基于多任务协同训练的视频语义理解模型

任君翔[1]，张鹏涛[2]，郑少棉[3]，杨平[4]，曹辰捷[5]
[1]中国太平洋保险（集团）股份有限公司
[2]小红书科技有限公司
[3]蚂蚁智能服务团队
[4]西南交通大学
[5]复旦大学
renjunxiang1215@sina.com

摘要：本文的目标，是解决视频语义理解问题，即通过视频及相关文本描述，输出视频的分类标签和语义标签。我们基于多任务协同训练的思路，将比赛任务拆解为两个文本分类和两个实体识别模型，达到良好的效果。

关键词：文本分类，实体识别，协同训练

1 引言

在移动互联网、大数据的时代背景下，互联网上的视频数据呈现爆发式增长，作为日益丰富的信息承载媒介，视频的深度语义理解是诸多视频智能应用的基础，具有重要的研究意义和实际应用价值。传统基于感知的视频内容分析缺乏语义化理解能力，而充分利用知识图谱的语义化知识并结合多模态学习和知识推理技术，有望实现更深入的视频语义理解。知识增强的视频语义理解任务，期望融合知识、NLP、视觉、语音等相关技术和多模态信息，为视频生成刻画主旨信息的语义标签，从而实现视频的语义理解。本评测任务以互联网视频为输入，在感知内容分析（如人脸识别、OCR 识别、语音识别等）的基础上，期望通过融合多模信息，并结合知识图谱计算与推理，为视频生成多知识维度的语义标签，进而更好地刻画视频的语义信息。

基于 bert 的文本分类以及实体识别任务正逐渐成为研究热点。关于文本分类，Xu 等[1]在自集成方式的基础上又提出了 Self-Distillation-Average(SDA)方式。SDA 进一步利用蒸馏的方式来提升模型效果。在该方法中包括两个模型，student model 和 teacher model，其中 student model 损失函数的值由预测值和真实标签之间的交叉熵损失函数值和 student model 和 teacher model 输出值之间的均方误差值相加构成，其中 teacher model 的参数是 student model 在 T 个时间步内参数的平均值，该模型在 IMDb 数据集上取得了优异的结果。Pan 等[2]提出了一个简单而通用的方法来调整文本分类任务中基于 Transformer 的编码器的微调。具体来说，在微调过程中，通过扰动模型的单词嵌入来生成对抗性示例，并对干净示例和对抗性示例进行对比学习，以教导模型学习噪声不变表示。在几个基准测试任务中，微调 BERT-Large 模型比 BERT-Large 基线平均提高了 1.7%。Malte 等[3]使用 BERT 来获取文档的表示，同时通过统计样本的元数据（如作者的个数、是否为学术性标题、标题所包含的单词数等）和基于维基百科的图嵌入模型（Graph embedding model）来为最后的分类提供额外的信息，从而最终提升分类模型的性能。Reimers 等[4]第一次展示了如何利用上下文词嵌入的强大功能对与主题相关的参数进行分类和聚类，从而在两个任务和多个数据集上取得令人印象深刻的结果。Gcs 等[5]提出将去噪 BERT (DeBERT)叠加作为一种新颖的编码方案，用于对不正确的句子进行不完全的意图分类和情绪分类。该模型的结构为嵌入层和普通变压器层的叠

加，类似于传统的 BERT，然后是新型降噪 Transformer 层。该模型的主要目的是通过对含有缺失词的句子进行隐藏嵌入重构，提高 BERT 对不完整数据的鲁棒性和有效性。关于实体识别任务，Li 等[6]等提出将嵌套实体问题巧妙得转换为阅读理解（MRC）来做。Canasai 等[7]针对低资源 NER 任务提出解决方案，提出对容易获取的句子级别标签以及标注的词级别标签联合建模，并引入 self-attention 的变种，在三种低资源数据上面验证效果。Li 等[8]针对中文 NER 提出了一种整合词汇信息的 FLAT(Flat-Lattice Transformer)框架，将 Lattice 结构转变为 Flat 结构，改变以往 Lattice 结构的复杂运算，并引入特定位置编码。

本文基于 CCKS 2021 知识增强的视频语义理解赛道，提出了多任务协同训练的方式进行语义理解。我们在预训练模型下游任务的基础上，通过文本分类模型识别视频的分类标签，通过实体识别和文本分类两类模型识别视频的语义标签，最大程度的实现了视频语义的信息挖掘。

2 数据

我们先分别对 title、asr、ocr 的文本长度进行分析，可以发现除了 title 外，asr 和 ocr 的都属于过长的文本。其中，title 包含了主要信息，asr 和 ocr 主要起到补充的作用。分类问题主要是以 title 为主，拼接 asr 和 ocr 的字段到 512 长度；实体识别任务，则需要考虑滑窗的方式分段抽取。

表 1 文本长度

分位数	title	asr	ocr	title+asr+ocr
50	19	64	133	210
75	27	155	344	499
90	35	276	887	1018
95	42	348	1700	1755
99	56	517	5152	5571
max	118	1924	66209	66498

2.1 视频分类

Topic 一二级组合的类别在三百左右不算特别多，通过对 topic 的分析发现，一级标题的分布相对均匀（图 1），一二级合并后就有明显的厚尾分布（图 2），大都聚集在前几类上。

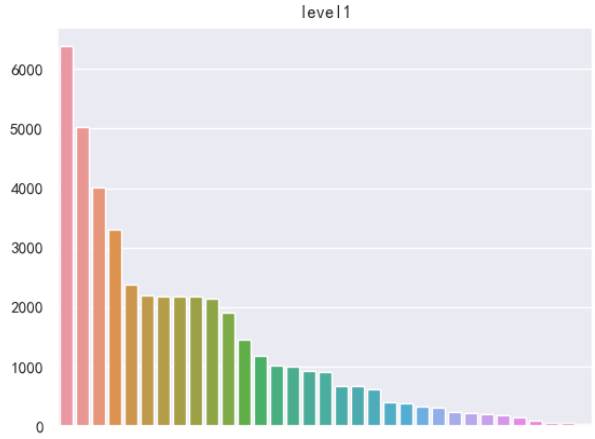

图 1　topic level1 分布

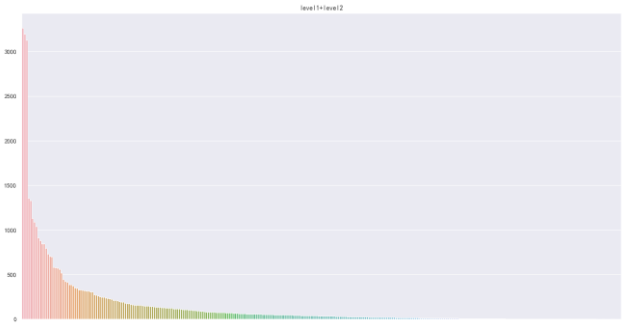

图 2　topic level1+ level2 分布

2.2 视频语义

我们发现，语义标签 tag 在原文中出现的占比并不是很高，原文去掉空格后占比有所提升。40%存在于 title 中、20%不在 title 但存在于 asr 和 ocr 中，40%不在原文中出现，分析结果见表 2。

表 2 tag 标签分布

字段	频数（原始文本）	频数（去掉空格）
不在原文	62649	54657
仅 title	18908	26956
仅 ocr	14256	11865
title+ocr	8652	10984
title+asr+ocr	5381	6167
asr+ocr	4781	3996
title+asr	2070	2460
仅 asr	1923	1535

不在原文的，一种表现为“实体 \1 侧面”，共计 11039 条、2960 种，占比约 10%。通过频数分析可以发现“侧面描述”存在明显的长尾分布，大多聚集在前几类，如“糖醋鱼\1 做法”、“瑜伽\1 教程”（见图 3）

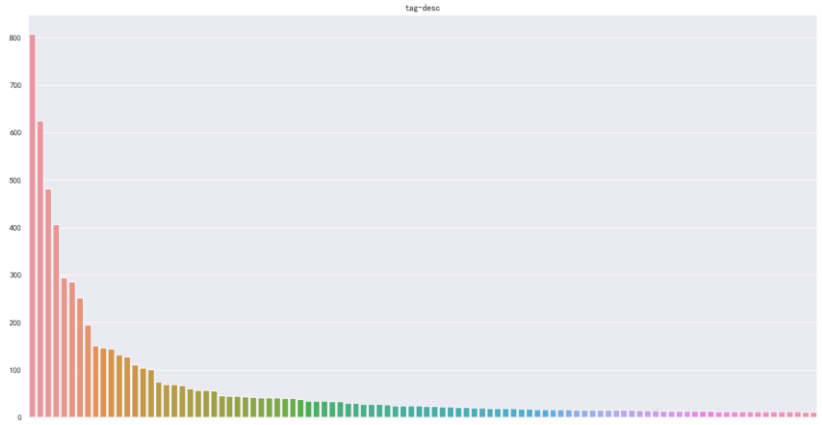

图 3　tag 侧面描述分布

剩余的表现为实体属性的类别，共计 43616 条、20956 种，占比约 30%。通过频数分析也可以发现存在明显的长尾分布，大多聚集在前几类，但是难点在于不知道类型描述对应的具体实体，如“中餐”、“古装剧”（见图 4）

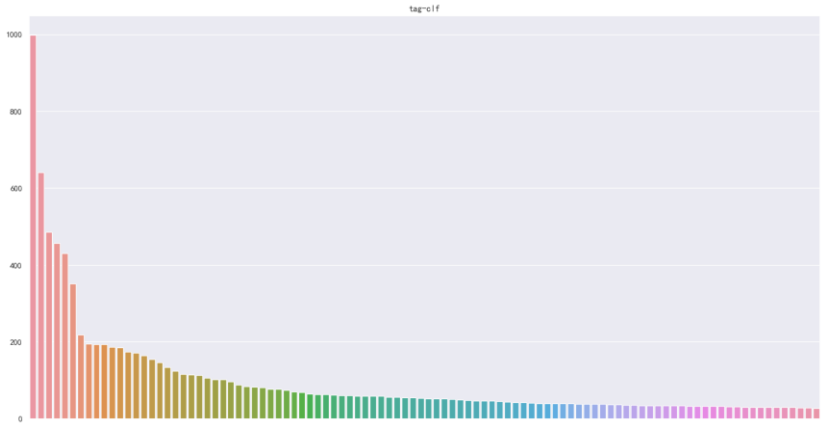

图 4　tag 属性类别分布

3 模型

3.1 视频分类

分类任务是预测 topic 的类别，由于一级、二级标题合并后并不多，我们采用组合的方式进行文本分类。整体模型思路为预训练模型（PTM）、最大池化后分类以及对 PTM 中 Word-Embedding 部分的 FGM 对抗训练，见图 5。

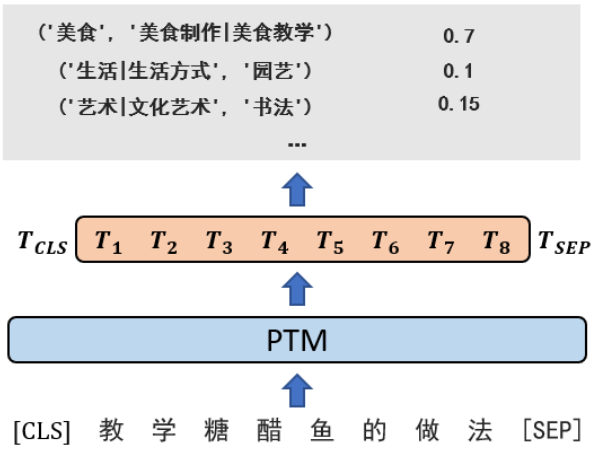

图 5　topic 文本分类

3.2 视频语义

经过数据分析我们已经知道，语义标签 tag 中很大一部分不在原文中出现，例如：教学糖醋鱼的做法，糖醋鱼、糖醋鱼 \1 制作、中国菜。

针对这三类 tag，我们归类为以下的任务：

1、原文出现的，可以作为实体识别任务；

2、原文未出现，表现为“实体”+“侧面描述”的，原文出现的部分作为实体识别任务，“侧面描述”作为实体分类任务；

3、原文未出现，表现为“实体类型”的，作为文本分类任务；

第一和第三类任务的标签存在重叠，和第二类任务没有重叠，三类任务的输出合并后作为 tag 的最终输出。

3.2.1 实体识别

第一类是相对简单的，tag 出现在原文中，例如：原文=教学糖醋鱼的做法，tag=糖醋鱼。这类标签直接采用双指针的实体识别模型，同样对 PTM 中 Word-Embedding 部分的 FGM 对抗训练，模型记为 tag-ner，见图 6。

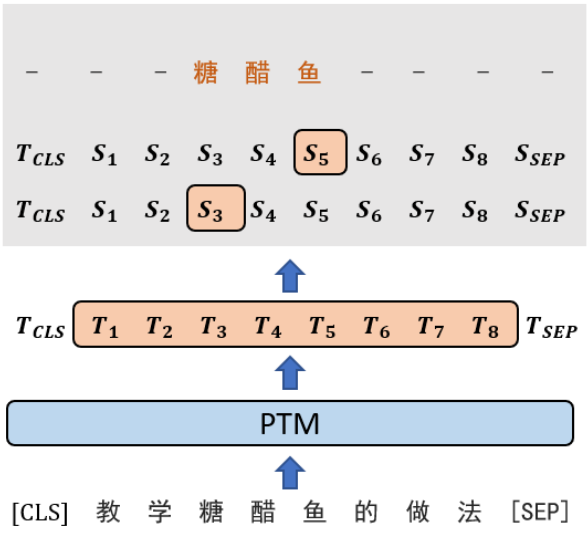

图 6　tag 实体识别

3.2.2 实体侧面描述

第二类是相对复杂的，tag 未出现在原文中，有\1 作为分隔符，例如：原文=教学糖醋鱼的做法，tag=糖醋鱼 \1 制作。这类标签同样采用双指针的实体识别模型，我们仅分析出现在原文的实体，且描述词汇出现频数大于 20 的。指针额外增加分类标记，同样对 PTM 中 Word-Embedding 部分的 FGM 对抗训练，模型记为 tag-desc，见图 7

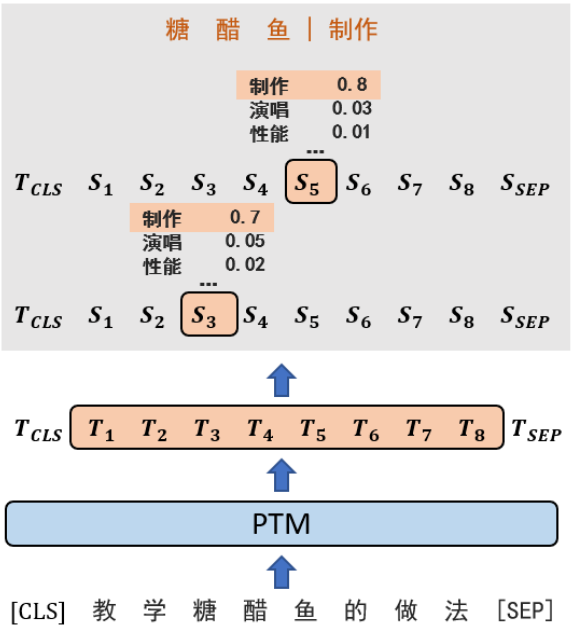

图 7　tag 实体识别+分类

3.2.3 实体属性归类

第三类是相对简单的，tag 未出现在原文中，但能反映实体的属性、和 topic 相关，例如：

原文=教学糖醋鱼的做法，tag=中国菜。由于实体属性这类标签暂时没法对应到到具体的实体，所以采用文本分类的方式统一处理，且仅对分类词汇出现频数大于 10 的，同样对 PTM 中 Word-Embedding 部分的 FGM 对抗训练，记为 tag-clf，见图 8。

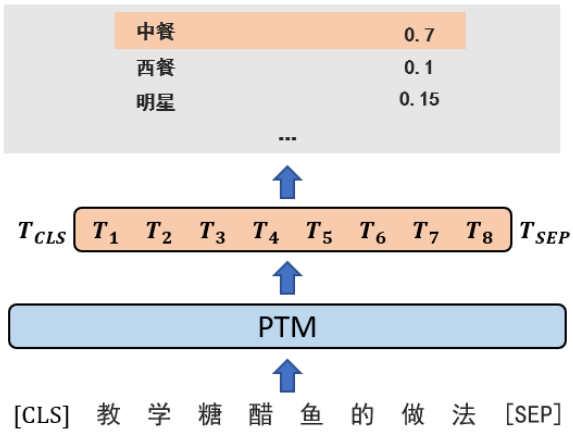

图 8　tag 实体分类

3.3 模型训练及融合

我们采用将视频语义理解拆解为文本分类和实体识别两类任务，由于任务之间信息并不是完全独立，但标签分布存在较大的差异，联合训练在此任务的表现可能不会太好。因此我们考虑通过协同训练（co-training）的方式迭代共享，即同类任务初期单独训练，后期在适当的时间会交替预训练模型的权重再继续训练，从而保持信息共享。目前业界对协同训练仍在探索中，我们测试发现略有提高但不是很显著。

各个单模训练完成后，topic 采用类别概率求和取最大值，tag 和 tag-desc 采用投票的方式、tag-clf 采用和 topic 一样的类别概率求和取最大值。

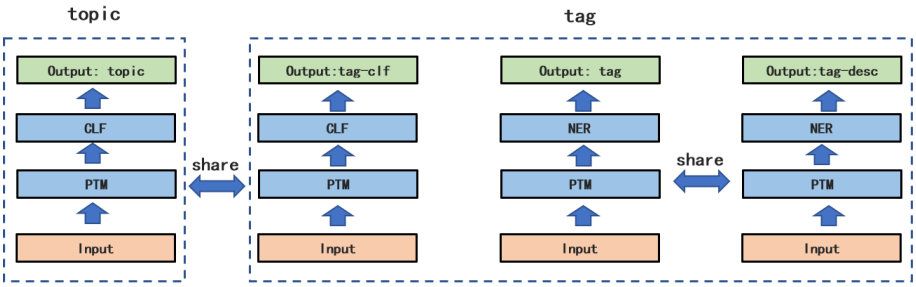

图 9　整体结构

4 实验结果

初赛我们针对各类 PTM 进行的 finetune，并对各类任务进行了测试。在文本分类问题上，BERT 系列模型相对较好。在实体识别任务上，Electra-180g 表现远胜其他模型、其次是 Nezha-cn-base。初赛测试分数如下：

表 5 topic 初赛得分

预训练	模型方案	线上得分
BERT-base	clf	0.191
Nezha-cn-base	clf	0.189
Electra-180g-base	clf	0.184
ensemble	clf	0.198

表 6 tag 初赛得分

预训练	模型方案	线上得分
BERT-base	ner	0.235
Nezha-cn-base	ner	0.249
Electra-180g-base	ner	0.255
Electra-180g-base	ner+desc	0.257
Electra-180g-base	ner+desc+clf	0.263
ensemble		0.283

5 结论

针对比赛任务，本文拆解为“文本分类（CLF）+实体识别（NER）”两类任务，共计四个模型，较全面的分析了各类标签的情况并给出了解决方案，同时提出基于协同训练的方式增强模型间的信息共享。最终 a 榜分数 0.4816、b 榜分数 0.5166，以绝对优势获得冠军。

无论是 topic 的分类还是 tag 的识别，模型分数还不是很高，厚尾分布的标签依旧没有被识别出来，我们将在后续就这些问题做进一步的研究。

参考文献

[1]. Xu Y， Qiu X， Zhou L , et al. Improving BERT Fine-Tuning via Self-Ensemble and Self-Distillation[J]. 2020.

[2]. https://arxiv.org/abs/2107.10137

[3]. Ostendorff M， Bourgonje P， Berger M , et al. Enriching BERT with Knowledge Graph Embeddings for Document Classification[J]. 2019.

[4]. Reimers N， Schiller B， Beck T , et al. Classification and Clustering of Arguments with Contextualized Word Embeddings[J]. 2019.

[5]. Gcs A， Ml B . Stacked DeBERT: All attention in incomplete data for text classification[J]. Neural Networks, 2021, 136:87-96.

[6]. Li X， Feng J， Meng Y , et al. A Unified MRC Framework for Named Entity Recognition[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020.

[7]. Kruengkrai C， Nguyen T H， Aljunied S M , et al. Improving Low-Resource Named Entity Recognition using Joint Sentence and Token Labeling[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020.

[8]. Li X， Yan H， Qiu X , et al. FLAT: Chinese NER Using Flat-Lattice Transformer[C]//

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020.