软件学报 ISSN 1000-9825, CODEN RUXUEW
2024,35(12):5470−5486 [doi: 10.13328/j.cnki.jos.007057] [CSTR: 32375.14.jos.007057]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

针对低资源场景下连续情感分析任务的持续注意力建模*

张 涵, 王晶晶, 罗佳敏, 周国栋

(苏州大学 计算机科学与技术学院, 江苏 苏州 215006)

通信作者: 王晶晶, E-mail: djingwang@suda.edu.cn

摘 要: 目前情感分析的研究普遍基于大数据驱动型模型, 严重依赖高昂的标注成本和算力成本, 因此针对低资源场景下的情感分析研究显得尤为迫切. 然而, 存在的低资源场景下的情感分析研究主要集中在单个任务上, 这导致模型难以获取外部任务知识. 因此构建低资源场景下的连续情感分析任务, 旨在利用持续学习方法, 让模型随时间步学习多个情感分析任务. 这样可以充分利用不同任务的数据, 并学习不同任务的情感信息, 从而缓解单个任务训练数据匮乏问题. 认为低资源场景下的连续情感分析任务面临两大核心问题, 一方面是单个任务的情感信息保留问题, 另一方面是不同任务间的情感信息融合问题. 为了解决上述两大问题, 提出针对低资源场景下连续情感分析任务的持续注意力建模方法. 所提方法首先构建情感掩码 Adapter, 用于为不同任务生成硬注意力情感掩码, 这可以保留不同任务的情感信息, 从而缓解灾难性遗忘问题. 其次, 所提方法构建动态情感注意力, 根据当前时间步和任务相似度动态融合不同 Adapter 抽取的特征, 这可以融合不同任务间的情感信息. 在多个数据集上的实验结果表明: 所提方法的性能显著超过了目前最先进的基准方法. 此外, 实验分析表明, 所提方法较其他基准方法具有最优的情感信息能力和情感信息融合能力, 并且能同时保持较高的运行效率.

关键词: 情感分析; 低资源场景; 持续学习; Adapter; 注意力机制

中图法分类号: TP18

中文引用格式: 张涵, 王晶晶, 罗佳敏, 周国栋. 针对低资源场景下连续情感分析任务的持续注意力建模. 软件学报, 2024, 35(12): 5470–5486. http://www.jos.org.cn/1000-9825/7057.htm

英文引用格式: Zhang H, Wang JJ, Luo JM, Zhou GD. Continual Attention Modeling for Successive Sentiment Analysis in Low-resource Scenarios. Ruan Jian Xue Bao/Journal of Software, 2024, 35(12): 5470–5486 (in Chinese). http://www.jos.org.cn/1000-9825/7057.htm

Continual Attention Modeling for Successive Sentiment Analysis in Low-resource Scenarios

ZHANG Han, WANG Jing-Jing, LUO Jia-Min, ZHOU Guo-Dong

(School of Computer Science and Technology, Soochow University, Suzhou 215006, China)

Abstract: Currently, sentiment analysis research is generally based on big data-driven models, which heavily rely on expensive annotation and computational costs. Therefore, research on sentiment analysis in low-resource scenarios is particularly urgent. However, existing research on sentiment analysis in low-resource scenarios mainly focuses on a single task, making it difficult for models to acquire external task knowledge. Therefore, this study constructs successive sentiment analysis in low-resource scenarios, aiming to allow models to learn multiple sentiment analysis tasks over time by continual learning methods. This can make full use of data from different tasks and learn sentiment information from different tasks, thus alleviating the problem of insufficient training data for a single task. There are two core problems with successive sentiment analysis in low-resource scenarios. One is preserving sentiment information for a single task, and the other is fusing sentiment information between different tasks. To solve these two problems, this study proposes continual attention modeling for successive sentiment analysis in low-resource scenarios. Sentiment masked Adapter (SMA) is first constructed, which is used

* 基金项目: 国家自然科学基金 (62006166, 62076175, 62076176); 江苏高校优势学科建设工程
收稿时间: 2023-03-31; 修改时间: 2023-08-20; 采用时间: 2023-09-05; jos 在线出版时间: 2024-01-03
CNKI 网络首发时间: 2024-01-04

to generate hard attention emotion masks for different tasks. This can preserve sentiment information for different tasks and mitigate catastrophic forgetting. Secondly, dynamic sentiment attention (DSA) is proposed, which dynamically fuses features extracted by different Adapters based on the current time step and task similarity. This can fuse sentiment information between different tasks. Experimental results on multiple datasets show that the proposed approach significantly outperforms the state-of-the-art benchmark approaches. Additionally, experimental analysis indicates that the proposed approach has the best sentiment information retention ability and sentiment information fusion ability compared to other benchmark approaches while maintaining high operational efficiency.

Key words: sentiment analysis; low-resource scenario; continual learning; Adapter; attention mechanism

近年来, 随着大数据驱动型深度神经网络快速发展, 其在计算机视觉 (computer vision, CV)、自然语言处理 (natural language processing, NLP) 等领域取得了显著的成果, 其中深度神经网络在各种高资源场景任务下的性能提升尤为显著. 但是大数据驱动型深度神经网络的训练过程需要大量数据, 这意味着高昂的标注成本和算力成本. 此外, 现实中充斥着各种低资源场景下的任务, 如跨语言通信、少数民族或方言翻译等. 因此越来越多的研究人员开始关注低资源场景下的任务, 即要求模型能够利用受限的训练数据量解决问题. 相关研究普遍借助数据增强或迁移学习等方法得到更多数据资源, 以克服标记数据的缺乏[1], 从而提高低资源场景下的任务性能. 在情感分析领域同样存在大量低资源场景下的任务, 这是因为情感分析领域中的标注数据往往集中于商品评论或社交媒体, 缺乏专业领域的标注数据. 此外, 真实用户的情感分析数据具有一定的隐私性, 有时难以获取[2]. 上述问题限制了现有的情感分析技术在低资源场景下的应用和推广. 目前针对低资源场景下情感分析研究主要集中于单个情感分析任务, 这导致模型无法利用其他情感任务的情感信息.

与以往研究不同的是, 本文构建了低资源场景下的连续情感分析任务, 旨在利用持续学习方法, 在多个情感分析任务上训练模型. 如图 1 所示, 随着时间步变化, 模型能够按顺序学习不同的情感分析任务. 具体而言, 模型在不同时间步学习不同的情感分析任务, 每个情感分析任务具有不同的分布. 其中 T_{N-1} 表示 N–1 个时间步, D^N 表示第 N 个分布, $\{x_k^N, y_k^N\}$ 表示任务 N 的训练数据, x_t 表示测试数据, 随着任务数量增多, x_t 所属的分布范围逐渐扩大. 连续情感分析任务既可以只包含一种情感分析任务类型 (如句子级情感分类), 也可以包含不同的情感分析任务类型 (如句子级情感分类和属性级情感分类). 在训练阶段, 模型仅使用当前任务的训练数据, 而在测试阶段, 模型则要预测所有已学任务的测试数据. 在该任务形式下训练得到的模型可以随时间步持续更新, 且可以捕捉多个情感分析任务的情感信息, 这可以有效缓解低资源场景下单个任务的训练数据匮乏问题.

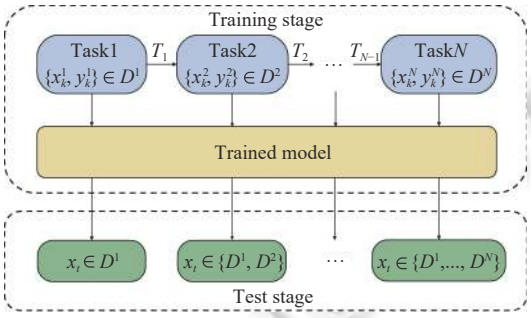

图 1 针对连续情感分析任务的持续学习过程

本文认为低资源场景下的连续情感分析任务存在两大核心问题. 一方面, 本文基于持续学习思想在多个任务的训练集上通过梯度更新的方法训练模型, 由于不同情感分析任务的数据分布不同, 这种增量更新可能会导致灾难性遗忘 (catastrophic forgetting, CF)[3,4]. 即引入新的任务之后, 由于参数更新对模型引起的干扰, 当前模型在学习新任务的过程中忘记了如何解决旧任务, 具体表现为模型在旧任务上的性能大幅下降. 另一方面, 本文旨在利用多个任务的情感信息缓解低资源下单个任务训练数据匮乏问题. 然而, 目前关于连续情感分析任务的持续学习方法普遍关注于缓解灾难性遗忘问题, 缺少关于情感信息融合的研究, 难以找到一个高效的模型来融合不同情感分析任务间的情感信息.

为了解决连续情感分析任务在低资源场景下面临的两大核心问题, 本文提出了针对低资源场景下连续情感分析任务的持续注意力建模方法 (continual attention modeling for successive sentiment analysis in low-resource scenarios, CAM). 在多个情感分析任务的学习过程中, CAM 既可以保留不同任务的情感信息, 缓解灾难性遗忘问题, 也可以高效地捕捉不同任务的情感信息并进行融合. CAM 主要由两个部分组成: 用于保留单个任务情感信息的情感掩码 Adapter (sentiment masked Adapter, SMA) 和用于融合不同任务情感信息的动态情感注意力 (dynamic sentiment attention, DSA).

首先, SMA 使用 Adapter 抽取数据特征, 然后通过生成任务相关的情感掩码控制神经元的梯度传播, 从而保留 Adapter 中的重要参数, 这可以有效保留不同任务的情感信息, 缓解灾难性遗忘问题. SMA 基于 Adapter 结构, 不仅可以有效捕获情感信息, 还可以较少模型参数, 提高运行效率. 其次, DSA 将 SMA 抽取到的数据特征通过注意力机制融合, 这可以进一步融合不同任务间的情感信息, 从而缓解单个任务训练数据匮乏问题. 此外, DSA 会根据时间步变化和任务相似度动态融合不同 Adapter 抽取到的数据特征, 这与模型的持续学习训练过程相符合.

在多个数据集上的实验结果与分析表明: CAM 的性能显著超过了目前最先进的基准方法 CTR, 具有最优的情感信息保留能力、情感信息融合能力, 并且能同时保持较高的运行效率.

综上所述, 本文的主要贡献有以下 3 点.

(1) 本文构建了低资源场景下的连续情感分析任务, 旨在利用持续学习方法, 让模型随时间步学习多个情感分析任务. 这种任务形式一方面可以融合不同任务的情感信息, 缓解低资源场景下单个任务的训练数据匮乏问题, 另一方面可以保证模型随时间步持续更新.

(2) 本文提出了针对低资源场景下连续情感分析任务的持续注意力建模方法 CAM, 旨在解决低资源场景下连续情感分析任务的两大核心问题. 首先, CAM 通过情感掩码 Adapter (SMA) 保留单个任务的情感信息, 缓解灾难性遗忘, 其次, CAM 通过动态情感注意力 (DSA) 实现高效的情感信息融合.

(3) 本文构建了多个低资源场景下的连续情感分析数据集, 这可以有效评估低资源场景下不同方法的性能. 多个数据集上的实验结果和分析表明: CAM 具有最优的情感信息保留能力和情感信息融合能力, 并且能同时保持较高的运行效率.

本文第 1 节介绍低资源场景下的情感分析和持续学习的相关工作. 第 2 节介绍 CAM 的详细结构、运算过程和损失函数. 第 3 节介绍实验设置、基准方法和实验结果. 第 4 节通过 3 组实验分析, 进一步分析不同 CAM 中核心模块的作用, 同时验证 CAM 的性能和效率. 最后在第 5 节总结全文并展望未来发展方向.

1 相关工作

1.1 低资源场景下的情感分析

深度神经网络和庞大的语言模型在情感分析应用中无处不在. 然而, 由于它们需要大量训练数据, 训练代价高昂, 研究人员逐渐开始关注低资源场景下的情感分析. 最常见的低资源场景为训练数据不足的有监督情感分析任务. 在这种场景下, 手工标注费时费力, 对于某些专业领域的任务, 甚至无法找到足够的专家标注人员. 目前针对低资源场景下的研究大多会引入辅助数据和外部知识.

相关研究往往需要一个阈值来划分当前目标任务是否属于低资源场景. 根据目标任务类型的不同, 低资源场景的阈值也有所不同, 复杂任务一般需要更多资源, 因此阈值往往会比较高. Garrette 等人[5]认为词性标记任务的低资源场景阈值为 2 000 条标记的训练数据, Yang 等人[6]认为文本生成任务的低资源场景阈值为 35 万条标记的训练数据. Ke 等人[7]认为针对产品评论领域的情感分析任务的低资源场景阈值为 200 条标记的训练数据.

为了缓解目标任务的训练数据匮乏问题, 目前的研究往往会通过数据增强、迁移学习等方法寻找训练数据的替代形式. 数据增强旨在对训练数据进行各种转换, 保留标签的同时修改数据特性. Wei 等人[8]提出使用同义词替换方法扩充训练集, Raiman 等人[9]提出使用同类型实体替换方法实现数据增强. Xie 等人[10]回译进行数据增强, 提升了模型在 IMDB 数据集[11]上的情感分类性能. 迁移学习通过传输学习到的表示减少了对标记数据的需求. 目前

迁移学习应用于情感分析领域的一种常见做法是在目标任务上微调预训练语言模型, 常用的预训练语言模型包括 BERT[12]、RoBERTa[13]等. 这种做法可以充分利用预训练语言模型的知识, 对低资源场景下的任务十分有效[14]. 然而, 微调整个预训练模型代价较大, 且每个不同的下游任务都需要一个全新的微调模型, 这会导致额外的时空代价. 研究人员希望一个模型能够完成多个下游任务, Adapter 凭借其即插即用的特性脱颖而出.

Adapter 首先由 Houlsby 等人[15]提出, 旨在实现模型参数的高效利用和模型的快速微调. 此方法固定预训练模型参数, 仅微调 Adapter 模块的参数, 针对每个任务仅添加少量的可训练参数. 为了避免顺序微调[16]导致的灾难性遗忘问题, 同时缓解多任务学习中[17,18]的效率问题, Pfeiffer 等人[19]提出了基于 Adapter 的两阶段学习算法 AdapterFusion. 第 1 阶段训练每个任务独有的 Adapter, 第 2 阶段使用单独的 Fusion 层进行知识组合. 通过分离知识抽取和知识组合, AdapterFusion 可以有效缓解避免灾难性遗忘. Wang 等人[20]提出 K-Adapter, 旨在向预训练语言模型注入知识. K-Adapter 是在不同的 Transformer layer[21]间插入 Adapter, 这与传统的 Adapter 有所不同. 在下游任务上的实验表明 K-Adapter 捕获的知识比 RoBERTa[13]更多. Rücklé等人[22]提出 AdapterDrop, 在训练和推理时候移除底部 Transformer layer 的部分 Adapter, 该方法既可以保持模型性能, 又可以提高模型的推理效率.

本文首先按照 Ke 等人[7]提出的低资源场景阈值构建了多个低资源场景下的连续情感分析数据集. 其次, 本文基于迁移学习思想, 使用不同的 Adapter 来提取不同任务的特征. 最后, 本文受到 AdapterFusion 的启发, 提出动态情感注意力机制. 动态情感注意力一方面可以实现不同任务间的情感信息融合, 另一方面可以实现模型的持续更新, 这可以有效解决持续学习训练过程与 AdapterFusion 的两阶段学习算法之间的矛盾.

1.2 持续学习

持续学习的发展已经有几十年的历史, 其核心思想起源于认知神经科学中对于记忆和遗忘机制的研究[23]. 持续学习利用已学任务的知识提高对未来任务的泛化, 从而可以在不同分布的数据流下进行自适应学习. 为了在吸收新知识的同时保留旧知识, 持续学习需要解决的一个关键问题就是稳定性-可塑性问题[24]. 其中可塑性表示调整、整合新知识的能力, 稳定性表示不遗忘旧知识的能力. 现实中, 由于计算和存储资源有限, 在两者之间寻找效用最大的平衡点是持续学习研究的核心.

持续学习分为 3 种场景, 分别为任务持续学习、领域持续学习和类别持续学习[25]. 本文专注于任务持续学习, 在任务持续学习范式中, 模型总是会被告知训练或需要预测的是哪些任务. 持续学习方法主要分为 3 类, 分别是基于正则化的持续学习、基于回放的持续学习、基于参数隔离的持续学习, 相关研究如下所述.

基于正则化的持续学习方法的主要思想是在新任务的损失函数中加入正则项, 通过对损失函数施加约束来保护旧任务的知识不被新知识覆盖, 从而缓解灾难性遗忘问题. LwF[26]利用知识蒸馏[27]的思想, 使用蒸馏损失保证旧任务头的输出和当前任务头的输出尽量相似. L2[28]是一种经典的正则化持续学习方法. EWC[28]在 L2 的基础上添加了基于贝叶斯框架的算法. RotateEWC[29]在 EWC 的基础上旋转不同的任务的参数分布角度, 使得不同任务的分布与对应参数正交, 这样可以更好地缓解灾难性遗忘问题.

基于回放的持续学习方法会保留一部分具有代表性的数据, 在学习新任务时, 会把新任务的数据和保留的旧任务数据一起训练, 如何选取旧任务的代表性数据是该方法的关键. iCaRL[30]同样引入了蒸馏损失, 同时在训练新任务时保留一部分具有代表性的数据. GEM[31]针对 iCaRL 容易对旧数据过拟合这个问题提出了梯度片段记忆方法. A-GEM[32]在 GEM 的基础上, 只要求模型对某些任务的损失不再增加. 这样可以大大降低计算成本. DER++[33]是一种简单、有效的回放式持续学习方法, 该方法同样引入了蒸馏损失.

基于参数隔离的持续学习通过对新旧模型的参数进行不同程度的隔离, 缓解灾难性遗忘问题. PackNet[34]对新旧任务的参数实行硬隔离, 通过剪枝为新任务保留模型空间. HAT[35]使用硬注意力掩码机制根据任务的不同对模型的不同部分进行掩盖, 具有较好的抗遗忘能力, 但性能较弱. CAT[36]会衡量任务间的相似度, 然后把相似的任务输入一个额外的知识模块进行知识融合. B-CL[37]利用掩码机制保护参数, 同时利用胶囊网络融合知识, 但运行效率较低. CTR[7]在 B-CL 的基础上改进了胶囊网络中的路由算法, 缓解了路由算法的超参选取问题, 可以有效提升知识融合效果, 但总体运行效率还是较低.

通过分析 HAT、B-CL、CTR 的优点和不足, 本文提出了针对低资源场景下连续情感分析任务的持续注意力建模方法 (CAM), CAM 既可以通过情感掩码保留任务的情感信息, 又可以通过动态情感注意力融合不同任务的情感信息. 实验表明 CAM 性能显著超过以上方法, 具有最优的抗遗忘能力和情感信息融合能力, 并且总体的运行时间效率显著高于 B-CL 和 CTR.

2 针对低资源场景下连续情感分析任务的持续注意力建模方法

正如上文所述, 大数据驱动型模型性能强悍, 但其训练过程严重依赖高昂的标注成本和算力成本. 因此针对低资源场景下的情感分析研究尤为迫切. 此外, 在下游任务上微调预训练语言模型已经成为 NLP 领域的常用学习范式. 然而, 微调整个模型代价较大, 且不同的下游任务需要不同的微调模型, 这导致了额外的时空开销. 本文为此构建了低资源场景下的连续情感分析任务, 该任务存在两大挑战. 一方面模型要能捕捉并保留单个任务的情感信息, 另一方面模型还要能够融合不同任务间的情感信息, 以缓解单个情感分析任务的数据匮乏问题.

为了解决上述问题, 本文提出了针对低资源场景下连续情感分析任务的持续注意力建模方法 (CAM), 如图 2 所示. 不同的情感分析任务对应一个分类头 (CLS head), SMA 用于保留单个任务的情感信息, DSA 用于融合任务间的情感信息. 本文采用 BERT[12]作为基础框架, 通过第 4.2 节的实验评估, 最终决定将 CAM 插入 4 层 Transformer layer 内, 这 4 层的 ID 分别为 0、5、7、11, 这样既可以保证 CAM 在不同层间的分布较为均匀, 又可以大幅提高模型的训练和推理效率. 针对不同的情感分析任务构建一个对应的分类头 (CLS head), 并根据任务 ID 训练对应的分类头. 此外, 训练阶段仅微调 CAM、layer norm 和 CLS head 的参数, 冻结模型的其他参数. CAM 主要由情感掩码 Adapter (SMA) 和动态情感注意力 (DSA) 组成, 下面就 CAM 和 DSA 的细节予以介绍.

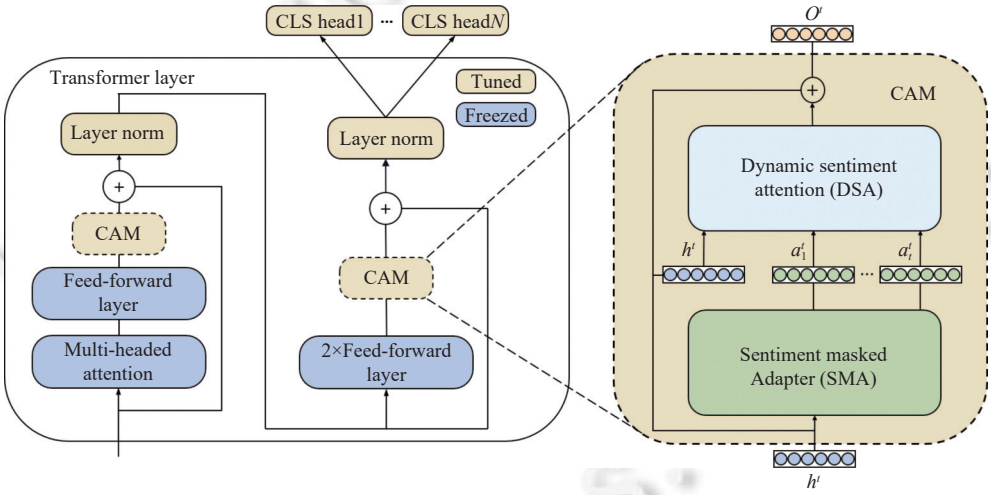

图 2 本文使用的 Transformer layer 和 CAM 模型总体结构

首先, SMA 为每个任务构建一个 Adapter, 旨在保留单个任务的情感信息. 具体而言, 当任务 ID 为 t 时, SMA 中共有 t 个 Adapter. 当任务 t 的数据输入 SMA 时, 模型会着重训练处于初始化状态的第 t 个 Adapter, 并且任务 t 的数据也会输入到前面 $t-1$ 个已经训练的 Adapter. 这样做一方面可以进一步训练前 $t-1$ 个 Adapter, 另一方面是利用不同 Adapter 所学的情感信息, 提取当前任务数据的特征, 用于后续的情感信息融合. 在模型使用任务 t 的数据训练时, 为了避免前面 $t-1$ 个已训练的 Adapter 发生灾难性遗忘, 本文提出了特定任务的掩码嵌入模块 (task-specific mask embedding, TSME), 用来生成任务对应的情感掩码. TSME 会生成可训练的情感掩码, 这可以掩盖 Adapter 的部分神经元, 保护 Adapter 已学的重要情感信息, 从而缓解灾难性遗忘问题.

其次, 本文利用 DSA 进行不同任务间的情感信息融合. DSA 会根据当前时间步和任务相似度动态融合 SMA 中不同 Adapter 提取的数据特征, 以融合不同 Adapter 所学的情感信息, 任务间的情感信息融合程度与任务间的相

似度息息相关. 例如, 当任务 ID 为 t 时, DSA 会将第 t 个任务的输入数据特征作为 Query, 然后将前 t 个 Adapter 提取得到的数据特征作为 Key 和 Value, 通过动态情感注意力进行情感信息融合.

本节第 2.1 节说明任务持续学习场景下的情感分析任务定义. 第 2.2 节详细介绍 SMA 的结构和 TSME 生成情感掩码的过程. 第 2.3 节介绍 DSA 的结构、初始化方式和情感信息融合过程. 第 2.4 节介绍 针对连续情感分析任务的损失函数.

2.1 连续情感分析任务定义

本文基于任务持续学习范式, 构建了连续情感分析任务. 在该任务形式下, 模型要随时间步学习一系列情感分析任务. 连续情感分析任务中的任务类型既可以是句子级情感分类, 也可以是属性级情感分类. 为了能够简单又有效地验证 CAM 的效果, 本文使用不同领域分布的句子级情感二分类任务来构建连续情感分析任务. 假设任务 t 的某一条评论文本输入为“Excellent book, the way it is written and what you get out of the book. Would recommend this book highly”. 首先对该句进行分词, 将分词后的句子记为 $I^t=\{w_1^t,\ldots,w_N^t\}$, 其中 N 为分词后的词语数目. 接着使用 *Bert-Tokenizer* 提取句子的表征作为模型的输入, 记为 $X^t=\{x_{\text{CLS}}^t,x_1^t,\ldots,x_N^t,x_{\text{SEP}}^t\}$. 然后将 X^t 输入 *BertModel*, 将 *BertModel* 的输出记为 $H^t=\{h_{\text{CLS}}^t,h_1^t,\ldots,h_N^t,h_{\text{SEP}}^t\}$. 由于本文任务形式为情感二分类, 所以我们使用 h_{CLS}^t 作为整句评论文本的表征. 将 h_{CLS}^t 输入任务 t 的分类头, 通过多层感知机 (multi-layer perceptron, MLP) 和 *Softmax* 函数得到预测结果, 记为 O_{CLS}^t. 最后通过 argmax 函数得到预测标签 P^t.

$$X^t=BertTokenizer(I^t) \tag{1}$$

$$H^t=BertModel(X^t) \tag{2}$$

$$O_{\text{CLS}}^t=Softmax(W_{\text{MLP}}^t\cdot h_{\text{CLS}}^t+b_{\text{MLP}}^t) \tag{3}$$

$$P^t=\text{argmax}(O_{\text{CLS}}^t) \tag{4}$$

其中, t 为任务 ID, *BertModel* 代表 BERT-base-uncased 模型, W_{MLP}^t, b_{MLP}^t 分别为任务 t 分类头中 MLP 的权重和偏置, P^t 为二进制标签, 即 $P^t\in\{0,1\}$.

2.2 用于保留情感信息的情感掩码 Adapter (SMA)

SMA 首先通过特定任务的掩码嵌入模块 (TSME) 生成情感掩码嵌入, 然后将情感掩码嵌入输入门控激活函数生成情感掩码, 最后将情感掩码与 Adapter 隐藏状态表征按位相乘. 为了减少模型的训练成本, 本文按照 HAT[35] 的设置, 将情感掩码设置为二进制硬编码, 即情感掩码中每一位的大小为 0 或 1. 详细过程如下所述.

假设当前任务 ID 为 t, 此时 SMA 中共有 t 个 Adapter, 如图 3 所示. 其中 Task ID 随时间步逐渐增加, TSME 用于生成情感信息掩码. 每个 Adapter 都有降维层 (FF Down) 和升维层 (FF Up), 因此每个 Adapter 都有两个中间表征. 将第 t 个 Adapter 的两个中间表征记为 $h_{a_{t,1}}^t$ 和 $h_{a_{t,2}}^t$, 其中上标 t 表示当前任务 ID 为 t, 下标 t 表示当前 Adapter 为 SMA 中的第 t 个 Adapter. 则 $h_{a_{t,1}}^t$ 表示任务 t 的数据输入第 t 个 Adapter 得到的第 1 表征. TSME 模块随机生成对应每个 Adapter 的情感掩码嵌入, 由于 Adapter 的中间表征有两个, 因此情感掩码嵌入也有两个, 分别记为 $E^1=\{e_1^1,\ldots,e_{t-1}^1,e_t^1\}$ 和 $E^2=\{e_1^2,\ldots,e_{t-1}^2,e_t^2\}$. 其中 E^1 表示对应每个 Adapter 第 1 表征的情感掩码嵌入, E^2 表示对应每个 Adapter 第 2 表征的情感掩码嵌入. 具体而言, e_1^1 表示对应于第 1 个 Adapter 第 1 表征的情感掩码嵌入, e_t^2 表示对应于第 t 个 Adapter 第 2 表征的情感掩码嵌入.

由于 Adapter 中两个表征的运算过程相同, 下面就以第 t 个 Adapter 的第 1 表征为例, 首先说明 TSME 如何生成对应表征的情感掩码, 然后说明情感掩码是如何进行运算的. 假设已有第 t 个 Adapter 的第 1 表征 $h_{a_{t,1}}^t$ 和 TSME 随机生成的情感掩码嵌入 e_t^1. 首先将 e_t^1 乘上放大尺度参数 s, 然后输入门控激活函数, 得到对应的情感掩码, 记为 m_t^1. 在本次实验中, 我们使用带有参数 s 的 *Sigmoid* 函数作为门控激活函数. 接着将 m_t^1 与 $h_{a_{t,1}}^t$ 按位相乘, 所得结果记为 $Input_{a_{t,Up}}$.

$$m_t^1=\sigma\left(s\cdot e_t^1\right) \tag{5}$$

$$Input_{a_{t,Up}}=h_{a_{t,1}}^t\otimes m_t^1 \tag{6}$$

其中, σ 表示 *Sigmoid* 函数, s 为模型的可训练参数, e_t^1 为 TSME 随机生成的情感掩码嵌入, $\otimes$ 表示向量间的按位相乘, $Input_{a_{t,Up}}$ 为第 t 个 Adapter 中 FF Up 的输入.

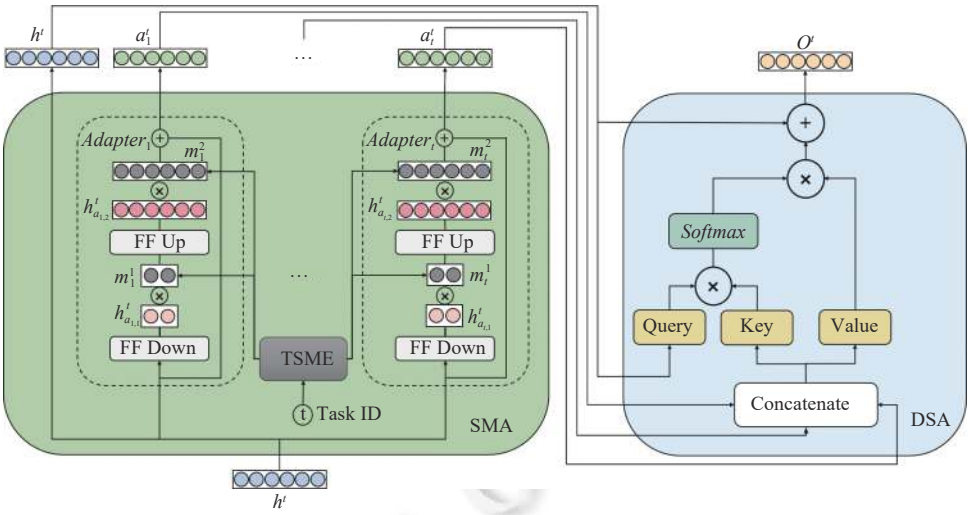

图 3 SMA 模块和 DSA 模块

将 m_t^1 与 $h_{a_{t,1}}^t$ 按位相乘的目的是用情感掩码的大小表明神经元对当前任务的重要程度, 从而保护重要的情感信息. 在学习新任务时, 模型会使用情感掩码修改梯度, 避免遗忘已学的情感信息. 具体而言, 假设当前任务 ID 为 t+1, m_t^1 中某个神经元的情感掩码大小为 1, 表明该神经元学到了有效的情感信息. 因此在当前任务的前向传播过程中, 该神经元会具有更高的权重. 而在反向传播阶段, 我们将 $1-m_t^1$ 乘上梯度, 此时该神经元的梯度会被修改为 0, 用以限制该神经元的参数更新, 从而保护已学的情感信息. SMA 利用任务 t 所学的情感掩码修改第 t 个 Adapter 在任务 t+1 上的梯度, 防止重要参数发生大幅度更新, 从而保留已经学习到的情感信息.

为了保证梯度正常传播, 本文引入一个放大尺度参数 s 使得 *Sigmoid* 函数成为可微的伪阶跃函数, $s_{\max}$ 代表之前任务训练阶段得到的 s 的最大值. 在每个任务的训练阶段开始时, 初始化当前任务的 s 为 1, 此时每个神经元的情感掩码都趋近于 0.5, 表明每个神经元都是同等活跃的. 随着训练过程的进行, s 根据公式 (8) 逐渐增大, 当 s 足够大时, 该函数就成为一种伪阶跃函数, 此时神经元的情感掩码 m_t^1 趋近于 0 或 1. 我们会取当前任务的 s 和已有的 $s_{\max}$ 之间的最大值更新 $s_{\max}$, 在测试阶段, 会使用更新后的 $s_{\max}$ 作为放大尺度参数.

$$g'_{a_{t,Up}} = \left(1-m_t^1\right) g_{a_{t,Up}} \tag{7}$$

$$s = \left(s_{\max} - \frac{1}{s_{\max}}\right)\frac{b-1}{B-1} + \frac{1}{s_{\max}} \tag{8}$$

梯度的修改过程如公式 (7) 所示, 其中 $g_{a_{t,Up}}$ 表示第 t 个 Adapter 中 FF Up 的原始梯度, m_t^1 为对应第 t 个 Adapter 中第 1 表征的情感掩码, $g'_{a_{t,Up}}$ 为修改后的最终梯度. s 的更新过程如公式 (8) 所示, 其中 b 表示当前的批次序号, B 表示一个 Epoch 包含的总批次. 每个任务的训练阶段开始时, s 会初始化为 1, $s_{\max}$ 为之前任务训练阶段得到的 s 的最大值, 用于更新当前任务的放大尺度参数 s. s 可以用来控制网络的稳定性和可塑性, 当 s 趋近于 1 时, 此时该函数趋近于普通的 *Sigmoid* 函数, 网络的可塑性最强. 随着训练的进行, s 逐渐增大, 该函数也转变为伪阶跃函数. 此时网络的可塑性下降, 但是稳定性增强. 在反向传播阶段, 通过情感掩码机制控制神经元梯度, 避免重要的参数发生变化.

参数 s 的引入可以实现可微的伪阶跃函数, 便于控制梯度传播. 但是会导致模型的情感掩码嵌入梯度很小, 无法达到理想的梯度更新幅度. 为了解决这个问题, 在每个 Epoch 的梯度更新前, 本文会对掩码嵌入的梯度进行补偿. 以第 t 个 Adapter 的第 1 表征为例, 梯度补偿计算过程如公式 (9) 所示:

$$g_{a_t}^1 = \frac{s_{\max}\left[\cosh\left(s \cdot e_t^1 + 1\right)\right]}{s\left[\cosh\left(e_t^1\right) + 1\right]} \tag{9}$$

其中, $g_{a_t}^1$ 表示针对第 t 个 Adapter 第 1 表征的情感掩码嵌入的补偿梯度, cosh 表示双曲余弦函数, $\cosh x = \frac{e^x + e^{-x}}{2}$, e_t^1 表示第 t 个 Adapter 第 1 表征的情感掩码嵌入, $s_{\max}$ 和 s 为控制网络活性的超参数.

2.3 用于融合情感信息的动态情感注意力 (DSA)

注意力机制考虑了序列的全部信息, 通过计算注意力得分来区分信息的重要程度, 然后通过 *Softmax* 函数对每一部分信息的权重进行归一化, 最后通过加权求和得到长序列的综合表示. 注意力机制擅长并行处理长序列文本, 因此在 NLP 领域中被广泛应用.

受到 AdapterFusion[19]的启发, 本文希望引入注意力机制用于融合不同 Adapter 学到的情感信息. 然而在持续学习场景下, 不同任务的数据是随着时间步持续到来的. 因此无法使用 AdapterFusion 的两阶段方法, 即先单独训练好每个任务的 Adapter, 然后通过训练一层单独的 Fusion 层进行融合. 因此, 本文提出动态情感注意力 (DSA), DSA 维护一组注意力参数 W_Q^t、W_K^t、W_V^t, 根据时间步变化和任务间的相似度进行动态融合, 与当前任务相似度更高的那些任务的 Adapter 所提取的情感信息会在融合过程中具有更大的权重.

为了保留 Adapter 抽取到的整体表示, 本文首先按照偏置为 0, 方差为 0.01 的规则, 随机生成 W_Q^t 和 W_K^t. 然后按照 Pfeiffer 等人[19]的设置, 本文将 W_V^t 的对角线数值初始化为 1, 其余位置按照均值为 0, 方差为 1E–6 的分布随机生成对应的权值. 此外, 本文在初始化过程中针对 W_K^t 使用 L2 范数, 目的是保持矩阵的稀疏程度, 避免引入额外参数和噪声.

如图 3 所示, 假设任务 ID 为 t. 首先将 CAM 的输入表示 h^t 作为 Query. 然后将 h^t 同时输入前 t 个 Adapter, 得到不同的表示 $a_1^t,\ldots,a_{t-1}^t,a_t^t$. 接着将 $a_1^t,\ldots,a_{t-1}^t,a_t^t$ 拼接起来得到 A^t. 在融合过程中, 我们直接将 A^t 作为 Key 和 Value. DSA 模块通过注意力机制融合不同 Adapter 抽取的表示, 可以利用不同 Adapter 的情感信息来解决任务. 由于我们拼接得到的 A^t 是随着时间步动态变化的, 随着任务数目增长, A^t 包含的特征也会逐渐增多, 因此 DSA 可以在持续学习场景下进行动态融合. 此外, 注意力大小反映了不同 Adapter 所学情感信息的相似度, 这进一步体现了任务间的相似度. 因此在情感信息融合过程中, 与当前任务相似度高的任务更加重要, 这些任务的情感信息会在融合过程中占据主导地位. 具体公式如下所示:

$$Q^t = W_Q^t h^t \tag{10}$$

$$a_i^t = Adapter_i(h^t) \tag{11}$$

$$A^t = Concatenate\left(a_1^t,\ldots,a_{t-1}^t,a_t^t\right) \tag{12}$$

$$K^t = W_K^t A^t \tag{13}$$

$$V^t = W_V^t A^t \tag{14}$$

$$Score^t = Softmax\left(\frac{Q^t \cdot K^{t\mathrm{T}}}{\sqrt{n}}\right) \tag{15}$$

$$O^t = Score^t \cdot V^t \tag{16}$$

其中, h^t 为 CAM 的初始输入数据表示, $Adapter_i$ 表示 SMA 中的第 i 个 Adapter, W_Q^t, W_K^t, W_V^t 为模型的可训练参数, n 为 Q^t 的特征维度, $Score^t$ 为归一化后的注意力得分, O^t 为 DSA 模块的输出.

2.4 针对连续情感分析任务的损失函数

在持续学习场景下, 模型会随时间步在不同的任务的训练集上进行训练, 因此本文的训练目标是最小化当前时间步对应任务的交叉熵损失. 此外, 为了保证任务掩码的稀疏性, 同时为未来任务预留参数空间, 本文额外增加一个 L1 正则项. 在训练阶段, 我们使用验证集数据调整模型参数, 并选用交叉熵损失最小的模型参数作为最优模型参数. 在测试阶段, 我们使用最优参数的模型来评估性能. 具体而言, 当模型完成任务 t 的训练后, 我们使用前 t 个任务的测试集数据验证模型的情感信息保留能力和情感信息融合能力. 假设模型当前任务 ID 为 t, 则损失函数

如公式 (17) 所示:

$$L^t\left(O_i^t,\widehat{Y}_i^t,m^t\right)=-\sum_{i=1}^{N}\left[\widehat{Y}_i^t\log\left(O_i^t\right)+\left(1-\widehat{Y}_i^t\right)\log\left(1-O_i^t\right)\right]+\lambda\|m^t\|_1 \tag{17}$$

其中, L^t 表示模型针对任务 t 的损失函数, N 为任务 t 的训练集中包含的所有评价文本的数目, O_i^t 是以任务 t 的评价文本 i 作为模型输入得到的输出, $\widehat{Y}_i^t$ 为任务 t 中评价文本 i 的情感极性标签, m^t 为模型完成任务 t 的训练后得到的情感信息掩码, λ 为 L1 正则化的超参数.

3 实验设置与结果

本节主要阐述实验细节, 内容组成如下: 第 3.1 节主要介绍实验设置, 包含数据集构建、实验超参数设置和实验所用的评价指标. 第 3.2 节主要介绍实验所用的相关基准方法. 第 3.3 节主要展示并分析了 CAM 和不同基准方法在 3 个数据集上的实验结果.

3.1 实验设置

3.1.1 数据集构建

如前文所述, 连续情感分析任务中可以包含不同的情感分析任务类型 (如句子级情感分类或属性级情感分类). 本文为了实现简单有效的评估, 基于两个包含句子级情感分类任务的公共基准数据集 Amazon Review Dataset[38]和 Mtl-Dataset[39], 构建了低资源场景下的连续情感分析任务数据集. 这可以有效评估低资源场景下不同方法的情感信息保留能力和情感信息融合能力, 下面就两个公共基准数据集和数据集构建过程予以介绍.

Amazon Reviews Dataset 数据集记录了用户对亚马逊网站里不同商品的评价, 包含数百万条亚马逊用户评论, 时间跨度从 1996 年 5 月至 2014 年 7 月. 商品被划分为 24 个领域, 如 Pet Supplies、Office Products、Kindle Store 等. Mtl-Dataset 是由 Liu 等人[39]提出的包含 16 个不同领域的评论文本数据集. 其中前 14 个数据集为针对不同产品的评论, 其余两个为 IMDB 数据集[11]和 MR 数据集[40]. IMDB 和 MR 均为电影评论数据集, 每个影评包含多个句子, 且情感是二元的, 常用于二元情感分类任务.

首先, 本文对两个公共基准数据集进行预处理. Amazon Reviews Dataset 的用户评分数据中包含商品 ID、评价文本、总体评分等. 总体评分分为 5 个等级 (1–5), 本文将评分为 1、2 的评论视为负面评论, 将评分为 4、5 的评论视为正面评论, 并按照上述规则提取 23 个不同领域的评价文本和对应的情绪极性标签. 因为 Automotive 领域下部分数据不足, 所以将该领域剔除, 使用剩余 23 个领域的数据.

其次, 为了方便与 BCL、CTR 的实验结果进行对比, 本文将 Amazon Review Dataset 划分为两个数据集 AR10 和 AR13, 其中 AR10 包含与 CTR 实验设置中相同的 10 个领域的商品评价, AR13 则包含其余的 13 种不同领域的商品评价. 本文使用 Mtl-Dataset 中已提取好的 16 个领域的评价文本和对应的情绪极性标签, 简称为 MD16.

考虑到标注成本和算力成本 , 本文人为构建了低资源场景下的连续情感分析任务数据集: AR10mini, AR13mini 和 MD16mini, 详细信息如表 1 所示. 在这 3 个数据集中, 不同情感分析任务都划分为训练集、验证集和测试集. 本文借鉴 Ke 等人[41]的设置, 仅对训练集做了低资源处理. 因为训练集在整个数据集的占比最大, 这样的设置在实践中能大大节约人工标注的成本. 此外, 我们认为分布更为多样化、数据更充足的验证集和测试集能够进一步增加任务难度, 从而更好地反映不同方法的有效性和泛化性. 因此, 本文的训练集仅包含 100 条正面评论和 100 条负面评论, 用于衡量模型在低资源场景下的学习能力. 验证集和测试集含有 250 条正面评论和 250 条负面评论, 用于验证模型保留情感信息和融合情感信息的能力.

表 1 预处理后 3 个数据集的详细数据

参数	AR10mini	AR13mini	MD16mini
Task number	10	13	16
Train	200	200	200
Dev	500	500	500
Test	500	500	500

3.1.2 超参数设置

本文模型中的所有超参数均通过多次实验得到, 在该设置下能充分验证本文方法的有效性. 本文使用的 BERT 模型使用 Huggingface Transformers 库中的 BERT-base-uncased 模型参数进行初始化 (https://huggingface.co/bert-base-uncased), 并将其最大输入长度设为 128, 此长度足以应对大部分评论文本[7]. 此外, 本文将 Adapter 的隐藏层维度和对应的掩码嵌入表示维度都设为 256, Dropout 设为 0.5, L1 正则系数设置为 0.5. 本文使用 Adam 优化器, 初始学习率设为 5E–5. 每个情感分析任务训练 10 个 Epoch, 训练阶段的 batch size 设置为 16, 验证和测试阶段的 batch size 设置为 32.

3.1.3 评价指标

本文使用准确率 (accuracy, *Acc*) 和 Marco-*F*1 (*MF*1) 值作为度量模型性能的指标, 使用 t 检验[42]评价两种方法之间性能差异的显著性. 此外, 本文进一步使用遗忘率 (forgetting rate, *FR*)[43]来验证模型的情感信息保留能力, *FR* 的计算过程如公式 (18) 所示:

$$FR = \frac{1}{N-1}\sum_{i=1}^{N-1} A_i^i - A_i^N \tag{18}$$

其中, N 表示任务总数, A_i^i 表示模型在任务 i 上完成训练后, 在任务 i 上测试得到的准确率, 称之为前向准确率 (forward accuracy). A_i^N 表示模型在任务 N 上完成训练后, 在任务 i 上测试的准确率, 此时模型已经学习完所有任务. 在持续学习场景下, A_i^N 是我们更关注的指标, 这体现了模型的最终性能. 由于模型学习完最后一个任务 N 后就进行测试, 所以 *FR* 的计算只考虑前 $N-1$ 个任务. *FR* 的取值范围在 $[-1, 1]$ 之间, *FR* 越小表明模型的情感信息保留能力越强. 当 *FR* 为 0 时表明模型不会发生遗忘, 当 *FR* 为负数时, 表明模型不仅没有遗忘, 反而借助情感信息融合, 获得了前向转移能力, 即能够利用旧任务的情感信息解决新任务.

3.2 基准方法

本文在 3 个低资源连续情感分析数据集上比较 CAM 与其他基准方法, 以全面评估 CAM 的有效性. 我们将选取的基准方法分为 3 大类, 具体如下所述.

(1) 多任务学习方法 (multi-task learning approach, MTL)

多任务学习方法将所有任务的训练集拼接成一个训练集, 然后利用拼接后的训练集进行训练, 该方法往往被认为是持续学习的性能上界[7]. 然而多任务学习方法无法随着数据更新动态更新模型, 每当出现新任务时, 都需要重新拼接训练集并重新训练, 这意味着高昂的训练成本和算力成本. 此外, 多任务学习需要一直保留所有任务的数据. 然而, 有些隐私数据有使用期限, 无法一直使用, 因此多任务学习经常会受到数据隐私问题的困扰. 在实验中, 我们拼接所有任务的训练集, 然后微调整个 BERT 模型.

(2) 非持续学习方法 (non-continual learning approach, NCL)

非持续学习方法不考虑情感信息保留问题, 直接使用不同任务的数据训练模型. 本文使用 3 种模型框架, 具体如下所示.

- BERT: 按照任务序列依次训练, 根据当前任务数据微调整个 BERT 模型和对应任务的分析头. 该方法微调参数最多, 最容易导致情感信息的灾难性遗忘.
- BERT-Frozen: 冻结 BERT 模型, 仅根据当前任务数据微调每个任务对应的分析头. 该方法微调参数最少, 不容易发生灾难性遗忘, 但分类头学习能力有限, 性能上限较低.
- BERT-Adapter: 冻结 BERT 模型, 根据当前任务数据微调 Adapter 和对应任务的分类头. 该方法微调参数适中, 抗遗忘能力介于前两种方法之间.

(3) 持续学习方法 (continual learning approach, CL)

为了保留情感信息, 避免情感信息的灾难性遗忘, 持续学习方法通常会添加正则项, 或者回放旧任务数据. 本文选择了几种具有代表性的方法和当前性能最佳的方法作为基准方法, 这些方法的细节如下. 为了保证公平, 以下方法都使用 BERT-Adapter 作为基础框架, 即训练过程中冻结 BERT 模型, 仅微调分类头和 Adapter. 针对 B-CL、CTR 这类在 Adapter 基础上添加其余模块的方法, 实验中会微调分析头、Adapter 和方法中的额外模块.

● L2[28]: L2 是一种简单且经典的基于正则化的持续学习方法.

● EWC[28]: EWC 是一种十分流行的正则化方法, 该方法基于贝叶斯框架算法, 引入了一个额外的和参数有关的正则损失, 该损失会根据不同参数的重要性来鼓励新任务训练得到的新模型参数尽量靠近旧模型参数, 从而避免情感信息的灾难性遗忘.

● DER++[33]: DER++是一种回放式持续学习方法, 该方法结合了知识蒸馏和正则化, 可以利用有限的资源, 提升模型的泛化性.

● HAT[35]: HAT 使用硬注意力机制根据任务的不同对模型的不同部分进行掩盖, 从而为每个任务分配模型的不同部分, 同时还使用正则化项对掩盖矩阵进行稀疏性约束, 具有良好的抗遗忘能力.

● B-CL[37]: B-CL 使用胶囊网络提取底层特征, 并使用动态路由机制融合不同胶囊提取的特征. 此外, B-CL 借鉴了 HAT 的硬注意力机制, 对模型的不同部分进行掩盖, 实现不同任务间的参数隔离. B-CL 通过固定某些任务重要的神经元参数, 从而保留情感信息, 缓解灾难性遗忘问题.

● CTR[7]: CTR 在 B-CL 的基础上把动态路由机制换成了转移路由机制, 转移路由机制一方面可以更好地融合相似任务的特征, 提升模型的知识融合效果, 另一方面也解决了动态路由超参选取较为困难的问题, 具有良好的抗遗忘能力和知识融合能力.

3.3 实验结果

在持续学习领域, 不同的任务序列对模型性能有一定影响[7]. 因此, 本文随机生成每个数据集的不同任务序列. 在实验中, 我们随机抽取 3 个任务序列, 并取其结果平均值作为最终的实验结果. CAM 和其他基准方法在 3 个数据集上的性能如表 2 所示. 其中 *MF*1 和 *Acc* 用来评估模型的情感知识融合能力 (越高越好), *FR* 用来评估模型的情感知识保留能力 (越低越好). 由于 MTL 拼接所有任务的训练数据, 模型只会训练一次, BERT-Frozen-NCL 方法只训练特定任务的分类头, 所以这两种方法不存在灾难性遗忘问题, 因此没有 *FR* 指标.

表 2 CAM 与其他基准方法的实验结果 (%)

Backbone	Baseline	AR10mini			AR13mini			MD16mini		
		*MF*1	*Acc*	*FR*	*MF*1	*Acc*	*FR*	*MF*1	*Acc*	*FR*
BERT	MTL	82.83	83.93	—	84.79	84.85	—	86.65	86.68	—
	NCL	69.30	71.92	5.85	62.40	66.46	17.68	56.61	62.56	22.04
BERT-Frozen	NCL	64.72	66.05	—	70.75	71.31	—	67.92	68.36	—
BERT-Adapter	NCL	41.71	53.15	14.65	49.56	53.62	21.83	41.89	51.18	20.80
	EWC	51.47	56.71	10.08	59.77	64.31	12.27	69.45	70.24	12.91
	HAT	76.81	79.01	−3.22	80.95	81.38	−2.22	83.29	83.58	0.59
	DER++	59.80	68.19	20.43	60.45	62.06	27.57	66.36	69.70	18.28
	L2	62.75	65.94	14.53	57.90	61.34	19.37	58.12	60.89	19.77
	B-CL	78.95	79.84	−1.39	79.58	80.08	0.72	80.39	80.95	3.07
	CTR	79.37	80.82	−3.44	81.67	82.40	−0.42	84.23	84.40	−0.35
	CAM	**81.15**	**82.27**	**−6.83**	**83.90**	**84.00**	**−2.27**	**85.22**	**85.29**	**−1.37**

通过分析实验结果, 可以得到以下信息.

(1) CAM 在 3 个数据集上的 *MF*1 和 *Acc* 显著超越其他基准方法, 十分逼近持续学习方法的理论性能上界 MTL, 这表明 CAM 可以有效捕捉不同任务的情感知识. 与 CTR 相比, CAM 在 AR10mini 和 AR13mini 上的性能显著提升 (p-value<0.05), 这进一步说明 CAM 具有最优的情感信息融合能力.

实验结果表明, 考虑情感信息融合的方法 (CAM、CTR 等) 的性能显著超越不考虑情感信息融合的方法 (EWC、DER++等). 这说明针对低资源场景下的连续情感分析任务, 模型的情感信息融合能力十分重要, 因为这可以充分利用不同任务的数据, 从而有效缓解单个任务训练数据匮乏问题.

CAM 在 3 个数据集上的性能显著超越 EWC、HAT、DER++、B-CL (p-value<0.05), 这充分验证了 CAM 的

有效性. CAM 相较于考虑情感信息融合且性能最好的基准方法 CTR, 在 AR10mini、AR13mini 和 MD16mini 这 3 个数据集上分别提升了 1.78%/1.45% (*MF*1/*Acc*)、2.23%/1.60% (*MF*1/*Acc*) 和 0.99%/0.89% (*MF*1/*Acc*). t 检验结果表明, 相较于 CTR, CAM 在 AR10mini 和 AR13mini 上提升显著 (p-value<0.05), 这说明 CAM 具有最优的情感信息融合能力. 此外, CAM 在 3 个数据集上的性能与 MTL 的差距仅为 1.08%/1.66% (*MF*1/*Acc*)、0.89%/0.85% (*MF*1/*Acc*) 和 1.43%/1.39% (*MF*1/*Acc*), 这说明 CAM 十分逼近持续学习方法的理论性能上界 MTL, 从而进一步验证了 CAM 的有效性.

(2) CAM 在 3 个数据集上的 *FR* 显著低于其他基准方法, 这说明 CAM 具有最优的情感信息保留能力. 此外, CAM 在 3 个数据集上的 *FR* 均为负数, 这说明 CAM 可以进一步实现情感信息的前向转移.

本文通过遗忘率 (*FR*) 来评估模型的情感信息保留能力, *FR* 指标越小, 模型的情感信息保留能力越强. 考虑情感信息保留的方法 (如 HAT、CTR) 的 *FR* 指标显著低于其他基准方法, 且 *FR* 较低的方法性能普遍较好, 这说明针对低资源场景下的连续情感分析任务, 模型的情感信息保留能力十分重要, 因为这可以有效缓解灾难性遗忘问题.

CAM 在 3 个数据集上的 *FR* 指标均为最小, 具体来说, 本文提出的 CAM 相较于 *FR* 最小的基准方法 CTR, 在 AR10mini、AR13mini 和 MD16mini 这 3 个数据集上分别继续降低了 3.39%、1.85% 和 1.02%, 这表明 CAM 具有最优的情感信息保留能力. 此外, CAM 在 3 个数据集上的 *FR* 都为负数, 这表明 CAM 在持续学习过程中不仅没有发生遗忘, 反而通过情感信息融合, 具有了一定的前向转移能力. 即模型能够利用已学任务的知识, 提升对新任务的学习效果.

(3) 本文基于持续学习方法构建的连续情感分析任务形式可以有效缓解低资源场景下单个情感分析任务中训练数据匮乏的问题.

实验结果表明随着数据集中任务数目增多, 持续学习方法的性能普遍都在逐渐增强. 具体来说, 相较于 AR10mini, CAM 在 AR13mini 和 MD16mini 上分别提升了 2.75%/1.73% (*MF*1/*Acc*) 和 4.07%/3.02% (*MF*1/*Acc*), 这体现了低资源场景下连续情感分析任务的意义, 该任务形式能够利用持续方法让模型随时间步学习多个情感分析任务, 这可以有效缓解单个任务训练数据匮乏问题, 从而提升模型性能.

另一方面, 随着任务数量增多, 不同方法的 *FR* 普遍会逐渐增大. 具体来说, 相较于 AR10mini, CAM 在 AR13mini 和 MD16mini 上的 *FR* 分别提升了 4.56% 和 5.46%, 这是由于持续学习场景中模型要依次学习每个任务, 需要学习的任务越多, 模型就越容易遗忘. 这与任务数量越多, 持续学习方法的性能越高并不矛盾. 因为 *FR* 关注的是模型的抗遗忘能力, 而 *MF*1、*Acc* 关注的是模型的总体性能.

4 实验分析

本节进行了 3 组实验来验证 CAM 的有效性和运行效率. 第 4.1 节通过针对 CAM 核心模块的消融实验, 验证 CAM 中不同模块的作用和有效性. 第 4.2 节通过改变 CAM 插入的 Transformer layer 数, 观察 CAM 在不同数据集上的性能变化趋势, 并说明本文实验设置的合理性. 第 4.3 节比较 CAM 和不同基准方法在不同数据集上的训练、预测时间, 验证 CAM 的运行效率.

4.1 消融实验与分析

本节设计了针对 CAM 核心模块的消融实验, 进一步验证 CAM 中各模块的效果, 消融实验结果如表 3 所示. 具体而言, CAM(–DSA, –SMA) 只使用一个 Adapter 学习所有任务. CAM(–SMA) 同样只使用一个 Adapter, 但会将 Adapter 抽取不同任务数据得到的特征输入 DSA 模块进行情感信息融合. CAM(–TSME) 删除了特定任务的掩码嵌入模块, 在模型的训练过程中不使用掩码. CAM(–DSA) 在融合不同 Adapter 抽取得到的特征时仅使用简单的 Concatenate 方法. 实验结果如表 3 所示, 接下来展开具体分析.

由表 3 可以发现, 删除 SMA 或者 TSME 都会引起严重的灾难性遗忘问题. 具体而言, 在 3 个数据集上, CAM(–SMA) 相较于 CAM 分别下降了 16.08%/15.72% (*MF*1/*Acc*)、15.12%/14.08% (*MF*1/*Acc*) 和 15.65%/13.81%

(*MF*1/*Acc*), 这表明情感掩码十分重要, 它可以有效提升模型的情感信息保留能力. 此外, CAM(–TSME) 在 3 个数据集上的性能稍强于 CAM(–SMA), 这表明 SMA 中的多个 Adapter 也能一定程度上缓解灾难性遗忘问题.

表 3 CAM 核心模块的消融实验结果 (%)

Model	AR10mini		AR13mini		MD16mini	
	*MF*1	*Acc*	*MF*1	*Acc*	*MF*1	*Acc*
CAM(–DSA, –SMA)	65.44	67.33	62.06	64.15	61.42	63.08
CAM(–SMA)	65.07	66.55	68.78	69.92	69.57	71.48
CAM(–TSME)	67.73	71.66	69.76	70.31	71.42	71.79
CAM(–DSA)	78.91	80.68	80.02	80.17	82.50	82.90
CAM	**81.15**	**82.27**	**83.90**	**84.00**	**85.22**	**85.29**

随着任务数量增加, CAM(–SMA) 的性能逐渐超越 CAM(–DSA, –SMA), 这说明注意力机制可以有效融合不同任务的情感信息, 从而提升模型的性能. 此外, CAM(–DSA) 在 3 个数据集上与 CAM 的 *MF*1 性能差距分别为 2.24%、3.88%、2.72%. 这说明相较于简单的 Concatenate 融合方法, DSA 可以更有效地融合不同 Adapter 抽取的特征, 实现不同任务间的情感信息融合, 从而提升模型的性能.

4.2 关于 Transformer layer 数的对比实验与分析

本文使用 BERT-base-uncased 模型作为基础框架. 传统方法普遍是将 Adapter 插入到每层 Transformer layer 中, 仅微调参数量很小的 Adapter. 然而, 在任务持续学习场景下, 随着任务数量的增加, 使用参数隔离方法训练得到的模型参数会逐渐增大. 例如, 假设当前任务 ID 为 1, 此时 SMA 中只有 1 个 Adapter 被激活. 而当任务任务 ID 变为 t 后, SMA 中会动态增加 $t-1$ 个 Adapter, 此时 SMA 中共有 t 个 Adapter 被激活. 为了给后续任务保留空间, 同时减少模型训练代价, CAM 会选择性的插入部分 Transformer layer. 本节通过对比实验观察不同的 Transformer layer 数对 CAM 性能的影响并验证 CAM 设置的合理性.

实验结果如图 4 所示, 3 条折线分别对应 CAM 在 3 个数据集上的性能, 横坐标为 CAM 插入的 Transformer layer 层数, 纵坐标为 CAM 的 *Acc* 和 *MF*1 指标. 本文发现, 当 CAM 仅插入 1 层 Transformer layer 时 (分别选择第 1 层和最后一层进行实验, 然后取两者结果的平均值作为最终结果), 模型的遗忘情况较为严重. 当 CAM 插入 3 层 Transformer layer 时 (本文按照 Pfeiffer 等人[19]的设置, 选取 0, 5, 11 层), 模型的效果提升到较为稳定的水平. 当插入的 Transformer layer 大于 3 时, CAM 的性能逐渐趋于稳定. 为了保证 CAM 的信息抽取能力, 同时加快模型训练, 本文最终将 CAM 插入的 Transformer layer 数设定为 4, 分别为 0、5、7、11.

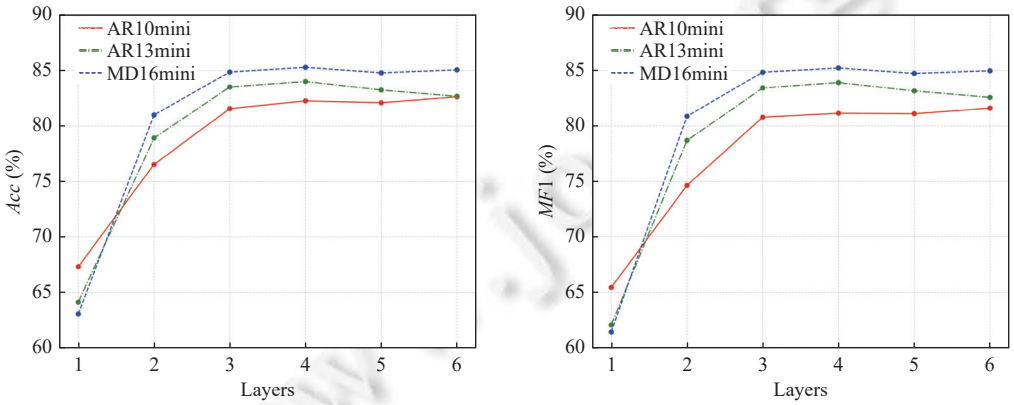

图 4 关于 Transformer layer 数的对比实验结果

4.3 针对 CAM 和其他基准方法的运行时间分析

本节通过比较不同方法在 3 个数据集上的训练时间和预测时间, 说明了 CAM 具有较高的时间效率. 实验设

备环境为单张 NVIDIA GeForce RTX 2080 Ti, 实验所用的深度学习框架为 PyTorch, 不同方法的实验参数与第 3.1.2 节保持一致. 不同方法在 3 个数据集上的运行时间如表 4 所示, 其中训练时间指的是模型按顺序在所有任务上完成训练所需的时间 (单位为 min), 预测时间指的是使用训练完成后的最终模型预测所有任务的测试集所需的时间 (单位为 s), 随着数据集中任务数量的增加, 不同方法的训练时间和预测时间普遍都逐渐增加. 根据表 4 的实验结果, 可以得到以下信息.

表 4　不同方法在 3 个数据集上的运行时间

Backbone	Baseline	Training time (min)			Inference time (s)		
		AR10mini	AR13mini	MD16mini	AR10mini	AR13mini	MD16mini
BERT	MTL	9.89	14.78	19.14	21.21	27.81	30.06
	NCL	17.55	24.31	29.29	12.02	15.86	25.17
BERT-Frozen	NCL	8.74	10.68	13.27	18.97	22.51	25.43
BERT-Adapter	NCL	31.27	36.85	38.23	26.29	38.52	44.37
	EWC	46.17	50.46	58.72	24.95	35.58	44.86
	HAT	31.14	42.68	50.77	17.19	43.64	45.58
	DER++	55.35	78.53	100.28	15.96	32.68	41.93
	L2	38.47	48.07	53.31	50.73	74.29	108.13
	B-CL	120.14	297.15	464.21	134.30	249.12	319.14
	CTR	157.94	371.92	601.08	184.36	226.73	356.87
	CAM	**38.98**	**48.87**	**72.67**	**46.56**	**87.56**	**117.25**

(1) 在模型性能和运行时间之间取得平衡十分重要.

通过实验结果比较, 本文发现任务的训练形式、保留的额外数据和可微调参数量会直接影响运行时间. 具体而言, BERT-MTL 和 BERT-Frozen-NCL 的训练时间非常短. 前者是因为多任务学习形式会把所有任务的训练数据拼接起来, 仅进行一次训练和一次预测, 后者是因为 BERT-Frozen-NCL 只训练每个任务的分类头, 因此可微调参数很少. 然而, 多任务学习形式成本较高, 无法持续更新. BERT-Frozen-NCL 可微调参数过少, 模型学习能力有限, 因此模型性能较低. DER++通过保留额外数据缓解灾难性遗忘问题, 然而额外数据的选取对性能影响较大, 并且随着任务数量增加, DER++保留额外数据的空间代价增长较大. 为了缓解上述问题, 本文认为在模型的性能和运行时间之间取得平衡十分重要.

(2) 随着任务数量增加, 不同方法的训练时间和预测时间普遍都增加. 与其他方法的比较说明: CAM 能够有效平衡模型的性能和运行时间.

当任务数量逐渐增加时, 正则化方法 EWC 的训练时间和预测时间增长幅度较小, 但性能相对较低. 回放式持续学习方法 DER++需要保留额外数据, 训练数据会随着任务数量增加逐步增长, 因此 DER++的训练时间增长幅度很大. 参数隔离的持续学习方法 (B-CL、CTR) 具有相对良好的性能, 但由于模型的参数随着任务数量增加逐步增长, 且路由算法的运算较为繁杂, 因此训练时间和预测时间增长幅度最大. HAT 具有良好的训练时间和预测时间, 但性能落后于 CTR、CAM.

本文提出的 CAM 在 3 个数据集上的运行时间都要显著优于 B-CL 和 CTR, 训练时间相较于 DER++有所降低. 此外, 随着任务数量的增长, CAM 运行时间的增长幅度相较于 B-CL 和 CTR 显著降低, 这表明 CAM 能保持较高的运行效率. 第 3.2 节的实验结果表明 CAM 性能显著超越其他基准方法. 综上所述, CAM 在 3 个数据集上既有最优的性能表现, 又能保持较高的运行效率, 这说明 CAM 能够在模型的性能和运行时间之间保持良好的平衡.

5　总结和未来工作

近年来, 在下游任务上微调预训练语言模型已经成为 NLP 领域的经典范式. 然而微调整个预训练语言模型代价较大, 且需要大量下游任务的标签数据, 这限制了低资源场景下情感分析任务的发展. 为了解决上述问题, 本文引入持续学习思想, 旨在通过更新维护的方式, 让一个模型去学习多个下游任务. 基于持续学习范式, 本文构建了

低资源场景下的连续情感分析任务, 并利用 Adapter 缩减训练代价. 低资源场景下的连续情感分析任务存在两个关键问题, 分别是单个任务的情感信息保留和不同任务间的情感信息融合. 针对这两个关键问题, 本文提出了针对低资源场景下连续情感分析任务的持续注意力建模方法 CAM. 具体而言, CAM 由情感掩码 Adapter 和动态情感注意力组成. 其中, 情感掩码 Adapter 通过情感掩码机制避免模型遗忘所学的情感信息, 动态情感注意力通过融合不同 Adapter 抽取的特征, 实现情感信息融合. 在多个数据集上的实验结果与分析表明, CAM 的性能超越了目前最先进的基准方法 CTR, 具有最优的情感信息保留能力和情感信息融合能力, 并且具有较高的运行效率.

未来的工作重点是如何进一步改善情感信息融合机制, 同时提高运行效率. 具体而言, 一方面希望寻找能够有效捕捉情感信息的方法, 并利用其他融合机制实现任务间的情感信息融合, 另一方面希望借鉴 AdapterDrop[22]和其他剪枝方法用以简化模型参数, 从而提高模型的运行效率.

References:

[1] Hedderich MA, Lange L, Adel H, Strötgen J, Klakow D. A survey on recent approaches for natural language processing in low-resource scenarios. In: Proc. of the 2021 Conf. of North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2021. 2545–2568. [doi: 10.18653/v1/2021.naacl-main.201]

[2] Lohar P, Xie GD, Bendechache M, Brennan R, Celeste E, Trestian R, Tal I. Irish attitudes toward COVID tracker APP & privacy: Sentiment analysis on Twitter and survey data. In: Proc. of the 16th Int'l Conf. on Availability, Reliability and Security. Vienna: ACM, 2021. 37. [doi: 10.1145/3465481.3469193]

[3] McCloskey M, Cohen NJ. Catastrophic interference in connectionist networks: The sequential learning problem. Psychology of Learning and Motivation, 1989, 24: 109–165. [doi: 10.1016/S0079-7421(08)60536-8]

[4] French RM. Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences, 1999, 3(4): 128–135. [doi: 10.1016/S1364-6613(99)01294-2]

[5] Garrette D, Baldridge J. Learning A part-of-speech tagger from two hours of annotation. In: Proc. of the 2013 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Atlanta: Association for Computational Linguistics, 2013. 138–147.

[6] Yang Z, Wu W, Yang J, Xu C, Li ZJ. Low-resource response generation with template prior. In: Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int'l Joint Conf. on Natural Language Processing. Hong Kong: Association for Computational Linguistics, 2019. 1886–1897. [doi: 10.18653/v1/D19-1197]

[7] Ke ZX, Liu B, Ma NZ, Xu H, Shu L. Achieving forgetting prevention and knowledge transfer in continual learning. In: Proc. of the 35th Conf. on Neural Information Processing Systems. NeurIPS, 2021. 22443–22456.

[8] Wei J, Zou K. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In: Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int'l Joint Conf. on Natural Language Processing. Hong Kong: Association for Computational Linguistics, 2019. 6382–6388. [doi: 10.18653/v1/D19-1670]

[9] Raiman J, Miller J. Globally normalized reader. In: Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing. Copenhagen: Association for Computational Linguistics, 2017. 1059–1069. [doi: 10.18653/v1/D17-1111]

[10] Xie QZ, Dai ZH, Hovy E, Luong MT, Le QV. Unsupervised data augmentation for consistency training. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 525. [doi: 10.5555/3495724.3496249]

[11] Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C. Learning word vectors for sentiment analysis. In: Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language technologies. Portland: Association for Computational Linguistics, 2011. 142–150.

[12] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional Transformers for language understanding. In: Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1. Minneapolis: Association for Computational Linguistics, 2019. 4171–4186. [doi: 10.18653/v1/N19-1423]

[13] Liu YH, Ott M, Goyal N, Du JF, Joshi M, Chen DQ, Levy O, Lewis M, Zettlemoyer L. RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692, 2019.

[14] Cruz JCB, Cheng C. Evaluating language model finetuning techniques for low-resource languages. arXiv:1907.00409, 2019.

[15] Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, De Laroussilhe Q, Gesmundo A, Attariyan M, Gelly S. Parameter-efficient transfer learning for NLP. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 2790–2799.

[16] Phang J, Févry T, Bowman SR. Sentence encoders on STILTs: Supplementary training on intermediate labeled-data tasks.

arXiv:1811.01088, 2018.

[17] Liu XD, He PC, Chen WZ, Gao JF. Multi-task deep neural networks for natural language understanding. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: Association for Computational Linguistics, 2019. 4487–4496. [doi: 10.18653/v1/P19-1441]

[18] Zhang Y, Yang Q. A survey on multi-task learning. IEEE Trans. on Knowledge and Data Engineering, 2022, 34(12): 5586–5609. [doi: 10.1109/TKDE.2021.3070203]

[19] Pfeiffer J, Kamath A, Rücklé A, Cho K, Gurevych I. AdapterFusion: Non-destructive task composition for transfer learning. In: Proc. of the 16th Conf. of the European Chapter of the Association for Computational Linguistics: Main Volume. Association for Computational Linguistics, 2021. 487–503. [doi: 10.18653/v1/2021.eacl-main.39]

[20] Wang RZ, Tang DY, Duan N, Wei ZY, Huang XJ, Ji JS, Cao GH, Jiang DX, Zhou M. K-Adapter: Infusing knowledge into pre-trained models with Adapters. In: Proc. of the 2021 Findings of the Association for Computational Linguistics. Association for Computational Linguistics, 2021. 1405–1418. [doi: 10.18653/v1/2021.findings-acl.121]

[21] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010. [doi: 10.5555/3295222.3295349]

[22] Rücklé A, Geigle G, Glockner M, Beck T, Pfeiffer J, Reimers N, Gurevych I. AdapterDrop: On the efficiency of Adapters in Transformers. In: Proc. of the 2021 Conf. on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2021. 7930–7946. [doi: 10.18653/v1/2021.emnlp-main.626]

[23] Parisi GI, Kemker R, Part JL, Kanan C, Wermter S. Continual lifelong learning with neural networks: A review. Neural Networks, 2019, 113: 54–71. [doi: 10.1016/j.neunet.2019.01.012]

[24] Mermillod M, Bugaiska A, Bonin P. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. Frontiers in Psychology, 2013, 4: 504. [doi: 10.3389/fpsyg.2013.00504]

[25] van de Ven GM, Tolias AS. Three scenarios for continual learning. arXiv:1904.07734, 2019.

[26] Li ZZ, Hoiem D. Learning without forgetting. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2018, 40(12): 2935–2947. [doi: 10.1109/TPAMI.2017.2773081]

[27] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv:1503.02531, 2015.

[28] Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, Milan K, Quan J, Ramalho T, Grabska-Barwinska A, Hassabis D, Clopath C, Kumaran D, Hadsell R. Overcoming catastrophic forgetting in neural networks. Proc. of the National Academy of Sciences of the United States of America, 2017, 114(13): 3521–3526. [doi: 10.1073/pnas.1611835114]

[29] Liu XL, Masana M, Herranz L, van de Weijer J, López AM, Bagdanov AD. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In: Proc. of the 24th Int'l Conf. on Pattern Recognition. Beijing: IEEE, 2018. 2262–2268. [doi: 10.1109/ICPR.2018.8545895]

[30] Rebuffi SA, Kolesnikov A, Sperl G, Lampert CH. iCaRL: Incremental classifier and representation learning. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 5533–5542. [doi: 10.1109/CVPR.2017.587]

[31] Lopez-Paz D, Ranzato MA. Gradient episodic memory for continual learning. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6470–6479. [doi: 10.5555/3295222.3295393]

[32] Chaudhry A, Ranzato MA, Rohrbach M, Elhoseiny M. Efficient lifelong learning with A-GEM. In: Proc. of the 7th Int'l Conf. on Learning Representations. New Orleans: OpenReview.net, 2019.

[33] Buzzega P, Boschini M, Porrello A, Abati D, Calderara S. Dark experience for general continual learning: A strong, simple baseline. In: Proc. of the 34th Conf. on Neural Information Processing Systems. Vancouver: NeurIPS, 2020. 15920–15930.

[34] Mallya A, Lazebnik S. PackNet: Adding multiple tasks to a single network by iterative pruning. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 7765–7773. [doi: 10.1109/CVPR.2018.00810]

[35] Serra J, Suris D, Miron M, Karatzoglou A. Overcoming catastrophic forgetting with hard attention to the task. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 4548–4557.

[36] Ke ZX, Liu B, Huang XC. Continual learning of a mixed sequence of similar and dissimilar tasks. In: Proc. of the 34th Conf. on Neural Information Processing Systems. Vancouver: NeurIPS, 2020. 18493–18504.

[37] Ke ZX, Xu H, Liu B. Adapting BERT for continual learning of a sequence of aspect sentiment classification tasks. In: Proc. of the 2021 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2021. 4746–4755. [doi: 10.18653/v1/2021.naacl-main.378]

[38] McAuley J, Targett C, Shi QF, van den Hengel A. Image-based recommendations on styles and substitutes. In: Proc. of the 38th Int'l

ACM SIGIR Conf. on Research and Development in Information Retrieval. Santiago: ACM, 2015. 43–52. [doi: 10.1145/2766462.2767755]

[39] Liu PF, Qiu XP, Huang XJ. Adversarial multi-task learning for text classification. In: Proc. of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: Association for Computational Linguistics, 2017. 1–10. [doi: 10.18653/v1/P17-1001]

[40] Pang B, Lee L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics. Ann Arbor: Association for Computational Linguistics, 2005. 115–124. [doi: 10.3115/1219840.1219855]

[41] Ke ZX, Liu B, Wang H, Shu L. Continual learning with knowledge transfer for sentiment classification. In: Proc. of the 2021 Joint European Conf. on Machine Learning and Knowledge Discovery in Databases. Ghent: Springer, 2021. 683–698. [doi: 10.1007/978-3-030-67664-3_41]

[42] Liu YY, Su YT, Liu AA, Schiele B, Sun QR. Mnemonics training: Multi-class incremental learning without forgetting. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 12242–12251. [doi: 10.1109/CVPR42600.2020.01226]

[43] Yang YM, Liu X. A re-examination of text categorization methods. In: Proc. of the 22nd Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Berkeley: ACM, 1999. 42–49. [doi: 10.1145/312624.312647]

张涵(1998－), 男, 硕士生, CCF 学生会员, 主要研究领域为自然语言处理.

罗佳敏(1997－), 女, 博士生, CCF 学生会员, 主要研究领域为自然语言处理.

王晶晶(1990－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为自然语言处理.

周国栋(1967－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为自然语言处理.