行标注，标注后的样本将从未标记样本集U移除，加入到已标记样本集L中。通过反复上述步骤，已标记样本的数量逐渐增加，模型性能不断提高。最后一旦达到终止标准（例如达到全数据量下的模型性能），主动学习过程将中止。

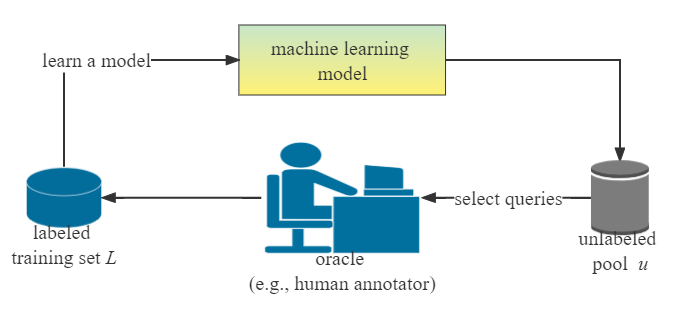


Figure 2: 基于池的主动学习过程

根据Settles (2009) [15]的分类，主动学习主要有三种场景设置：membership query synthesis[22, 23]，stream-based selective sampling[24]，以及pool-based active learning[25]。

Membership query synthesis: 这种场景的主动学习策略是由模型生成的。即机器学习模型可以请求标记任何未标记的实例，其并未在潜在的自然分布中取样[15]。对于比较绝对的问题领域，例如对绝对坐标采用回归预测的任务中，此查询方式通常是有效的。因为它可以解析简单的数据分布，构造合理数据供人类标注。但是, Baum and Lang (1992)[26]的研究中可以看出对于自然语言处理这类复杂任务中，模型可能会产生不易理解的文本串，导致人类无法对这些混乱文字进行判断。在深度主动学习背景中，可以通过生成对抗网络（GANs）用于数据增强来解决成员查询合成的场景，因为GANs能够生成合理度高的实例[27]。

Stream-Based Selective Sampling：此场景设置相比于Membership query synthesis的不同之处在于，此方式可以不受输入分布的影响。基于流的抽样是一次从实际的分布中抽取一个未标记实例，模型通过“信息性度量”或“查询策略“来决定是否请求实例的标签，相当于一种带有偏见的随机抽样[28]。

pool-based AL: 作为普遍用于主动学习的设置，pool-based AL与基于流的采样的区别在于，pool-based AL在选择最佳查询之前采取贪婪机制，对整个数据集进行比较，但是后者单独接收数据进行评估。但是，当数据集非常大时，由于需要对全部数据进行评估，可能导致计算耗时。除此以外，此方法适用于人工标注成本较大的场景，通过选择最有价值的数据进行标注，从而最大化标注投入的价值。

2.3.2 Acquisition Functions

本节考虑主动学习中的各种acquisition function，在论文设计（第三章）部分中将使用基于ExpBERT的文本多分类模型结合不同acquisition functions实现性能的提升。

随机抽样

随机抽样随机选择实例，既不基于预测，也不基于数据以及模型，通常被用作任务的基准。因为，它可能忽略了潜在地信息，从而降低了学习效率。在这种情况下，随机抽样将作为底线与下面提到的更复杂的策略对比，特别是当标记池过于庞大时[29]。

不确定性抽样

不确定性抽样是一种特殊的主动学习策略，通过查询最难以判定其分类的实例，进行标注后提高模型判定准确度。在用概率模型做二分类问题中，这种实例的后验正概率最接近0.5[30]。但是对于更复杂多标签分类问题，将使用基于熵的方法。概率分布越均匀，熵越大，随机变量的不确定性越高，信息量越大。当概率集中在几个数据点，表示不确定性越低，信息量越小。



表示在分类的概率分布。

在文本分类领域，对于不确定性的测量，常用一种替代方式，即Least Confidence[31]：



表示可能性最大类标签。对于二值文本分类，该方法等价于基于熵的算法有效性。考虑那些模型预测概率最大但是可信度较低的样本。具体方式是通过选择最大概率最小的样本进行标注，此更全面的考量方式将在本论文中采用。

除了上面两种较为常用的测量方式之外，（Munro 2020）[13]提到了margin of confidence 以及ratio of confidence这两种方式，前者是两个最自信的预测之间的差异，后者则是这两个预测的比率。



是最自信，是次自信。

Semantic-based diversity sampling：

Peng, Hao, et al.（2023）[32]提出了一种基于语义的多样抽样方式，该方式可应用在文本分类中。与采用不确定性抽样的测量置信度的方式的不同之处在于，基于语义的多样抽样的方式是利用欧氏距离在语义上消除文本样本的冗余。在实践中，这种距离度量常用于各种领域，包括数据聚类和嵌入空间中的近邻搜索。在机器学习中，计算一个嵌入到一个聚类中心的欧氏距离的这种方式，可以为每一个嵌入到每一个聚类中心计算一个距离矩阵，其中"distances[i, j]"代表第i个嵌入到第j个聚类中心的欧氏距离[33]。

为确保在后续过程中提供给模型（学习者）更丰富的，重复度更低的样本，此抽象方式使用（Sener和Saveravarese，2017）[34]的贪婪k中心算法进行聚类操作。数据集包含个无标签的文本，并将分成个批次，每个样本集包含m个实例。是数据集编码的结果。首先选择vectors从中去初始化聚类，这里examples将被认为是聚类中心。然后适用k中心算法searches from ,这是一个包括不在中的成员的集合。寻找距离现有聚类中心最远的样本。通过这种方式，可以逐渐构建一个代表各种不同文本类型的样本集合，从而确保模型在学习过程中接触到尽可能多样化的内容。算法公式如下：



Where：



随后，通过将现有的聚类更新为。经过循环执并将输出结果合并到中。中的所有文本实例会收敛成一个核心集合，在语义空间中最好地体现并概括数据集。

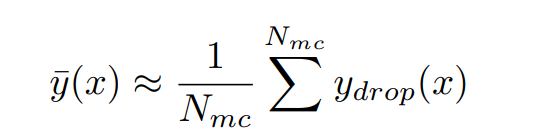
Bayesian active learning by Disagreement：

BALD（Bayesian Active Learning by Disagreement）是文本分类中的一种主动学习策略，它结合了贝叶斯推断和不确定性度量来选择信息丰富的样本。在BALD中通过估算每个样本的类别概率分布，计算预测间的不一致性来衡量不确定性（Houlsby, N（2011））[35]。学习者（模型）通过输入x最大化模型参数的不确定性，H[y|x,D]表示了目标变量的不确定性，等式第二项表示在参数 θ 服从训练数据集 D 的后验概率分布 p(θ|D) 的条件下，对于 H[y|x, θ] 的不确定性（熵）进行期望值的计算，衡量了平均的不确定性。



为了更好地计算预测间的不一致性 ，Gal等人在2017年提出具体抽样函数使用 Monte Carlo dropou(MCD)t 方法，[36]。MCD是一种训练过程中的正则化方法，通过在训练过程中随机关闭一些神经元，来模拟贝叶斯网络的后验分布[37]。从而可以通过对每个样本的多次预测并计算模型的预测之间的不一致性来选择具有最高不确定性的样本进行标注。

MC dropout本质是通过概率性停用部分神经元以防止过拟合的正则化方法，在推断过程中，通过有限次（T次）对神经网络进行输入，使用dropout来在预测阶段产生不同的结果[38]。因此，这些不同预测结果进行不确定性量化后，可以用来估计模型对未标记样本的不确定性，并应用在主动学习的Bayesian Active Learning by Disagreement(BALD)抽样策略中，来选择不确定性最大的未标注数据。MC dropout的推断结果通过以下方式得出：



其中，ydrop是dropout网络的输出的不同结果，x是网络的输入，Nmc是获取分布所需的采样次数。最后，BALD通过比较每个样本的BALD值，选出具有最大信息增益的样本进行标注，并将其加入已标注的训练集。除此以外，BALD可以和前文提到的Least Confidence代替原始使用熵的计算，来选择模型预测概率最均匀（即没有预测概率特别高的类别）的样本。

Small-Text Library

本项目中的文本分类器倾向于专注于一个模型，因此可能错过其他可行模型的应用。然而，切换模型和主动学习策略的时间成本，以及代码的冗余性，将会显著影响实验的进展。小型文本库整合了scikit-learn、transformers和PyTorch等可以在Python环境中应用的常用库[39]。基于池的主动学习的文本分类架构连接了查询策略、分类器和中止策略的接口。它不仅为文本分类工作提供了先进的主动学习框架，而且提供了一系列的分类器和查询策略组件，以便于可以混合搭配用于实验和应用的主动学习任务，使主动学习在Python生态系统中易于实施。与最常用的ModAL库[40]相比，Small-Text提供了更灵活的定制服务，前者更专注于模型集成和查询策略的选择。然而，这种包罗万象的方法并不一定会改善基于ExpBERT的项目的性能，因为性能差距会因情况而异。

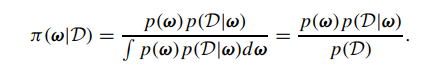
GPT annotator simulation explanation generation

GPT（生成预训练变换器）是OpenAI团队在2018年开发的一种预训练语言模型[41]。基本算法是Transformer，这是一种基于自我注意机制的深度神经网络结构，具有强大的序列建模和表示学习过程。通过预训练和微调，该模型可以分析和生成自然语言文本，并在多个场景中有所帮助，例如自动回答、智能客户支持、语言翻译等。因此，作为一种节省时间和高效的文本解释系统，该项目将使用此模型作为基于ExpBERT的主动学习注释器。主要任务是为ExpBERT生成合适的解释，并将它们提供给模型以帮助提高分类性能。在此配置中的主动学习通过利用GPT强大的生成模型[42]根据输入文本生成人类可读的解释。常见的方法是使用GPT对抽样出的特定类别的文本进行关键字抽取，结合关键字集合成解释，然后选择其中一个或多个作为解释加入到原有解释中，最后结合文本传递给ExpBERT模型。这样通过对抽样出的文本进行分析，可以弥补人力记忆上的以及语义相似度计算能力不足的缺陷，从而找到最能帮助模型理解数据内在结构的重要特征词句。

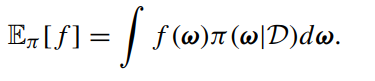
Active Learning for 贝叶斯Neural Networks

在现代的机器学习和深度学习应用中，贝叶斯神经网络以其灵活的推理方式成为一个关键组件。通过引入权重和预测的不确定性，贝叶斯神经网络提供了一种丰富和灵活的建模框架。这不仅增强了模型的可靠性，还使模型能够更好地泛化、适应噪声和异常值。与传统的神经网络不同之处在于，贝叶斯神经网络中的权值不是固定不变的。该模型的思想是将神经网络和随机建模的优势结合起来，可以为预测提供概率上的保证[43]。这里的权值是基于我们观察到的来推断的，是一个逆概率问题，需要通过贝叶斯定理来解决[44]。这里权重w代表的是无法直接观察分布的不可见变量，无法直接观察真实分布，贝叶斯允许我们得到模型参数在已观察到的数据条件下的分布p(ω|D)，即后验分布。

同时，p(D|w) 似然function在多分类问题里将神经网络的预测（经过Log Softmax转换）与实际观测类别标签相结合，来量化他们之间的一致性。在确定似然项以及p(w)先验分布后，便可根据贝叶斯理论计算后验分布，



得到后验分布后，将使用边缘化方法进行预测：



贝叶斯神经网络提供了一种不确定性的量化方式，使得主动学习系统能够识别模型在哪些样本上最没有信心，从而选择这些样本进行人工标注。但是，对模型参数进行积分通常需要进行昂贵的计算。例如，markov-chain monte carlo（MCMC）来估计后验分布。MCMC是目前最流行的基于采样来估计后验分布的方式，主要思想是构建前后样本的依赖关系[78]，进而模拟分布。但是由于连续样本可能出现自相关的情况，则需要更加庞大的样本集来抽样去获得近似独立样本，这样做会增加计算成本。因此，对于BNN来说，Metropolis-Hasting算法更加有效[文献]。通过与分布成比例的函数，省去获取精确概率分布的过程，并使用接受-拒绝机制来完成对候选样本的选取。如果样本比前一个数据点出现概率低则拒绝样本。除此以外，汉密尔顿蒙特卡罗算法（HMC）[81]是更加改进的方式，通过更新提议来拒绝少量样本的同时减少相关性来缩短burn-in时间。

这在面对大规模数据集时可能会成为问题，特别是在主动学习的上下文中，通常需要进行多次迭代和模型更新，所以真正实现BNN会延长应急系统的启动时间[45]。

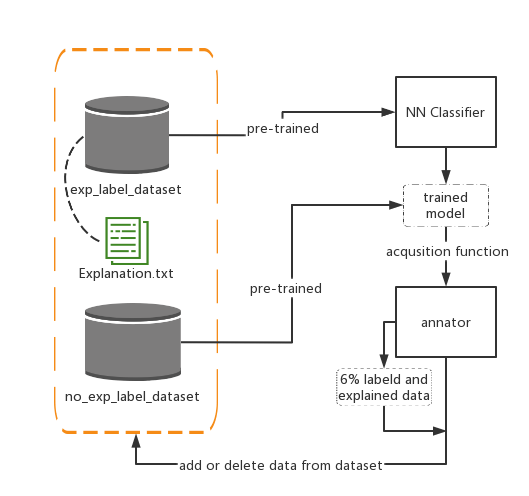
3 设计

设计这一章将对主动学习框架进行一个概括（3.1节），然后按照主动学习任务的执行顺序来一一拆解并设计。从对数据进行预训练开始（3.2节）再到获得嵌入表示的分类器模型的设计（3.3节），接着3.4节介绍了使用训练好的分类器模型进行了多个角度的主动学习抽样策略的设计并分析了不同抽样策略的实施方式以及作用。最后进行注释者的模拟功能的设计（3.5节）。

3.1 整体流程

整体实验框架基于第二章提到的基于池的主动学习技术而搭建的，从固定的未标记的样本池中进行抽样，利用训练好的模型来评估样本。与传统主动学习不同的是，我们改变了对数据集的要求，引入了带标签及解释（exp\_label\_dataset）和无标签及解释(no\_exp\_label\_dataset)的两个数据集。exp\_label\_dataset中包括用于测试，验证以及训练的数据集。为了充分利用ExpBERT预训练模型，每轮标注不仅提供标签，还会增加解释到默认解释集Explanatio.txt中。三个主要集合将全称参与到三大模块中，不断迭代，这三个模块分别是训练，标注以及更新：

训练： 初始训练阶段使用占全数据量百分之二十的带标签及解释数据集，exp\_label\_dataset中的每个样本连接了默认的9个解释（标签描述）进行第一轮训练。第一轮训练后，训练好的模型将对no\_exp\_label\_dataset数据集进行评估，并通过不确定性或多样性等角度计算每个样本的“信息量”，然后选择“信息量”最大的数据进行标注。整体的主动学习的架构设计及流程如下图所示。



标注：图中的标注过程中注释者会总结出抽样文本的关键句作为解释，并提供每个文本的分类标签。增加后的解释数量将大于默认解释数量，更新解释集Explanation.txt。使用acquisition function进行量化后，我们将抽样出的数据进行全标注后加入到exp\_label\_dataset中，形成下一轮迭代所需的数据集，这些数据集均会被ExpBERT模型预处理后输入到神经网络文本分类器中。

更新数据集：然后把标注了的6%的数据从no\_exp\_label\_dataset数据集中移除。每轮exp\_label\_dataset的数据量会增加百分之六，解释会增加一个或多个，反复执行上述过程，直到验证集的评估分数接近或超过全数据量下的评估分数。

3.2 预训练

根据上方框架设计，所有数据集在文本分类器的训练前，需要进行预训练过程。State for 第二章对ExpBERT的介绍，预训练模型通常在大量的文本数据上进行训练，可以捕获了深层次的双向的语言表示，因此在特定的任务上往往能实现更高的性能。在许多的文本分类情况中，使用预训练模型作为基础并进行微调，可以大大减少文本分类模型的训练时间。通常将文本输入模型后获取其的嵌入表示（embeddings）。这种嵌入表示可以用作下游文本分类器的输入。

已标注数据集在主动学习迭代时对应的解释量将增加new\_exp个，因此，在每次迭代运行到预训练这一步时需要重新将更新后的训练集，测试集以及校验集的文本与更新后的解释全部连接来作为预训练模型的输入，因此每一次预训练模型将多处理num\_tweets\*new\_exp数量的数据。这也是此实验在实现中耗时时间较长的部分。

对于未标注数据集，为了更趋近于现实中的情况，进行预训练时仅需要对文本数据进行处理。但是，训练好的神经网络期望接受与训练时相同维度的输入。因此，对于未标注数据的嵌入表示，将使用F.pad(embedding，(0,pad\_amount))方式确保两个数据集的嵌入表示的张量匹配。

由于原有的模型下获取的嵌入表示量级过于庞大，导致运算时间拉长。因此，在训练过程中采用（自然语言推理NLI）来微调BERT，通过NLI将给出三个大小的向量输出，使其一个样本的嵌入大小仅为768加上解释数量和文本的三倍。对于未微调模型，推文与解释全连接的扩展数据集将直接通过BERT传递。除此以外，为了提高每次迭代的计算效率，预训练使用基于池的方式提高计算速度。利用Python的‘multiprocessing’库的‘Pool’类来并行处理数据的批次。在多个核心的机器上，多个批次可以同时处理，灵活地控制进程并提高处理效率。

3.3 分类器模型设计

根据上面的框架图，我们采用神经网络作为分类器模型。评估阶段将对比传统神经网络分类器以及带有dropout的神经网络分类器结合主动学习技术的表现，此实验将对传统的神经网络进行结构上的调整。最终的模型结构将采用PyTorch[到时候插入文献]来搭建。

3.3.1 Before dropout:

在模型结构上，本实验的神经网络使用了一层的隐藏层以及100个神经单元，输出特征数量即‘output\_size’等于9（标签种类），向前传播的设计可以表示为下面的这个公式，wi和bi分别是线性层i的权重和偏置，中间经过激活函数后得到模型的输出即y。此实验将使用ReLU非线性激活函数，导致网络稀疏性从而缓和过拟合问题。

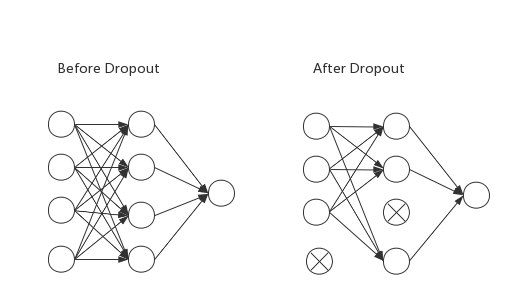


在训练方法上，普通神经网络使用了标准的损失计算方法（例如交叉熵[文献]），除此之外，前向传播、损失计算、反向传播和优化的这些步骤是分开执行的，即使用了PyTorch的标准方法。

3.3.2 After dropout：

此论文对模型的结构进行了改进，使用了dropout神经网络模型作为文本分类器。Hinton 提到 Dropout 可以防止神经元过度依赖网络中的其他特定神经元，从而提高其独立性并增强模型的泛化能力[文献]。将dropout应用于神经网络相当于用经过dropout仍然存活的单元组成的“稀疏”网络。一个有n个单元的神经网络可以有2^n个可能的系数神经网络的集合。对于每一次呈现的每个训练案例，都会采样并训练一个新的稀疏网络。[文献dropout]。

Dropout通过随机的丢弃单元来提供有效地近似不同神经网络结构，丢弃过程如下图所示，通过将部分神经元从网络中移除以及断链来实现。



Dropout Neural Net Model. Left: A standard neural net with 1 hidden layer. Right:

An example of a thinned net produced by applying dropout to the network on the left.

在本论文中将’dropout\_prob’参数在神经网络初始化时设置为0.2，即神经元在每次向前传播时被临时从网络中移除的概率是百分之二十。这时，原始神经网络的向前传播公式将改变：

h = W1\*x + b1，a = activation\_fn(h)，r = Bernoulli(p), a\_dropout = r \* a

y = W2\*a\_dropout + b2

这里的Bernoulli(p)代表服从伯努利分布的随机变量，"\*"代表元素级的相乘，即随机地将向量a中的一些元素设为0。

3.4 抽样策略

我们在神经网络分类器中采用了四种不同的抽样策略来提升模型性能：随机抽样作为基准；基于"Least Confidence"的不确定性抽样；基于情绪的多样性抽样；以及基于MC dropout的贝叶斯主动学习异议抽样（Bayesian Active Learning by Disagreement, BALD）。这些策略各自通过不同的途径挖掘和利用数据的独特属性，作为主动学习框架的核心共同促进了模型学习的效率和准确性。通过对不同角度的抽样来发掘对模型性能提升最好的方式。

3.4.1 随机抽样：

抽样过程中，acquisition function将对接近占全数据集75%的未标注数据集进行评估，每次评估给出6%的未标注数据集交给注释者来进行标注。随机抽样来确保没有任何偏见的样本选择。通过设置随机种子直接对未标注数据集抽样，并未将初训练后的模型纳入考虑，在其影响下的模型性能将作为基线。

3.4.2 基于"Least Confidence"的不确定性抽样：

此论文采用了Least Confidence方式来计算不确定性，As described in第二章，对多标签文本分类问题Least Confidence常常作为计算熵的替代方式。在此实验中，通过将模型设置为评估模式，在模型给出的概率分布‘prob\_dist’中寻找最大的概率值作为不确定得分，计算标签数量num\_labels并将最有信心的预测进行标准化，标准化公式如下，normalized\_lc得到标准化的最小信心值。

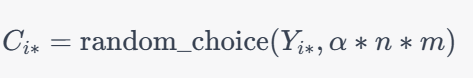
normalized\_lc= (1 - max(prob\_dist)) \* (num\_labels / (num\_labels - 1))

通过最小信心值升序排序，选择前6%的的样本作为此不确定方法的输出。

3.4.3基于语义的多样性抽样：

在本研究中，采用了一种基于语义的多样性的抽样方式。第二章中介绍的这种方式可以在学习过程利用不同于不确定性抽样的角度来找出丰富且低重复的样本。对于实验的数据集，每个类别的样本数量存在显著的不平衡，一些类别的样本数量仅占全数据集的2%，这种数据不平衡可能会导致某些类别的样本训练不充分，并且在评估过程中产生较大的不确定性。如果仅根据不确定性来进行样本抽样，可能会出现对某一类别的样本过度抽样的问题，这样会导致抽样结果的类别分布失衡，从而影响模型的学习效果。因此，为了在主动学习过程中能够平衡利用各类别的样本，实验采取了一种基于语义多样性的抽样策略。

此方式需要加载无标注数据集使用预训练后的嵌入向量，通过将训练后的神经网络模型设置为评估模式，来对输入的嵌入向量计算对应的概率。根据第二章的介绍将每个文本的嵌入向量记作 $Y\_{i\*}$，其中 $i$ 代表文本的索引。假设将数据集分为n个批次，每个批次有m个实例。首先，随机选则一部分嵌入向量作为初始聚类的中心$C\_{i\*}$，$C\_{i\*}$在次实验中如下表示，这里将初始聚类中心占全数据的比例即 $\alpha$设置为0.2：



然后，利用贪婪k-中心算法来搜索 $Y\_{i\*}$ 中不在 $C\_{i\*}$ 中的嵌入向量，找到距离现有聚类中心最远的嵌入向量，并将聚类中心的集合更新。这个过程可以表示为：



Dist用来计算嵌入向量和聚类中心的语义上的距离，最后，我们返回 $C\_{i\*}$ 中每个聚类中心在 $Y\_{i\*}$ 中的索引，记作 $O\_i$，这样我们就可以从未标注的数据集中抽取出对应的样本。

3.4.4 MC Dropout-Least Confidence BALD

在本研究中，采用了一种基于蒙特卡洛（Monte Carlo）dropout的抽样策略，以寻找那些模型不确定性较大的样本，对模型进行更有效的训练。此策略的基础是Gal 和 Ghahramani 在 2016 年的一项研究[文献]，他们提出在训练神经网络时可以使用dropout作为一种近似贝叶斯推断的方式。

与其他主动学习相同的是加载了无标注数据集的嵌入表示E，不同之处在于文本分类模型为带有dropout机制的神经网络，同时需要开启训练模式，而不是评估模式。因为，MC dropout需要通过训练开启dropout。经过J次预测，产出预测概率列表p\_i。可以表述为：



其中，p\_i$ 代表嵌入向量 $E\_i$ 的预测概率列表

不同于传统的BALD策略，实验使用最大化预测概率的负最大值来度量不确定性，而非常规的BALD计算方法。



接着，将所有样本的索引及其对应的BALD分数存储到一个列表中，并按照BALD分数从大到小进行排序。这样可以得到那些模型对其分类最不确定的样本，即BALD分数最高的样本。这种方法可以看作是融合了蒙特卡洛（Monte Carlo）Dropout和最小置信度（Least Confidence）策略的BALD（Bayesian Active Learning by Disagreement）算法的变种。当面对不平衡数据集时，模型往往会倾向于预测数量占主导的类别。通过使用最小置信度策略，即选择模型预测不确定性最大的样本，我们能在数据收集过程中优先考虑那些被模型低估的少数类别，进而改善模型在处理这些类别时的性能。相比与只应用了least confidence的不确定性抽样，这种方式能够捕捉到更全面的模型不确定性信息。

3.5 注释过程

在主动学习策略引导的抽样过程后，得到一组无标签但具有高信息含量的文本。对于这些文本，注释者需要根据其内容给出相应的分类，并决定在当前迭代过程中增加哪些具有引导性的解释。

3.5.1标签标注：

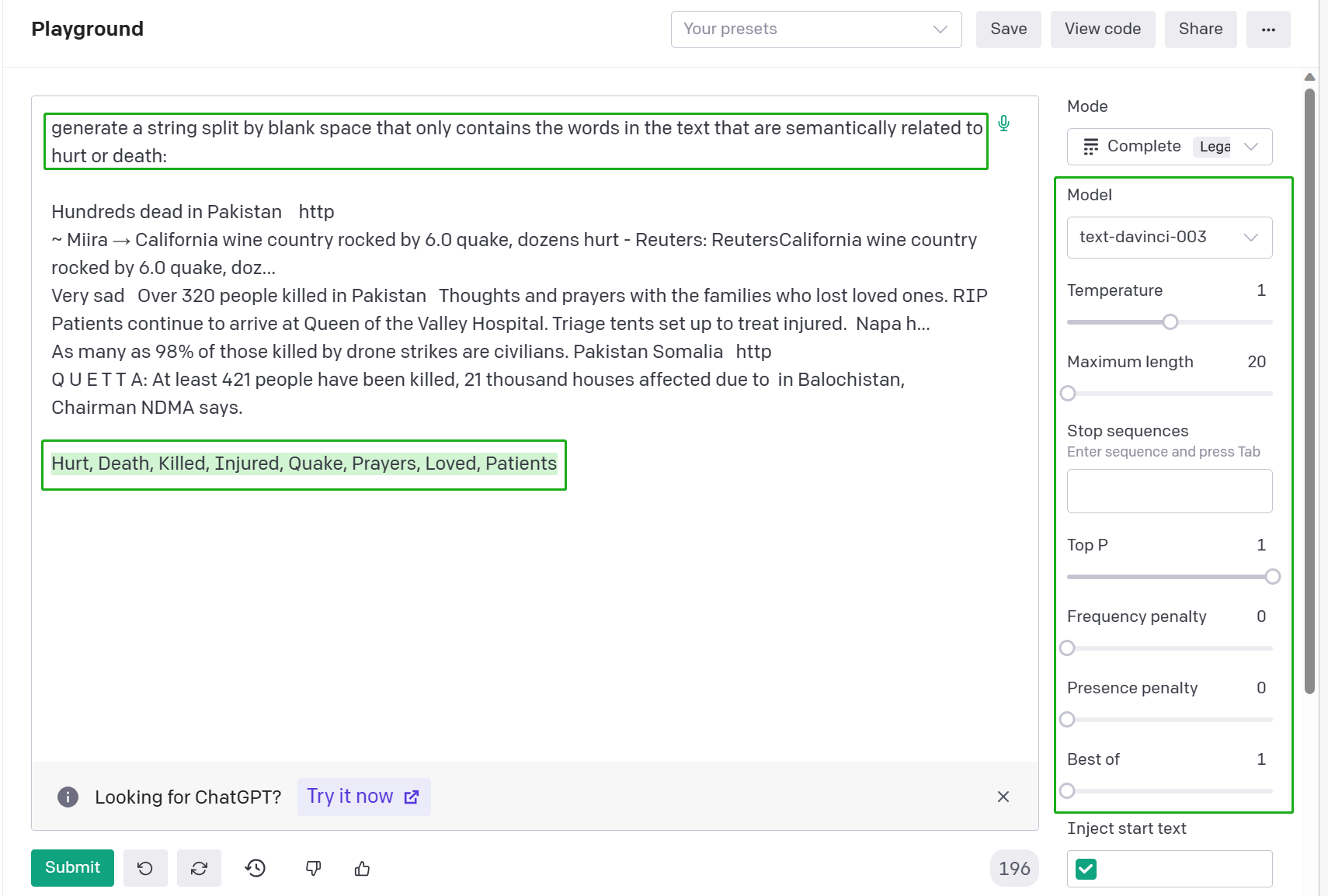
这项实验的重点是探究在主动学习过程中增加解释标注功能是否能提升文本分类器的性能。为了模拟理想情况，假设在每轮迭代中，标注者对文本的分类有完全的理解，因此提供的标签将准确的与源数据对应。在评估阶段，为了消除增加标签和文本数量可能对模型性能的影响，我们将对比两种情况：一种是只增加了标签的主动学习，另一种是同时增加了标签和解释的主动学习。

3.5.2解释标注：

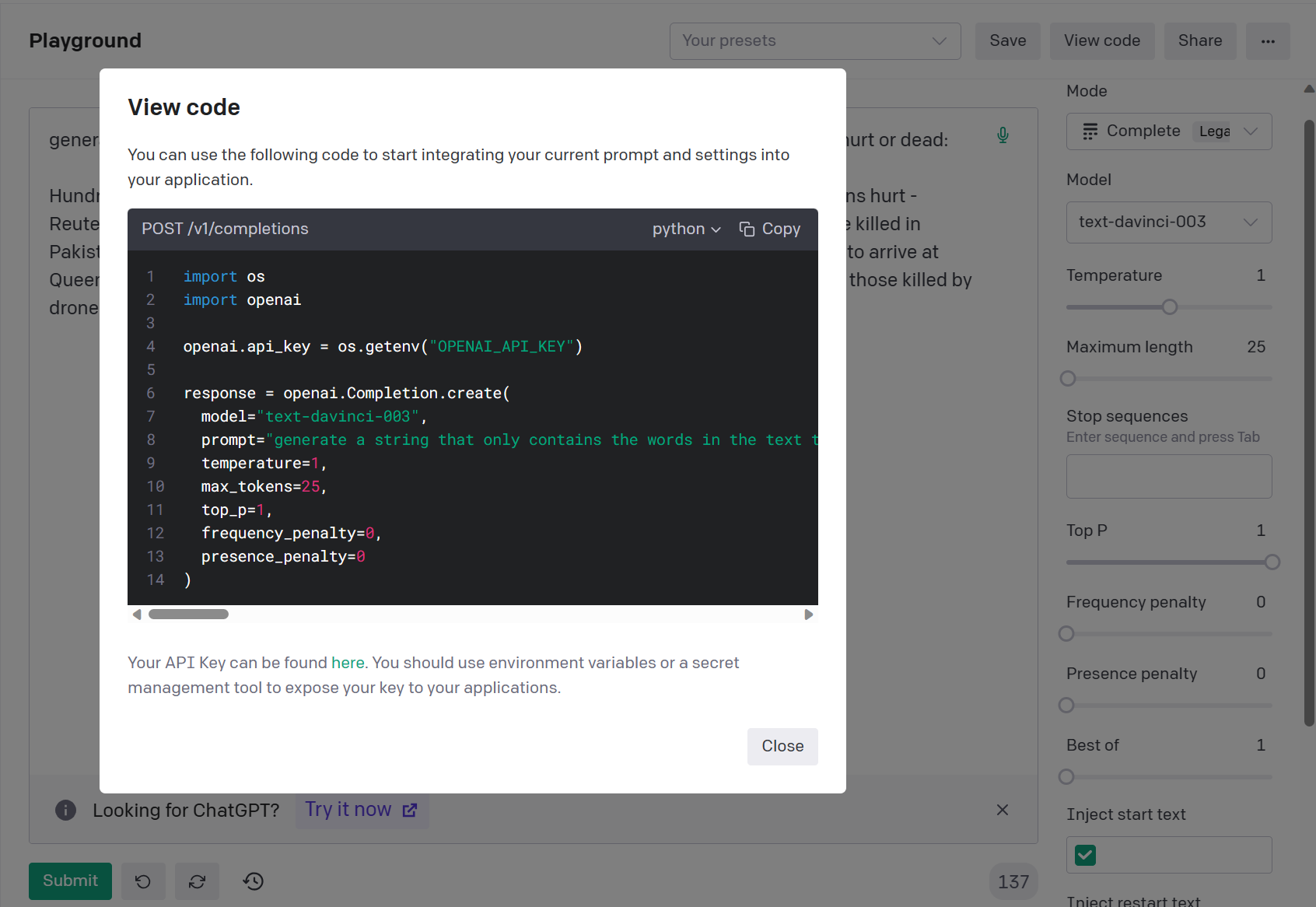
此实验设计了三种不同的解释标注策略。提供的解释将存放在此章架构图中显示的Explanation.txt集中，供后续预训练前的拼接文本的任务。系统操作人员可以根据人力分配或者成本控制的情况来选择这三种方式，从而更好地支持模型的学习过程：

第一种策略是通过预先查看每轮迭代中的文本，并模拟可能会增加的解释，就像注释者会做的那样。预设的解释将保存在'annotator.txt'文件中。执行到标注阶段时，自动从这个文件中抽取解释。尽管这种方式可以减少人力消耗，但它的适应性不足，只适用于本实验相关的数据集。

第二种是允许用户直接提供输入作为解释。这种策略要求用户根据选出的文本特性进行分析，给出他们自己的解释。在每次迭代中，注释者将专注于某一类别的十个突出的文本，以便快速的对这些文本进行分析。这种方式增加了人的判断，也适用于其他应急相关的数据，有一定的灵活性。



最后一种模式应用了OpenAI模型自动提供解释，来减少人力的消耗。此实验选择了"text-davinci-003"引擎来进行文本分析[文献]。同上图一样，通过openai接口的开发者平台的Playground可以将要输入到模型的prompt在Playground中进行模拟，右侧导航栏中可以选择模型，温度以及回复的最大长度。根据回复的合理性去选择表现最好的超参数。经过测试发现“text-davinci-003”远比“text-ada-001”的回复的合理性以及精确度好。经过调试发现当“温度”设置为1时，Top\_P设置为1以及最大回复长度为25时，模型的产出是最合理的。Playground提供了python环境下的配置代码，最终模型配置如下图所示。为了顺利连接openai接口需要进行openai key的申请，需要将这个键在生成回复前加载：



利用这个模型可以对某一类别的五个突出的文本进行关键词分析，选出语义上与这一类别的标签描述最接近的关键字去拼接成一个字符串。最终返回一个最能反映文本特性并能帮助模型从中学习的这个字符串作为增加的解释。但是，其缺点是对于一次性生成多个解释或者上百token的主动学习的需求并不友好，因为虽然减少了人力成本，但是每个token的生成会消耗一定资金。

4 实现：5-6页

4.1 实验环境

本实验使用了Python 3.9 作为自然语言处理任务的主要开发语言。其高适用于自然语言处理任务，Python拥有很多数据科学以及机器学习库，在本实验中提供了PyTorch, Pandas, Numpy等库带来的的数据分析，预处理和机器学习功能[文献]。Python的高兼容性可以在本实验中使用PyCharm连接远程High Performance Computing Systems(HPC) 或者使用Google Colab结合Google Drive来运行。本实验使用了PyTorch与Python集成来构建动态神经网络，结合了PyTorch的自动计算梯度，Dynamic Computational Graph的特性以及利用CUDA加速计算的能力可以保持神经网络模型的高灵活性以及准确性[文献]。

4.2数据集

4.2.1 CrisisNLP

在本实验中使用了CrisisNLP数据集，包含了数据id，文本信息，以及其所属分类的描述。一共有近16000条数据。需要对此数据集进行预处理，拆分，预训练以及分类。CrisisNLP是专门为人道主义应急和危机应对相关的自然语言处理（NLP）研究和开发的开源资源。数据集中包含多个大型的社交媒体的交互信息，涵盖了多种危机事件，例如台风，地震等需要有关部门做出及时反应的事件。信息的性质一共可以分为九类，标签描述以及分布情况如下表所示：

|  |  |  |
| --- | --- | --- |
| Label | Description | percentage |
| 0 | injured\_or\_dead\_people | 13% |
| 1 | missing\_trapped\_or\_found\_people | 2% |
| 2 | displaced\_people\_and\_evacuations | 3% |
| 3 | infrastructure\_and\_utilities\_damage | 8% |
| 4 | donation\_needs\_or\_offers\_or\_volunteering\_services | 14% |
| 5 | caution\_and\_advice | 6% |
| 6 | sympathy\_and\_emotional\_support | 11% |
| 7 | other\_useful\_information | 30% |
| 8 | not\_related\_or\_irrelevant | 13% |

根据表中的分布可以发现数据占比非常不平衡，标签7所占比例最高提供的信息也最为嘈杂，很好的体现了真实情况中的应急信息的分布状态。对文本分类任务而言是一个挑战。因此，这个数据集对于研究人员来说非常有价值，因为它记录了关于各种不同危机情况下人们的真实反应和行为对应的的数据以及分布。进而，研究人员能够开发出针对真实的数据不平衡的情况下更有效的系统，用来应对这些危机，例如通过自动检测危机相关的社交媒体帖子，或者通过分析社交媒体数据来预测危机的发展趋势等。

4.2.2数据准备

预处理：

在输入预训练模型之前，对数据集中的推文进行预处理是一项关键步骤。目标是最小化文本的噪声并提升模型的学习效率。首先，标准化网络用语，将非正式的缩写或俚语转换成标准形式，拆分驼峰命名的单词。接下来，利用emoji库替换推文中的表情符号为对应的文本形式，使模型更好地理解这些符号的含义。最后，清理或替换推文中的网页链接、用户名和“#”tag等，因为这些元素通常对模型的学习没有太大的帮助。这样的预处理步骤有助于降低推文中的噪声，提升模型的学习效率和表现。

数据集拆分：

针对主动学习任务需要将数据集拆分为未标注数据集以及带标注的数据集。同时，带标注的数据集将拆分为训练集，验证集以及测试集。主动学习需要每轮增加或者删减数据，但是，为了主动学习的每轮迭代都可以公平的测试模型在unseen 数据上的表现，因此测试集的数量不能随意变化。所以测试集，未标注数据集，标注数据集需要在主动学习循环外进行拆分。而训练集以及验证集将在每轮迭代中随着数据集的扩大而扩大，但是两者的比例不改变。由于标签1对应的数据量仅占全数据量的百分之二左右，为确保所有数据集中至少存在标签1对应的数据因此使用’StratifiedShuffleSplit’来保持原始数据中的类别分布，相比于随机拆分可以更好的评估分类器的性能。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 数量 | | 占比 |
| 已标注数据集 | Training dataset | 2890 | 18% |
| Validation dataset | 331 | 2% |
| Test dataset | 1606 | 10% |
| 未标注数据集 | 11241 | | 70% |

为了符合主动学习使用少数据量训练达到高性能的标准，在主动学习中第一次迭代时，已标注数据集的数据量不再是全数据量，如上表所示，总数仅为为全部数据集的百分之三十，长度只有4827。而未标注数据集的长度设置为11241。而用全数据量（训练集占比最高）来训练时，已标注数据集的训练以及验证集的数据量长度总和将为12845。测试集在全数据量下以及在主动学习需要的数据量中的数量不变，长度一直是1606。每次循环将会抽样数据集百分之四的数据用于标注。在后续的评估这一章中发现经过9次迭代模型性能便可接近全数据下校验集对应的模型性能。这时，未标注数据集的长度还剩6741。

4.3性能指标

性能指标的设定可以很大程度上影响主动学习项目的有效性。合适的评估标准可以衡量并对比不同主动学习策略以及模型框架对人在学习中的应急系统的影响。从而确定模型，主动学习策略和参数的选择。为了评估主动学习的性能，首先按照4.2.2节的标准对数据进行划分。然后，开始对训练集进行训练。在每轮基于池的主动学习迭代后，用设定好的性能指标对训练好的模型在测试集和验证集上进行性能评估。每轮迭代后，将根据性能指标绘制出学习曲线。

为了对模型的性能进行全面的评估，实验使用了两个性能指标：准确率和F1得分。准确率衡量了模型预测正确的实例数量占总实例数量的比例，是一个有一定偏见的评估标准。而F1得分是精确率（Precision）和召回率（Recall）的调和平均数，相比准确率来说消除了偏见，更常出现在多分类问题上，计算公式如下：

F1\_score = 2 / (1/Precision + 1/Recall)

其中，Precision是模型预测为正类的样本中真正为正类的比例，Recall是真正的正类样本中被模型预测为正类的比例。F1得分在处理CrisisNLP类似的不平衡数据集时，可以提供更全面的性能度量。

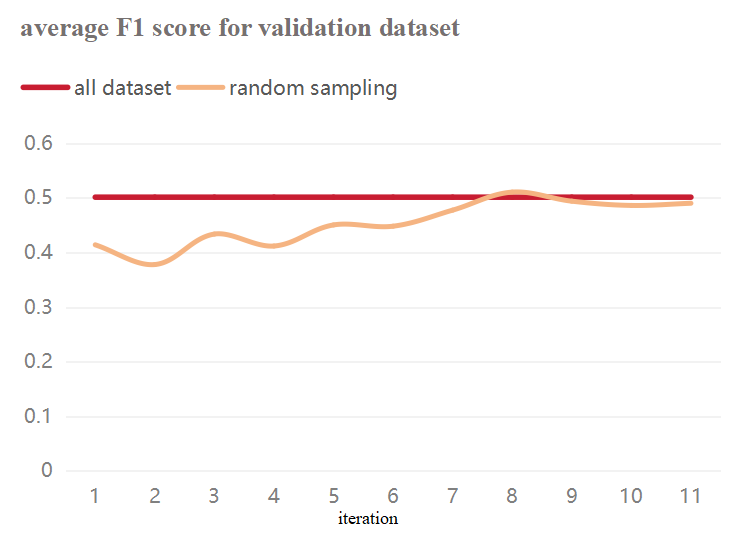
除此以外，学习曲线也可以作为性能的评估标准。通过观察学习曲线的变化，我们可以对模型的学习能力和稳定性有更深入的理解。使用了tensorboard来记录主动学习迭代过程的学习曲线。学习曲线可以表示模型的性能与用于训练的样本数量（主动学习迭代数量）的关系，评估带标注的数据集的数量的增加是否可以有效地提升模型的性能。除此以外，实验通过tensorboard也记录了训练过程中的每个epoch下的模型性能，确保模型是在每次主动学习迭代后为这次迭代对应的无过拟合下的最佳性能。

4.4 中止策略

在主动学习框架下，中止策略是用来决定何时停止学习过程或不再向标注者请求新数据的

略。中止策略的制定直接影响标注的成本、时间消耗以及模型的最终性能。虽然中止策略在某种意义上类似于模型的超参数，但它并不直接影响模型的训练过程。相反，它的主要目的是在保持性能优化的同时，降低标注成本和节省时间。这种策略需要兼顾时间效率和性能输出，以达到既有效率又具有高性能的学习过程。

对于神经网络模型的训练，其超参数设置则直接影响训练过程和模型性能。在早期的研究中，已有研究者对各种超参数进行了详细的研究和确定。在这里，我们参考了这些研究，并选择了适合神经网络训练的超参数，即使用8为批处理大小，以及0.00005的学习率。



在制定主动学习实验的中止策略时，需要关注模型性能是否已经达到饱和。对于这个实验，主要关注的是模型在验证集上的平均F1分数，即我们希望观察基于随机采样策略的模型在何时能够超过基于全数据训练的模型的性能，或者在何时性能增长率开始变缓。图表反映了基于随机采样策略和基于全数据训练策略两种方式下的模型性能对比。在第8次迭代时，基于随机采样的模型的性能已经超过了基于全数据训练的模型。然而，这个性能提升并不稳定，因此选择继续迭代。在第9次迭代后，模型的性能增长已经开始变得缓慢。考虑到进一步迭代将增加时间成本和标注成本，决定在第十次迭代时终止实验。这个策略允许以最小的成本获取最优的性能。

4.5运行模块

在本研究中，通过执行一系列主动学习策略对文本数据进行分类。实验通过PyCharm连接远程环境进行，且在每次迭代中添加一个解释，总共运行9次迭代，共耗时36小时。使用了MC dropout机制的耗时会接近37小时。

随机采样：通过“Whole\_train.py”脚本实现，对全数据模式和主动学习模式进行处理。

多样性采样：基于“diversity\_sampling\_whole.py”脚本，此方法有三种不同的解释标注方式，分别是预设解释、人工输入以及使用OpenAI模型生成的解释。（openai需要接口密钥）

不确定性采样：利用“uncertaintyWhole.py”脚本，此策略的目的是选取对模型分类最具不确定性或挑战性的数据点。

基于BALD的Dropout NN采样：通过“BALD\_MCD.py”脚本实施，它结合了Dropout神经网络与Bayesian Active Learning by Disagreement (BALD) 策略，通过最小置信度专门寻找模型预测中的高不确定性数据点。

实验运行的研究主旨在于对各种策略在文本分类任务中的表现进行深入评估，以确定哪种策略最适于提高模型的学习效率。

5 评估

5.1 期望以及评估结构

首先，对于人在循环中的框架有以下几个预设的期望。值得注意的是全数据集是已标注数据集（有９个默认解释连接），训练集占比接近一的数据集划分。

1：主动学习通过增加已标注数据集可以有效提高模型性能（在测试集以及验证集上）

2：增加标注了解释的未标注数据集，比只增加标签标注的未标注数据集影响更大

3：多数策略可以超过基线（随机抽样）的同时，中止策略之前超过或接近全数据量下的性能

4：可以选择出最佳的主动学习策略，解释标注策略以及模型架构策略来实现小数据高性能

5：提供的解释的数量以及质量会极大影响模型性能

基于这些期望来设置评估框架，这个框架将应用在测试集以及验证集上：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 评估ID | 模型结构 | 数据集 | 抽样策略 | 解释方式 | 性能指标 |
| 1 | None dropout | 全数据集 | None | 不标注解释 | Average F1 score, average accuracy, F1 score(label 0-8) |
| 2 | None dropout | 主动学习数据集 | Random sampling | 不标注解释 | Average F1 score, average accuracy, F1 score(label 0-8) |
| 3 | None dropout | 主动学习数据集 | Random sampling | 标注解释-非openai | Average F1 score, average accuracy, F1 score(label 0-8) |
| 4 | None dropout | 主动学习数据集 | Uncertainty sampling | 标注解释-非openai | Average F1 score, average accuracy, F1 score(label 0-8) |
| 5 | None dropout | 主动学习数据集 | Diversity sampling | 标注解释-非openai | Average F1 score, average accuracy, F1 score(label 0-8) |
| 6 | None dropout | 主动学习数据集 | Diversity sampling | Openai 标注解释 | Average F1 score, average accuracy, F1 score(label 0-8) |
| 7 | Dropout | 主动学习数据集 | BALD sampling | 标注解释-非openai | Average F1 score, average accuracy, F1 score(label 0-8) |
| 8 | None dropout | Active learning datasets | Diversity sampling | Two exp in each iteration | Average F1 score, average accuracy, F1 score(label 0-8) |

5.2 期望一：

根据5.1节的期望来进行分点评估。对于第一点期望，需要观察主动学习通过增加已标注数据集是否可以有效提高模型性能。因此需要观察在4.3节中提到的学习曲线。期望在学习曲线中，模型性能是随着已标注实例的数量上升而不断的上升的。

为减少运算时间，使用具有代表性的评估方式，因此选择了5.1节评估框架中评估ID为2的评估方式，使用随机抽样策略。每轮只标注分类对应的标签，不再新增解释，而是用默认的9个解释。评估结果如下图所示。实线部分为校验集和测试集的平均F1 score，虚线部分为平均准确率。通过设置线性趋势线可以发现，无论是在验证集还是测试集上，学习曲线的趋势符合第一个期望。因此，此人在循环中的主动学习框架确实有效地提升了模型的性能。

5.3 期望二

5.1节中期望二探究的是人在循环中的框架在基于ExpBERT上的有效性，探究在标注过程中除了增加标签以外，增加解释是否可以使性能变得更高。因此，使用5.1节表格中评估ID对应为２以及对应为３的评估方式来进行对比实验。查看是否ID为3（增加解释）的评估结果比ID为２（只增加标签）的评估结果更优。在ID为３的评估过程中，解释量每轮增加１。这里评估结果使用校验集以及测试集对应的F1 平均分数以及准确率平均分数来评估，截取最后一次迭代的数据展示到下面表格中。

|  |  |  |
| --- | --- | --- |
|  | F1 average | Accuracy average |
| Val ID 2 | 0.6159 | 0.599 |
| Val ID 3 | **0.625** | **0.6131** |
| Test ID 2 | 0.4489 | 0.5761 |
| Test ID 3 | **0.5056** | **0.6216** |

表格中可以看出ID为3（增加了解释并标注了标签）的所有评估结果都比ID为2（只标注了标签）的评估结果要高。因此，不仅标注了标签，还增加了解释的标注可以有效地提高模型的性能。最后的结论是期望二得到了满足。

5.4 期望三

期望三主要查看主动学习策略的有效性，以及是否可以超过基线（随机采样）的性能以及是否可以使用少量数据，在中止策略之前超过或接近全数据量下的性能。因此，采用了不同策略在主动学习最后一次迭代过程时对应的性能，即使用5.1节表格中ID为1，3，4，5和7的评估方式。对比基线以及所有主动学习策略以及全数据下的性能。使用F1平均分数以及准确率平均分数来评估。评估结果如下图所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Full datasets ID1 | Random sampling(baseline) ID3 | Uncertainty sampling ID4 | Diversity sampling ID5 | BALD ID7 |
| test\_avg\_ac | 0.6248 | 0.6216 | **0.6256** | **0.633** | 0.6215 |
| test\_avg\_f1 | 0.5363 | 0.5056 | **0.51** | **0.5466** | **0.5329** |
| val\_avg\_ac | 0.6602 | 0.6131 | 0.6026 | **0.6348** | **0.6275** |
| val\_avg\_f1 | 0.5011 | 0.486 | **0.4981** | **0.5522** | **0.5286** |

结果：从表中可以发现不确定性抽样以及多样性抽样在测试集上性能超过了基线，在验证集上多样性抽样以及BALD策略实施后的性能超过了基线。尤其是多样性抽样，在不同数据集上根据不同性能指标的制定下，性能都超过了基线。除此以外，不确定性抽样以及多样性抽样对应的的测试集的平均准确率超过了全数据下的准确率。更甚的是，多样性抽样对应的三种指标都超过了全数据下的性能指标。BALD策略下在验证集上的平均F1分数也超过了全数据下的分数。

原因：产生这样结果的原因在于相比于基线来说，其他策略是具有偏见的，不确定性抽样帮助模型学习最不确定数据进而克服模型在这类数据上的弱点。多样性抽样则是选择语义上更为多样的数据，让模型更全面的进行训练。BALD则是通过多次的采样来找出不确定性最高的数据，同样起到了提升性能的作用。因此，对于无偏见的随机采样来说，剩下三种主动学习策略可以根据不同的角度，有偏见的帮助模型训练。

因此，根据大多数主动学习策略超过了基线甚至部分策略在不同指标下超过了全数据的性能的发现，可以认为期望3是被满足的，因此，选择更先进的主动学习策略可以更进一步的提升模型的性能，甚至在少部分数据训练时即可超过全数据下的性能。

5.5 期望四：

期望四主要需要选择对于应急反应系统来说最适合最能提升性能的方案，需要选择并对比不同的抽样策略，以及对不同策略产生的结果来分析产生这样的结果的原因，是否是偶然现象。首先，为了排除偶然情况，将绘制这几种主动学习策略对应的折线图（学习曲线），查看每个阶段的情况。测试指标与5.4节的测试指标相同，但是不止观察最后一次迭代的情况，而是观察每一次迭代的情况。

结果：

分别绘制了四个折线图，如下图所示，分别代表了不同策略下的测试集上的和验证集上的平均F1分数以及平均准确率。通过折线图可以观察到居于上方直线是全数据量的性能。每次迭代更加靠近甚至超过这条直线的是多样性抽样策略以及BALD策略对应的学习曲线。而不确定性策略在四种指标上与基线（随机抽样）在第9以及第10次迭代之前的分数呈现出不确定性抽样策略低于基线的现象。但是在最终迭代时，不确定性抽样策略在测试集上超过了基线。

除此以外，需要观察对应于不同标签下，这些主动学习策略的表现，查看对于大多数标签来说更合适的学习策略。采用对后一次迭代的结果对应到测试集上的F1分数来评估。最终的评估结果如下图所示，多样性采样方式对在多数分类中的F1分数较高，尤其是分类2，3，4和7。不确定性采样在标签5对应的分数上更高，BALD策略则在分类１上表现较好。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class label  test | exp\_rs\_add\_1 | uncertainty\_add\_1 | diversity\_sampling | MCD\_BALD\_sampling\_20epoch |
| 0 | 0.8698 | 0.8626 | 0.8696 | 0.8622 |
| 1 | 0.2127 | 0.3128 | 0.3103 | 0.3492 |
| 2 | 0.5421 | 0.5412 | 0.6392 | 0.5895 |
| 3 | 0.2803 | 0.3079 | 0.3886 | 0.355 |
| 4 | 0.6398 | 0.6029 | 0.6636 | 0.6058 |
| 5 | 0.06 | 0.2197 | 0.1308 | 0.1081 |
| 6 | 0.6894 | 0.6603 | 0.6809 | 0.6627 |
| 7 | 0.6309 | 0.6312 | 0.6395 | 0.635 |
| 8 | 0.625 | 0.615 | 0.5973 | 0.584 |

总而言之，综合表现来看基于语义的多样性抽样在次实验中更适合用在主动学习的循环中去提高模型的学习率以及最终的性能。使用基于语义的不确定性抽样也可以轻松的在少量循环后超过全数据的性能。

原因：

在4.2.1节中提到过数据集中的标签对应的数据量的分布极度不平衡。因此，在数据量极度不平衡的情景中，基于语义的多样性抽样策略表现最好，其背后的原因是这种策略选择的样本覆盖了多种类别的具有多样性以及代表性信息，从而为模型提供了更为全面的视角。这对于不平衡数据特别重要，因为某些较为稀少的类别可能会被忽略或者欠采样。

其次，结合BALD与least confidence的策略也能取得不错的效果，因为BALD注重从多次采样中挑选出不确定性最高的数据，与least confidence策略共同找出模型最不确定的数据点。然而，纯粹使用least confidence不确定性抽样在这种情况下可能无法达到模型的最佳效果，因为此算法主要关注模型最不确定的数据点，但在极度不平衡的数据中，这可能导致模型持续采样占主导地位的类别，而忽视了较为稀有的类别。

5.6 期望五

在5.4节中得出了基于语义的多样性抽样最可以提高模型学习率的结论。因此进一步探索基于此策略，在标注过程中改变解释的质量以及数量对整体性能的影响。因此，选择了在5.1节中ID为5以及6来进行质量上的对比。ID 6相比与ID5同样使用了多样性抽样方式。不同的是6对应的评估方式中标注部分使用了3.5.2节中提到的OpenAI模型来生成解释。值得注意的是，OpenAI模型生成的解释只是接近合理解释，但是相比于对应ID为5中使用预设解释来说还是有一定的瑕疵。在OpenAI模型生成解释时由于’prompt’的字数有限制所以不能让此模型充分的解析所有的文本。因此，生成的解释中往往含有很多噪音。比如会出现包括地点，人物等与分类描述不相关的词语。

质量影响：

所以，OpenAI模型生成解释的过程可以看作是较低质量解释的标注过程。因此可以对比这两个评估方式的区别来查看解释的质量带来的影响。通过下图可以发现，主动学习最后一次迭代时，ID为5的评估方式在所有测试集以及性能指标下均比ID为6的评估方式高约2%。即使模型很好的体现了鲁棒性（性能差距较小），但是解释的质量好坏依旧引起了性能差距。

|  |  |  |
| --- | --- | --- |
|  | F1 average | Accuracy average |
| Val ID 5 | **0.5522** | **0.6348** |
| Val ID 6 | 0.538 | 0.6294 |
| Test ID 5 | **0.5466** | **0.633** |
| Test ID 6 | 0.5353 | 0.6144 |

所以，使用高质量的解释在主动学习循环的注释者注释环节是可以有效提高性能的。但是高质量解释可能会考虑到人力成本，如果希望节省人力成本为中心，性能上与高质量解释带来的性能误差范围在2%左右，OpenAI生成解释的方式则值得考虑。

数量影响：

假设每轮用于标注的新增解释量增加到2，每轮依旧提供6%的数据用来标注。因此，期望的效果是用更少的迭代次数即可超过基线并达到全数据下的性能。但是，每一轮ExpBERT模型需要训练的成本和时间也相应的增长了。因此也要观察当达到期望性能时，所耗时间是否等于或者少于每次迭代新增解释数量为1的情况下的时间。

左侧图片可以发现，在第六次迭代时增加了解释的测试集的准确率已经达到甚至超过了原有解释标注的准确率，此时主动学习框架的运行时长仅为原有方式的二分之一的时间即仅用了18h左右。如果应急响应系统的需求是只要求高准确率，可以根据需求调整中止策略，如果继续运行最终的准确率可达到0.65。同样的，右侧表示测试集的F1分数也呈现了在第六次和第七次时性能赶超的情况。但是，根据中止策略，出现性能赶超情况的同时中断循环可以减少应急系统反应时间。

所以，每次迭代多增加一个解释可以既缩短系统的反应时间，又提高了准确率。由于此方式用了更少次的迭代（根据中止策略可以在第六次迭代后中止），所以增加的数据也变少了，因此可以节省注释者的消耗。这个方式是一个巧妙地平衡了运行时间，人力以及性能的方式。

5.7 总结

经过评估，主动学习可以有效的提高模型的性能，减少人力以及标注成本。同时，标注过程中除了增加标签以外还增加了解释的标注后的模型性能有显著的提升。通过结合不同的高级主动学习策略可以有效地超过基线（随机采样）甚至是在有限次迭代以较少的训练集超过全数据量下的性能。性能最突出的方式是采用基于语义的多样性采样，结合高质量的解释支持。在标注成本以及预训练时间消耗上进行控制的前提下，适当少量增加每轮迭代的解释数量的这种方式可以达到模型的最佳性能。

结论（3页）

本论文的目标是探究人在循环中框架对基于ExpBERT模型的文本分类器应用于灾难应急情况的有效性。简单来说，是寻找减少人工标注成本以及运算时间上的同时提升性能的最佳方案。项目结合了生命周期，首先对框架以及整个主动学习流程进行设计，其次搭建兼容性大的实验环境并提供中止策略以及数据，最后评估了影响系统的关键模块，验证主动学习以及解释标注的有效性，评估了不同的主动学习策略以及解释标注方式带来的影响，以及衡量解释的质量和数量的平衡。在本章中，将对整体项目进行验证与总结。本章还将提到此框架带来的局限性，以及在未来需要做的工作。

项目总结

项目的主要框架是人在循环中结合文本分类功能提升文本分类任务性能。采用不同的主动学习策略以及不同的标注方式以及模型结构来提高模型性能。最终项目的实现完成了1.3.1的所有目标。这些目标生成了几项结果。1.3.1节的目标如下：

•开发一个基于池的人在循环中框架，并探索其有效性。

•构建注释器仿真流程，设计循环中止标准（例如达到全数据下模型性能后再运行一次）。

•生成基线（无主动学习下全数据的模型性能或无标注的随机抽样），研究主动学习是否可以提高文本分类模型的性能。比较不同查询算法抽样的影响。

•探索了解释质量以及数量对本系统的影响。例如，提供具有噪声的解释们或者引入随机性来代替精确的解释。

目标一的框架实现在第三章进行了详细的设计，其有效性在5.2以及5.3中得到了验证。基于池的主动学习框架无论是在验证集还是测试集上，学习曲线的趋势不断增加。此人在循环中的主动学习框架有效地提升了模型的性能。

对于第二个目标，通过3.5节的三种不同解释标注方式以及4.4节的中止迭代参数的设置得以实现。在不同标注解释方式中，5.6节中发现预先设置解释或者人工标注这类高质量解释所带来的性能比OpenAI模型生成解释的方式要高2%。2%的小差距也表明，OpenAI模型生成解释也是一种在人力标注成本较大时的可替代方式。

第三个目标的实现对性能有着重要的影响。通过3.4节设计了性能的下限以及不同的带有偏见的主动学习策略。发现使用基于语义的多样性抽样的模型的平均表现更好（5.5节），高于带有dropout机制的BALD抽样方式以及不确定性抽样方式。正如理由中的分析：在数据量极度不平衡的情景中，其表现突出背后的原因是这种策略选择的样本覆盖了多种类别的具有多样性以及代表性信息，从而为模型提供了更为全面的视角。

第四个目标用于对标注的解释变量进行分析，如5.6节的评估中发现应用多样性采样后，通过每轮迭代使用多于数量一的高质量的解释可以极大提高模型性能的同时，减少了反应时间。

最后整体项目完成了整个生命周期的搭建，其中包括设计（第3章），实现（第4章）以及第5章中的评估。可在基于池的主动学习框架中结合基于语义的多样性抽样以及高质量的解释标注来更有效提高模型性能。注释过程结合OpenAI模型可以在人力标注成本较大时作为替代方式。

局限

首先，值得注意的是人在循环中的框架对模型性能的提升存在固有的局限性。具体来说，在标注过程中，如果用户的主要关注点是降低人力成本，那么系统可能更偏向于使用OpenAI模型生成的解释。然而，由于OpenAI的prompt字数限制为4097，这限制了每轮能够进行语义分析的样本数量。因此，OpenAI生成的解释可能不足以全面地考虑所有相关信息，从而可能限制模型的性能。当考虑到人工标注时，存在误标的风险。尽管我们在5.6节中讨论了模型对噪声的鲁棒性，但过多的误标仍可能对模型性能产生显著影响。

进一步地，项目的迭代终止参数是基于CrisisNLP数据集来确定的。对于不同的任务或数据集，这些参数可能需要调整。由于这不是一个自动化的调整过程，因此这一点也限制了项目的应用灵活性。此外，虽然在数据严重不平衡的情况下，基于语义的多样性抽样策略表现优异，但在数据平衡度发生变化时，其优势可能会被削弱。最后，我们必须考虑到模型更新所需的时间和计算资源。在有限的资源环境中，高昂的计算成本可能限制了项目的可行性。

未来工作

随着本项目逐步完善，存在多个潜在的进一步研究领域。针对这些领域，本节提出以下具有研究价值的工作方向：

应对数据分布的不平衡：在本研究中，5.5节发现基于语义多样性的主动学习策略在数据严重不平衡的场景下具有优越性能。然而，鉴于现实场景下的数据分布可能会经常变化，未来研究将探讨结合其他策略，或与基于不确定性的策略相结合，以适应变化的数据分布并提高模型的鲁棒性。

优化解释生成：当前，解释的生成考虑了OpenAI模型和人工标注的能力。尽管如此，其有效性依赖于反复的实验验证。为了更加系统地评估和优化解释，我们计划采用对抗性测试和可信度评估[文献]。通过提出一系列的目标问题或误导性问题来揭示解释生成过程的潜在弱点，并据结果进行评估。

框架的广泛应用：虽然本项目主要集中在应用主动学习于基于ExpBERT的文本分类任务，但考虑到该策略的有效性，未来研究可以考虑将此框架应用于更广泛的自然语言处理（NLP）任务中，如命名实体识别（NER）和问答（QA）系统，进一步挖掘其在不同场景下的价值。

实际环境下的性能测试：虽然第5章的评估方法是基于特定实验环境，但为了深入了解模型在实际场景中的适用性，预期设计在线系统和用户界面，进行直观的模型性能评估。这不仅有助于全面了解模型在真实环境中的性能，还在大数据环境中的应用提供了前景。

1. McCreadie, R., Buntain, C., & Soboroff, I. (2019). TREC incident streams: Finding actionable information on social media.

[2] Murty, S., Koh, P. W., & Liang, P. (2020). Expbert: Representation engineering with natural language explanations. arXiv preprint arXiv:2005.01932.

[3] Baram, Y., Yaniv, R.E., Luz, K.: Online choice of active learning algorithms. J. Mach. Learn. Res. 5(Mar), 255–291 (2004)

[4] Prabhu, S., Mohamed, M., & Misra, H. (2021). Multi-class text classification using bert-based active learning. arXiv preprint arXiv:2104.14289.

[5] A. Prest, C. Schmid, and V. Ferrari, “Weakly supervised learning of interactions between humans and objects,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 3, pp. 601–614, 2012.

[6] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Muñoz-Marí, “A survey of active learning algorithms for supervised remote sensing image classification,” IEEE Journal on Selected Topics in Signal Processing, vol. 5, no. 3, pp. 606–617, 2011.

[7] B. Settles, “Active learning literature survey,” University of Wisconsin, Madison, vol. 52, pp. 55–66, 2010.

[8] Y. Fu, X. Zhu, and B. Li, “A survey on instance selection for active learning,” Knowledge and Information Systems, vol. 35, no. 2, pp. 249–283, 2013.

[9] Schröder, C., & Niekler, A. (2020). A survey of active learning for text classification using deep neural networks. arXiv preprint arXiv:2008.07267.

[10] Ren, P., Xiao, Y., Chang, X., Huang, P. Y., Li, Z., Gupta, B. B., ... & Wang, X. (2021). A survey of deep active learning. ACM computing surveys (CSUR), 54(9), 1-40.

[new 1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT.

[11] Tomaszewski, J. E. (2021). Overview of the role of artificial intelligence in pathology: the computer as a pathology digital assistant. In Artificial intelligence and deep learning in pathology (pp. 237-262). Elsevier.

[12] Godbole, S., Harpale, A., Sarawagi, S., & Chakrabarti, S. (2004). Document classification through interactive supervision of document and term labels. In Knowledge Discovery in Databases: PKDD 2004: 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Pisa, Italy, September 20-24, 2004. Proceedings 8 (pp. 185-196). Springer Berlin Heidelberg.

[13] Munro R (2020) Human-in-the-loop machine learning. Manning Publications, Shelter Island.

[14] Mosqueira-Rey, E., Hernández-Pereira, E., Alonso-Ríos, D. et al. Human-in-the-loop machine learning: a state of the art. Artif Intell Rev 56, 3005–3054 (2023). <https://doi.org/10.1007/s10462-022-10246-w>

[15] Settles B (2009) Active learning literature survey. Tech. rep., University of Wisconsin-Madison. Department of Computer Sciences, <https://minds.wisconsin.edu/handle/1793/60660>

[16] Amershi S, Cakmak M, Knox WB et  al (2014) Power to the people: the role of humans in interactive machine learning. AI Magazine 35(4):105–120. <https://doi.org/10.1609/aimag.v35i4.2513>

[17] Dudley JJ, Kristensson PO (2018) A review of user interface design for interactive machine learning. ACM Trans Interact Intell Syst. <https://doi.org/10.1145/3185517>

[18] Michael CJ, Acklin D, Scheuerman J (2020) On interactive machine learning and the potential of cognitive feedback. arXiv e-prints arxiv:2003.10365 [cs.HC]

[19] Ramos G, Meek C, Simard P et  al (2020) Interactive machine teaching: a human-centered approach to building machine-learned models. Hum Comput Interact 35(5–6):413–451. <https://doi.org/10.1080/07370024.2020.1734931>

[20] Devidze R, Mansouri F, Haug L et al (2020) Understanding the power and limitations of teaching with imperfect knowledge. In: Bessiere C (ed) Proceedings of the twenty-ninth international joint conference on artifcial intelligence, IJCAI-20. International Joint Conferences on Artifcial Intelligence Organization, 2647–2654, <https://doi.org/10.24963/ijcai.2020/367>

[21] Burr Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison. 2009.

[22] Dana Angluin. 1988. Queries and Concept Learning. Machine Learning 2, 4 (1988), 319–342.

[23] Ross D King, Kenneth E Whelan, Ffion M Jones, Philip G K Reiser, Christopher H Bryant, Stephen Muggleton, Douglas B Kell, and Stephen G Oliver. 2004. Functional genomic hypothesis generation and experimentation by a robot scientist. Nature 427, 6971 (2004), 247–252.

[24] Ido Dagan and Sean P. Engelson. 1995. Committee-Based Sampling For Training Probabilistic Classifiers. In Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995. Morgan Kaufmann, 150–157.

[25] David D Lewis and William A Gale. 1994. A sequential algorithm for training text classifiers. (1994), 3–12.

[26] B.Baum and K. Lang. Query learning can work poorly when a human oracle is used. In Proceedings of the IEEE International Joint Conference on Neural Networks, 1992.

[27] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, (2014).

[28] Dagan and S. Engelson. Committee-based sampling for training probabilistic classifiers. In Proceedings of the International Conference on Machine Learning (ICML), pages 150–157. Morgan Kaufmann, 1995.

[29] Ozan Sener and Silvio Savarese. “Active Learning for Convolutional Neural Networks: A Core-Set Approach”. In: 6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings. 2018.

[30] Lewis and W. Gale. A sequential algorithm for training text classifiers. In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, pages 3–12. ACM/Springer, 1994.

[31] Rong Hu, Brian Mac Namee, and Sarah Jane Delany. “Active Learning for Text Classification with Reusability”. In: Expert Systems with Applications 45.C (2016), pp. 438–449.

Diversity sampling:

1. Peng, H., Guo, S., Zhao, D., Wu, Y., Han, J., Wang, Z., ... & Zhong, M. (2023). Query-efficient model extraction for text classification model in a hard label setting. Journal of King Saud University-Computer and Information Sciences, 35(4), 10-20.
2. Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.
3. Sener, O., Savarese, S., 2017. Active learning for convolutional neural networks: A core-set approach. arXiv preprint arXiv:1708.00489.
4. Houlsby, N., Huszár, F., Ghahramani, Z., & Lengyel, M. (2011). Bayesian active learning for classification and preference learning. arXiv preprint arXiv:1112.5745.
5. Gal, Y., Islam, R., & Ghahramani, Z. (2017, July). Deep bayesian active learning with image data. In International conference on machine learning (pp. 1183-1192). PMLR.
6. Myojin, T., Hashimoto, S., & Ishihama, N. (2020). Detecting uncertain BNN outputs on FPGA using Monte Carlo dropout sampling. In Artificial Neural Networks and Machine Learning–ICANN 2020: 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15–18, 2020, Proceedings, Part II 29 (pp. 27-38). Springer International Publishing.
7. Tsymbalov, E., Panov, M., & Shapeev, A. (2018). Dropout-based active learning for regression. In Analysis of Images, Social Networks and Texts: 7th International Conference, AIST 2018, Moscow, Russia, July 5–7, 2018, Revised Selected Papers 7 (pp. 247-258). Springer International Publishing.
8. C. Schr¨oder, L. M¨uller, A. Niekler, and M. Potthast, “Small-text: Active learning for text classification in python,” arXiv preprint arXiv:2107.10314, 2021.
9. T. Danka and P. Horvath, “modal: A modular active learning framework for python,” arXiv preprint arXiv:1805.00979, 2018.
10. A. Azaria, “Chatgpt usage and limitations,” 2022.
11. Y. Shi, H. Ma, W. Zhong, G. Mai, X. Li, T. Liu, and J. Huang, “Chatgraph: Interpretable text classification by converting chatgpt knowledge to graphs,” arXiv preprint arXiv:2305.03513, 2023.
12. Mullachery, V., Khera, A., & Husain, A. (2018). Bayesian neural networks. arXiv preprint arXiv:1801.07710.
13. Goan, E., & Fookes, C. (2020). Bayesian neural networks: An introduction and survey. Case Studies in Applied Bayesian Data Science: CIRM Jean-Morlet Chair, Fall 2018, 45-87.
14. Neal, R. M. (2012). Bayesian learning for neural networks (Vol. 118). Springer Science & Business Media.

第三章dropout：Improving neural networks by preventing co-adaptation of feature detectors

第三章dropout那里：Dropout: A Simple Way to Prevent Neural Networks from Overfitting

1. BALD那里：Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. ICML.

第三章解释标注那里：Zhao, X., Li, M., Weber, C., Hafez, M. B., & Wermter, S. (2023). Chat with the environment: Interactive multimodal perception using large language models. arXiv preprint arXiv:2303.08268.

1. 实验环境那里 [Pedregosa et al., 2011, Scikit-learn: Machine Learning in Python]
2. 实验环境pytorch那里：Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Desmaison, A. (2019). Pytorch: An imperative style, high-performance deep learning library. In Advances in neural information processing systems (pp. 8026-8037).

最后一张未来工作：Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?” Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).