视频描述报告

秦建洲，张博彦，郭浩文

December 2023

1 背景介绍

随着科技的飞速发展，视频已成为互联网上最受欢迎的内容形式之一。与此同时，人们对视频内容的理解与交互需求也日益增长。视频描述生成技术，正是在这样的背景下应运而生，成为将深度学习应用于计算机视觉 (Computer Vision) 与自然语言处理 (Natural Language Processing) 领域的绝佳范例。

视频描述生成，简单来说，即给定一段视频，通过计算机技术，自动输出一段描述这段视频内容的文字。这个看似简单的任务，实则需要融合多个领域的技术，包括计算机视觉、自然语言处理、信息抽取等。

近年来，随着深度学习的迅猛发展，数据驱动的方法在视频描述生成任务中占据了主导地位。通过大量的数据训练，模型可以学习到从视频中提取关键信息并转化为自然语言的能力。这不仅极大地提高了视频描述的准确度，还为进一步推动相关技术的发展奠定了基础。

值得一提的是，为了提高模型的鲁棒性，很多时候一个视频会对应多个人工标注的描述。这种方式不仅为训练提供了丰富的数据，还为模型提供了更全面的视角，使其能够更好地适应各种情况。

而视频描述生成技术的发展并非一帆风顺。在早期的探索中，由于技术限制和数据不足，很多方法都遭遇了瓶颈。但随着大数据时代的来临，以及深度学习技术的不断突破，视频描述生成开始取得显著进展。如今，我们已经可以轻松地让计算机为我们生成一段关于任何视频的描述。

视频理解技术在生活中具有广泛的应用价值。在互联网领域，通过对视频内容的理解，我们可以实现更高效的视频搜索、更智能的视频摘要、更精准的视频问答系统等。例如，利用视频理解技术，用户只需输入关键词或问

题，系统便能快速从海量视频中筛选出相关内容，大大节省了用户的时间和精力。此外，在广告、二维码识别、无意义直播识别等领域，视频理解技术也发挥了重要作用。

在安防领域，视频理解技术的应用更为广泛。通过对监控视频的分析，我们可以实时检测异常事件、识别暴恐涉政内容等，为公共安全提供有力保障。同时，人车分析等技术也为交通管理提供了便利。

在机器人领域，视频理解技术同样大有可为。通过机器视觉和深度学习技术，机器人可以实现自主导航、定位、抓取等功能。这不仅提高了机器人的智能化水平，还为工业自动化、物流配送等领域提供了新的解决方案。

此外，在扶残助残方面，视频理解技术同样发挥了重要作用。例如，通过为盲人提供导航、将电影或短视频描述给盲人听等技术，我们可以帮助盲人更好地融入社会、提高生活质量。

总的来说，视频描述生成技术作为深度学习在计算机视觉与自然语言处理领域的应用典范，已经取得了显著的成果并展现出广阔的发展前景。未来随着技术的不断进步和应用场景的拓展，视频描述生成技术将继续发挥更大的作用，为人类生活带来更多便利与价值。

2 深度学习方法

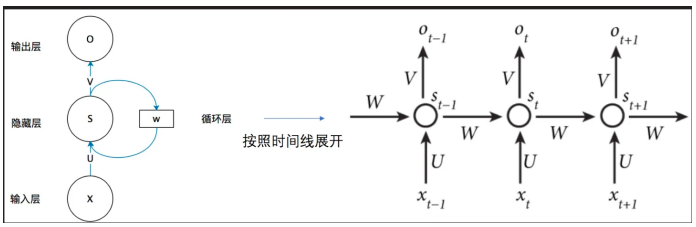
2.1 RNN 模型

循环神经网络（Recurrent Neural Network，RNN）的诞生确实是神经网络领域的一大突破，特别是在处理序列数据方面。传统的神经网络，如多层感知机（MLP），在处理输入和输出之间存在固定映射关系的问题时表现良好。然而，对于序列数据，如时间序列、语音、文本等，这些数据的输入和输出之间往往存在一种动态的、时间上的关系，这是传统神经网络难以有效处理的。

1986 年，Elman 等人提出的循环神经网络通过引入循环连接来解决这一问题。在 RNN 中，每个神经元的输出不仅作为下一层的输入，还会反馈回自身，形成一个循环。这种循环结构使得 RNN 能够具有“记忆”功能，能够捕捉序列数据中的时间依赖性。。RNN 的主要优点包括：

- 处理变长序列：不同于传统神经网络需要固定长度的输入，RNN 可以处理任意长度的序列数据，这使得它在处理自然语言等变长序列数据时具有很大的优势。

- 参数共享：RNN 在每个时间步使用相同的参数，这大大减少了模型的参数数量，降低了模型的复杂度，同时也使得模型更容易训练。
- 捕捉时间依赖性：由于 RNN 的循环结构，它能够捕捉序列数据中的时间依赖性，这对于处理语音、文本等具有时序关系的数据非常有效。



然而，RNN 也存在一些局限性，如梯度消失和梯度爆炸问题，这限制了其在处理长序列数据时的性能。为了克服这些问题，后续的研究提出了许多改进的 RNN 变体，如长短时记忆网络（LSTM）和门控循环单元（GRU）等。这些变体通过引入门控机制等技巧，有效地缓解了梯度消失问题，使得 RNN 能够在处理长序列数据时取得更好的性能。

2.2 LSTM 模型

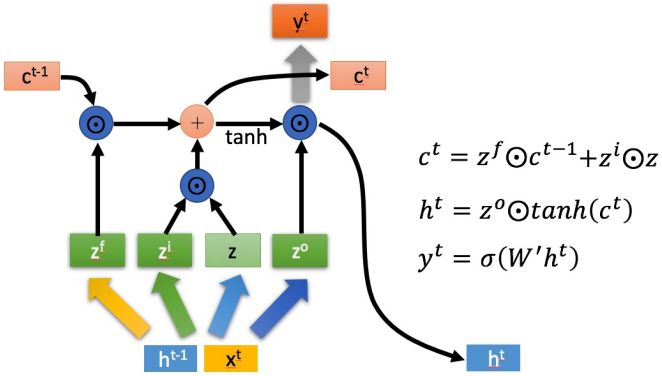
长短时记忆单元（LSTM）是循环神经网络（RNN）的一种重要变体，它的提出为解决 RNN 的梯度消失问题提供了一种有效的方案。Hochreiter 和 Schmidhuber 在 1997 年的开创性工作使得 LSTM 成为当前深度学习领域中广泛应用的模型之一。

LSTM 的核心思想是通过引入门控机制来控制信息的流动。具体来说，LSTM 有三个门控：输入门、输出门和遗忘门。这些门控的作用是决定哪些信息应该被记住，哪些信息应该被遗忘，以及哪些信息应该被输出。通过这种方式，LSTM 能够选择性地记住或遗忘信息，从而长时间地保存和传输上下文信息。

相比于标准的 RNN，LSTM 具有以下优点：

- 缓解梯度消失问题：梯度消失是标准 RNN 的一个主要问题，它导致模型在处理长序列时无法有效地学习到有用的信息。而 LSTM 通过门控机制有效地控制了梯度的传播，从而避免了梯度消失的问题。
- 更强的上下文信息处理能力：由于 LSTM 能够长时间地保存和传输信息，它能够更好地处理需要长时间依赖的序列数据，如语音、文本等。

- 参数共享：与标准 RNN 一样，LSTM 在每个时间步使用相同的参数，这大大减少了模型的参数数量，降低了模型的复杂度，同时也使得模型更容易训练。



然而，虽然 LSTM 具有许多优点，但它也有一些局限性。例如，由于其复杂的门控机制，LSTM 的计算复杂度相对较高。此外，对于一些简单的任务，使用 LSTM 可能会过度复杂化模型。因此，选择使用 LSTM 或其他 RNN 变体需要根据具体任务和数据来决定。

2.3 S2VT 模型

S2VT 模型，全称为 Sequence to Sequence Video to Text 模型，在 2015 年由 Venugopalan 等人首次提出。这一模型的提出，标志着视频描述生成领域的一个重要里程碑。[1]

S2VT 模型的核心思想是将通用的序列到序列（Seq2Seq）模型应用于视频描述任务中。Seq2Seq 模型是一种深度学习模型，主要用于处理序列到序列的映射问题，例如机器翻译、对话生成等。在 S2VT 模型中，首先通过一个 LSTM 网络对视频帧序列进行编码，将视频帧序列转化为一个固定长度的向量表示；然后，另一个 LSTM 网络对编码后的向量进行解码，生成相应的文本描述。

整个 S2VT 模型可以接受 RGB 图像或光学流图像两种输入类型。对于 RGB 图像输入，通常使用卷积神经网络（CNN）对每一帧图像进行特征提

取，得到每个帧的特征表示，然后作为 LSTM 网络的输入。而对于光学流图像输入，由于其包含了视频帧之间的时序信息，可以直接作为 LSTM 网络的输入。

S2VT 模型的优点在于它能够处理可变长度的输入帧。传统的 RNN 模型在处理变长序列时会出现梯度消失或梯度爆炸的问题，而 LSTM 网络通过引入门控机制，有效地解决了这些问题。S2VT 模型还能够学习并使用视频的时序结构。由于视频是连续的时序数据，每一帧都与前后帧存在关联，S2VT 模型能够捕捉这种时序信息，生成更加准确和连贯的文本描述。

此外，S2VT 模型还通过学习语言模型来生成既符合语法规范又能自然表达视频内容的句子。语言模型的训练通常采用大规模语料库，通过对大量文本数据的分析，学习到语言的语法和语义规律。这样，S2VT 模型在生成文本描述时，就能够遵循语言规则，避免出现语法错误或不自然的语言表达。

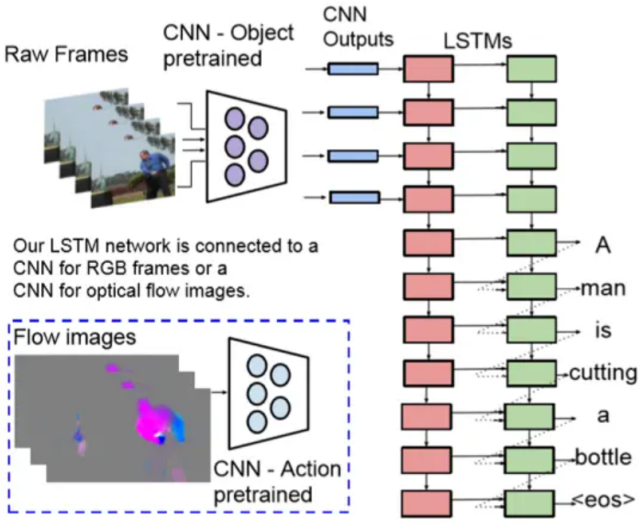


图 1: S2VT 方法-用于视频描述的序列到序列模型。它包含两个 LSTM 网络，首先读取帧序列，然后生成一个单词序列。整个模型的有 RGB 图像或光学流图像两种输入类型，都是经过 CNN 网络再作为 LSTM 的输入。

3 实验数据集

在本次竞赛中，数据集被精心地划分为两个主要部分：原视频和视频描述数据。

原视频部分是数据集的核心，包含了比赛所需的所有视频素材。这些视频以.avi 格式进行存储，以确保比赛参与者在处理时能够有统一的标准和兼容性。在训练集中，共有 3420 个视频片段，这些视频都是从央视视频网站上精心挑选的，涵盖了各种主题和内容，旨在为参赛者提供丰富多样的训练样本。这些视频片段的长度各异，从几秒到几十秒不等，反映了现实生活中的视频内容多样性。

与此同时，与每个原视频相配套的是视频描述数据部分。这部分数据与原视频一一对应，为每个视频提供了相应的文本描述。这些描述是由专业人员根据视频内容撰写，确保了描述的准确性和完整性。这些描述数据不仅为模型提供了必要的输入信息，还帮助模型更好地理解视频内容，从而生成更有意义的输出。

视频描述数据部分，由班上所有同学共同参与视频标注，同一个视频由 12 个同学分别标注，标注格式为 id,video_name,video_class 和 description，以.csv 格式进行保存，之后由助教进行数据的统计和规范化，将.csv 转化为.json 文件，.json 文件中的数据格式参考 MSR-VTT 数据集格式。具体形式如下：


```
{
  "category": 9,
  "url": "https://tv.cctv.com/2023/05/01/VIDERxDwIiK0QEZcxCO1VORR230501.
  "video_id": "G_23000",
  "start␣time": "0:3:47,3",
  "end␣time": "0:3:56,12",
  "split": "train",
  "id": 23000
}
```


- category 为视频类别，包括时政 0、国际 1、军事 2、警法 3、社会 4、公益 5、教育 6、财经 7、娱乐 8、文化 9
- url 为视频源地址

- video_id 为视频标识
- start_time 为选用的起始时间
- end_time 为选用的终止时间
- split 为人工标识为训练数据还是测试数据
- id 为视频编号

4 模型训练及测试

4.1 数据预处理

对于原始数据，我们首先在 prepro_feats.py 中读取.avi 格式视频，之后抽取部分帧进行特征提取，最后将提取出的特征保存在.npy 文件中。之后再 prepro_vocab.py 中读取.json 文件，提取其中的描述词进行词库构建，将得到的数据存储在 info.json 和 caption.json 中。

4.2 预处理具体参数

使用以下命令行进行数据预处理:

```
python prepro_feats.py --output_dir data/feats/resnet152
--model resnet152 --n_frame_steps 40  --gpu 0 --mode train

python prepro_feats.py --output_dir data/feats/test_feats/
--model resnet152 --n_frame_steps 40  --gpu 0 --mode test

python prepro_vocab.py
```

将 data/feats/resnet152 作为存储特征路径，选用 resnet152 作为处理模型，将视频提取的帧数设置为 40，并选用 gpu 进行数据处理，mode 使用 train 用于处理训练数据

将 data/feats/test_feats/作为存储特征路径，选用 resnet152 作为处理模型，将视频提取的帧数设置为 40，并选用 gpu 进行数据处理，mode 使用 test 用于处理测试数据

4.3 模型训练

训练模型时,我们使用 dataloader 进行数据加载,之后选用 S2VTAttModel 模型进行训练,得到训练好的模型之后,我们将相关参数保存至 opt_info.json 文件中

4.4 训练具体参数

```
python train.py --gpu 0 --epochs 80 --batch_size 300
--checkpoint_path data/save --feats_dir data/feats/resnet152
--model S2VTAttModel  --with_c3d 0 --dim_vid 2048
```

将模型置于 gpu 上进行训练，训练轮数设置为 80，批处理大小设为 300，模型参数保存在 data/save 中，特征路径为 data/feats/resnet152，模型选用 S2VTAttModel，不进行 3D 特征提取，同时根据视频的特征提取维数，将 dim 设置为 2048。

4.5 模型测试

使用 eval.py 文件对模型进行测试，将模型参数进行载入后，再测试集上进行视频描述，并保存相关描述结果

4.6 测试具体参数

```
python eval.py --recover_opt data/save/opt_info.json
--saved_model data/save/model_80.pth --batch_size 100
--gpu 0
```

选用训练了 80 轮的模型 model_80 同时设置批测试大小为 100

4.7 相关实验过程记录

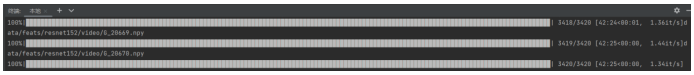


图 1: 训练集特征提取

图 2: 测试集特征提取

图 3: 词库构建

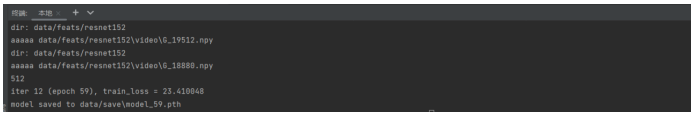


图 4: 训练过程

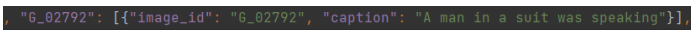


图 5: 测试视频描述

图 6: 视频画面

5 总结

在本次视频描述竞赛中，我们团队决定采用 S2VT 模型，并决定使用自行构建的数据集进行训练。这个选择并非轻率之举，我们深知面临着重重挑战。视频描述任务本身就是一个复杂的问题，需要模型能够理解视频内容，并将其转化为有意义的文字描述。而使用自建数据集，我们不仅要确保数据的多样性，还要确保数据的质量和准确性。

预处理测试集数据的过程充满了挑战。由于我们的数据集与参考代码所用的 MSR-VTT 数据集存在差异，我们无法直接使用参考代码的预处理方法。这意味着我们需要重投开始，自行编写代码对测试集数据进行适当的预处理。这是一个细致且繁琐的过程，最终我们成功利用自己编写的代码生成了 test_info.json 文件。在这个过程中，我们不断调整和优化预处理步骤，以确保数据的准确性和一致性。

在模型训练方面，我们同样遇到了困难。由于我们的数据集相对较小，我们担心模型可能会出现过拟合现象。为了解决这个问题，我们尝试了多种训练策略，包括使用不同的学习率、正则化方法等。经过多次试验和调整，我们发现将训练轮数设定在 80 轮左右能够获得较为理想的模型参数。在这个过程中，我们深刻体会到了超参数调整的重要性，以及选择合适的超参数对于提高模型泛化能力的影响。

通过这次实践，我们团队对视频描述任务有了更深入的理解。我们认识到，虽然每个人的知识和能力有限，但通过团队协作，我们可以充分发挥各自的优势，共同应对更具挑战性的任务。这次经历不仅让我们全面掌握了 RNN、LSTM 和 S2VT 等深度学习技术，还让我们对这些算法的思想和灵感来源有了更深入的了解。

同时，我们也意识到深度学习领域仍存在许多问题亟待解决。例如，如何有效利用无标签数据进行半监督学习、如何进一步提高模型的泛化能力

等。这些问题不仅极具挑战性，而且对于推动深度学习技术的发展具有重要意义。虽然我们团队目前尚未找到明确的解决方案，但我们已对这些前沿话题产生了浓厚的兴趣，并计划在未来的学习和研究中继续探索。

总之，这次视频描述竞赛为我们提供了一个难得的学习和实践机会。通过解决实际问题的过程，我们不仅提升了自身的技术水平，还培养了团队协作和解决问题的能力。这将对我们在未来深度学习领域的探索产生深远影响。我们深信，只有不断挑战自我、勇于探索未知领域，才能在深度学习的道路上取得更多的突破和创新。

References

[1] Subhashini Venugopalan et al. “Sequence to Sequence – Video to Text”. In: *arXiv e-prints*, arXiv:1505.00487 (May 2015), arXiv:1505.00487. DOI: 10.48550/arXiv.1505.00487. arXiv: 1505.00487 [cs.CV].