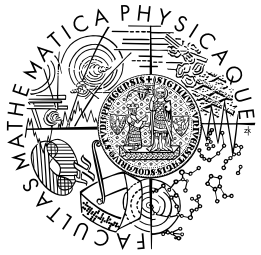


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Andrew McIsaac

Temporal Reasoning in Vision and Language Models

Institute of Formal and Applied Linguistics

Supervisor of the master thesis: doc. RNDr. Ondřej Bojar, Ph.D.
Prof. Raffaella Bernardi
Prof. Paolo Rota
Study programme: Computer Science
Study branch: Language Technologies and Computational Linguistics

Prague 2024

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

Thank you first, to my supervisors at the University of Trento, Prof. Raffaella Bernardi and Prof. Paolo Rota, for introducing this engaging topic to me, and helping to guide me through this thesis-writing process, and for providing me access to compute clusters, without which I would not have been able to do this work. Thank you also to my co-supervisor at Charles University, Doc. RNDr. Ondřej Bojar, and to all my professors who have taught me throughout this program. I would also like to thank the Erasmus Mundus Language and Communication Technologies program for generously providing me with a scholarship during my Master's, and for giving me the opportunity to travel and spend an extended period of time in two wonderful countries.

But the places wouldn't be anything without the people you meet, so thank you to the friends I have made along the way, for your support and kindness throughout. Finally, thank you to my family, who have supported me from afar over the last two years.

Title: Temporal Reasoning in Vision and Language Models

Author: Andrew McIsaac

Institute: Institute of Formal and Applied Linguistics

Supervisor: doc. RNDr. Ondřej Bojar, Ph.D.
Prof. Raffaella Bernardi
Prof. Paolo Rota, Institute of Formal and Applied Linguistics

Abstract: Are vision and language models able to reason across time? We evaluate the performance of vision and language models (VLMs) on the task of video question answering, with a particular focus on their temporal reasoning abilities. We probe the STAR video QA dataset on two VLMs with data perturbation methods of text and video inputs, and find that models are generally unable to identify the meaning of before and after in sequential questions. We then ask how a model can effectively learn these temporal relations, and design a new dataset drawn from videos and annotations from the Charades dataset. We create annotations that include targeted hard negative examples for the contrastive loss objective of one VLM, Merlot Reserve, such that the model must adapt to learn temporal relations. We further explore how to model fine-grained temporal relationships, and evaluate the benefits. We find that our approach shows promising signs of improvement on tasks that require temporal understanding, although it gains little sensitivity to temporal relations when probed.

Keywords: multimodal video LM temporal reasoning contrastive learning

Contents

Introduction

Research in vision and language models (VLMs) has bloomed over recent years. With larger and larger datasets and models, particularly based on the Transformer architecture (Vaswani et al., 2017), the performance and capabilities of VLMs have increased on common multimodal tasks such as visual question answering, image captioning, visual dialogue generation and image-text retrieval (Alayrac et al., 2022; Li et al., 2022, 2023a; Radford et al., 2021).

Video language models (vidLMs) are VLMs which are capable of modelling video. This provides the additional challenge of modelling long sequences of frames, and reasoning temporally across these images. Models must be able to recognise how a scene changes over time not only with respect to objects and relations between them, but they must also model the causal link between actions and events. For tasks such as video question answering, where a model is given a video and a question, and must pick the correct answer out of a number of multiple choice options, to answer questions such as "What did the man do after opening the door?", or "Why was the toddler crying at the end of the video?", it must be able to relate potentially distant events to one another and reason about them. Even if a model is able to select the correct option, how do we know that it has applied the correct reasoning steps required to make its prediction? Lei et al. (2023) and Buch et al. (2022) show that models trained with just a single frame can match or outperform the state of the art on multiple video and language tasks. This suggests that existing evaluation datasets have a "static appearance bias" (Lei et al., 2023) without challenging enough questions or options that would require event-level understanding to distinguish between them, and potentially that pre-training datasets and objectives are not incentivised to learn temporal information (Momeni et al., 2023). We explore whether current downstream datasets are able to show that temporal reasoning has been learned, for instances that should require temporal reasoning. VidLMs are often trained using contrastive learning, where the objective is to correctly match video-text pairs to each other, while repelling non-matching pairs in the joint embedding space.

In this thesis, we aim to understand the abilities of vidLMs to reason across time, and to propose a method for instilling a fine-grained temporal reasoning ability in such models. We design perturbation experiments to look at the temporal reasoning abilities of multiple video language models trained with a contrastive objective function. Following work that questions the ability of contrastive learning to pay attention to order structure in VLMs (Yuksekgonul et al., 2023), we ask a similar question of their ability to reason across time. Can vidLMs learn a grounded representation of temporal relations, especially "before" and "after"? Using questions which require sequential information from the STAR dataset (Wu

et al., 2021), a video question answering (video QA) dataset which tests situated reasoning questions in real-world videos, we find that these models do not learn to distinguish between actions occurring before or actions occurring after one another.

Following this finding, we ask how a model can be incentivised to learn to encode these grounded representations. We propose a method for learning temporal reasoning abilities, using targeted hard negatives in the contrastive objective to improve the model's understanding of temporal relations. We use videos from the Charades dataset (Sigurdsson et al., 2016), which has annotated events and their corresponding timespans, to create annotations with a temporal relation connecting a pair of actions, and hard negatives which modify the annotation in the temporal dimension only. For example, in a video that has the action annotations "someone is dressing" and "taking a cup from somewhere", with the first action occurring before the second, we generate the temporally-aware label "someone is dressing before taking a cup from somewhere". We then create hard negatives that modify the label in the temporal dimension (e.g. "someone is dressing *after* taking a cup from somewhere"), which is added to the batch of video-text pairs as a non-matching pair.

We evaluate our approach on multiple video QA datasets to test different types of temporal understanding and the generalisability of our approach. We also explore the effect of using a wider range of temporal relations than just before and after to model more fine-grained relations. We use relations from Allen's Interval Algebra (Allen, 1983), a calculus for reasoning about temporal intervals, which defines a range of temporal relation types that capture possible relations between actions. We find that using more fine-grained relations improves performance compared to both using just before and after, and on downstream tasks, although we find limited evidence that our models become more robust to temporal reasoning probing tests.

The rest of this thesis is organised as follows:

- Chapter 1 goes into the background of language models, image recognition models, and vision and language models (VLMs) which combine the two modalities. We discuss one popular method for training, Contrastive language-image pre-training (CLIP), and one key downstream task, visual question answering. We finish the chapter with a broad overview of video language models and an overview of the temporal reasoning literature in video and in language.

- Chapter 2 explores related work on video language models, with a particular focus on work that explores the impact of contrastive pre-training. We highlight previous work that has explored temporal reasoning in video language models.

- In Chapter 3 we look at the main datasets used, STAR and NExT-QA, as well as the particular pre-trained models that we work on.
- In Chapter 4 we test the current temporal reasoning abilities of multiple video language models on the STAR dataset.
- Chapter 5 details experiments that show how current models perform on temporal reasoning tasks, and describes our approach to generating additional hard negatives focussing on temporal words for contrastive training.
- Chapter 6 shows performance of our model on STAR and NExT-QA. We evaluate different design decisions in our dataset, and compare our approach to related work.
- Finally, the Conclusion summarises our findings and discusses the use of contrastive pre-training methods in video language models for temporal reasoning.

1. Background

In this chapter, we briefly cover progress in obtaining useful representations of language (Section 1.1), images (Section 1.2), and efforts to combine these two modalities (Section 1.3). We then look at the extension of vision and language models to videos (Section 1.4), which introduces the extra complexity of reasoning across sequences of images, and optionally adding a further modality, audio. Finally, we explore the literature on temporal reasoning in language and in vision (Section 1.5).

1.1 Language Modeling

Language modeling is the task of predicting the next word given some number of previous words. A neural language model (Bengio et al., 2003) performs this by receiving as input to a feedforward neural network a representation of previous words in a sequence and outputting a probability distribution over possible words. The probability of a sequence of T words w_1^T is thus the combined probability of all words given their context:

$$\hat{P}(w_1^T) = \prod_{t=1}^{T} \hat{P}(w_t|w_1^{t-1}).$$

The conditional probability can be approximated by using a fixed context length N,

$$\hat{P}(w_t|w_1^{t-1}) \approx \hat{P}(w_t|w_{t-N+1}^{t-1}),$$

greatly reducing the computational requirements for longer sequences. This section summarises common approaches to modelling text sequences.

1.1.1 Recurrent Neural Networks

The recurrent neural network (RNN) takes individual items from a sequence, one at a time, and outputs a prediction based on the single unit and a hidden state. The hidden state is a recursive unit learnt from previous hidden states, so that at timestep t, the hidden state h_t is a combination of the previous hidden state h_{t-1} and the current input x_t. The hidden state is therefore a representation of the entire input sequence up to time t. This avoids the problem faced by feedforward neural language models of only representing a limited context window of size N. In theory, an RNN can represent an unlimited context.

In practice, RNNs struggle to encode long-distance dependencies well, with the information encoded in hidden states being biased towards more recent items of

the input, and struggling from the vanishing gradient problem, whereby repeated matrix multiplications for backpropagation through time drive the gradient to zero. The long short-term memory network (LSTM) (Hochreiter and Schmidhuber, 1997) was proposed to extend the RNN by modifying the architecture of the recurrent unit to include three gates: the forget gate, the add gate, and the output gate. Combined, these three gates keep the context vector, the previous hidden state, simple by removing information considered no longer useful, add useful information from the current input, and output information considered useful for the current hidden state.

RNNs are often used in sequence-to-sequence, or encoder-decoder, set-ups, in which the input sequence is processed by the encoder section, creating a context vector which is a representation of the entire input sequence. This is then fed as the initial hidden state of a decoder network to generate the output. The benefit of this is that the output size is not related to the input size. For tasks such as machine translation, image captioning, or open-ended question answering, the ability to generate an answer is critical.

The bottleneck problem is alleviated slightly by the attention mechanism (Bahdanau et al., 2015), where an additional context vector is used by the decoder to dynamically attend to different hidden states of the encoder based on the current input token in the decoder. This context vector is created by a weighted sum of encoder hidden states, recomputed at each timestep during decoding. Attention with RNNs improved the state of the art in machine translation, particularly on sentences with longer input. It has also been used in vision and language models (VLMs) to attend to key parts of the image vector for visual question answering (Yang et al., 2016).

1.1.2 Transformer

Vaswani et al. (2017) introduced the Transformer architecture for sequence tasks, replacing the recurrent nature of the RNN and its variants with multi-head self-attention. This allows for parallel computation since computation at each timestep is independent of all others, greatly increasing the ability to train on larger and larger data and model sizes. Self-attention assigns attention scores to each item of the input sequence itself, regardless of input size, to compute a representation of the sequence. The Transformer uses stacked layers of self-attention to capture the many ways that an input sequence can relate to itself. Each self-attention head can learn to encode different relationships between sequence tokens, and these heads are combined and linearly projected into the original dimensionality. Vaswani et al. (2017) use attention between the encoder and decoder, so each position in the decoder can attend to all items of the input sequence, and further use self-attention in both the encoder and decoder. In the decoder, a modification is

made to prevent knowledge of future information being generated, masking out all values in the input that correspond to future input connections. Finally, positional encodings are included for each token to keep some notion of sequence order that would otherwise be lost from the RNN architecture.

The Transformer achieved state of the art performance on machine translation, and has since been used as the de facto architecture for many sequence tasks, in both language and vision. It can be trained in an encoder-decoder setting, or the two parts can be separated to train only the encoder (e.g. BERT (Devlin et al., 2019)), or for text generation using only the decoder with a simple language modelling objective (e.g. GPT-3 (Brown et al., 2020)). Its ability to scale to larger dataset and parameter sizes has led to massive improvement in zero-shot and few-shot ability on a wide range of downstream tasks (Hoffmann et al., 2022).

1.1.3 Masked Language Modeling

BERT (Devlin et al., 2019) uses only the encoder layers of the Transformer to create strong representations of an input sequence. Since it is trivial to predict the next token in a sequence when provided with the entire context, BERT, and its descendants such as RoBERTa (Liu et al., 2020), train on a masked language modeling objective on unlabeled data, where the task is to mask some percentage of the input tokens at random, and then predict those masked tokens.

These representations on their own are not especially useful for common tasks, but once trained, they provide a great starting point for finetuning to a specific task, where there may not otherwise be enough data to learn these rich representations of language. Downstream tasks can include question answering, natural language inference, or sentence classification. Pre-training then finetuning has become a common paradigm due to the relative low cost of finetuning once a large model has been pre-trained. BERT achieved state of the art on eleven NLP tasks, all of which were finetuned in less than an hour on a TPU. BERT has successfully been adapted to vision and language models, as we will discuss in Section 1.3.

1.2 Image Recognition

A key part of video and language models is learning representations of frames in sequence, which involves the classical tasks of object detection, image segmentation, and image classification. Much like in NLP, the standard approach is to pre-train on large image datasets and finetune to a specific desired task. Convolutional neural networks (LeCun et al., 1989; Krizhevsky et al., 2012; He et al., 2016) use convolution kernels, pooling, and optionally batch normalisation, dense or residual layers to create representations of image features. AlexNet (Krizhevsky

et al., 2012) uses a multi-layer convolutional neural network (CNN) to classify images from ImageNet (Deng et al., 2009), a dataset of over 15 million images from around 22,000 categories, and was the first to show the scaling power of large datasets and model sizes for producing strong image features.

There have been attempts to combine CNN architectures with self-attention mechanisms. This may provide more scope for non-local computation which may be required on tasks such as object detection with large objects. However, due to the quadratic cost of self-attention in the number of pixels, naive implementations are infeasible, and approximations struggle to scale efficiently (Carion et al., 2020). The Vision Transformer (Dosovitskiy et al., 2021) does away with the CNN for image recognition, and uses an adapted version of the Transformer for greater scalability.

1.2.1 Vision Transformer

Dosovitskiy et al. (2021) introduced the Vision Transformer (ViT), which takes the impressive performance of the Transformer architecture on sequence tasks and applies it to image tasks. The authors represent an image as a sequence of patches of an image, with an extra patch embedding added alongside the positional embedding of the Transformer to maintain the 2-dimensional information of an image when projected into a linear sequence. The model is shown in Fig. 1.1. The ViT matched or exceeded state of the art on many image classification datasets, while being trained for comparatively less time. As we discuss in Sections 1.2.2 and 3.3, it has been used as the visual encoder for multiple multimodal models due to its scaling ability (Zhai et al., 2022).

1.2.2 CLIP

Radford et al. (2021) introduced CLIP (Contrastive Language-Image Pre-training), which uses the Info-NCE loss (van den Oord et al., 2019) to jointly learn relationships between encodings of text captions and extracted feature representations of associated images. The Info-NCE loss trains a multimodal embedding space to maximise the cosine similarity of matching pairs of captions and images, while minimising the cosine similarity of non-matching pairs in the batch. The approach is shown in Fig. 1.2. The Info-NCE loss is a symmetric cross-entropy loss, defined

$$\mathcal{L} = -\sum_{(i,t)\in\mathcal{B}} \left(\log \mathrm{NCE}(z_i,\ z_t) + \log \mathrm{NCE}(z_t,\ z_i)\right),$$

where NCE is the normalised cross entropy

$$\mathrm{NCE}(z_i, z_t) = \frac{\exp(z_i \cdot z_t^+)}{\sum_{z\in\{z_t^+, z_t^-\}} \exp(z_v \cdot z)}$$

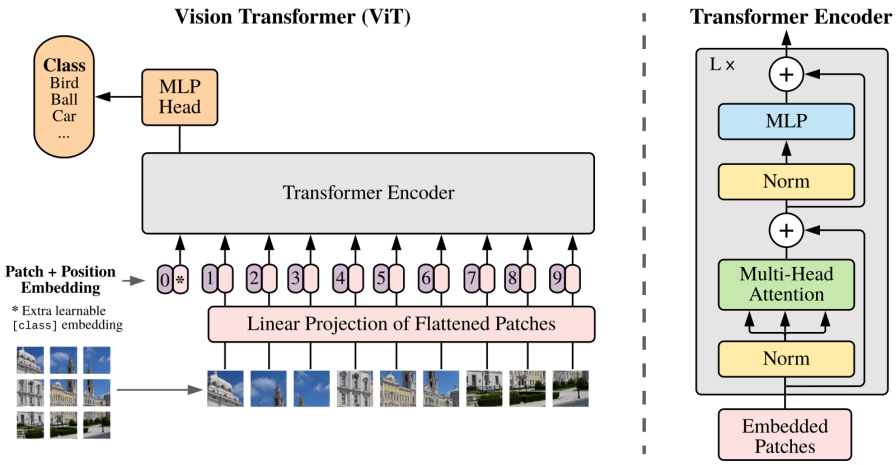


Figure 1.1: The Vision Transformer splits an image into patches, embeds them, and feeds them into a Transformer encoder. Classification is learned via an MLP head following the encoder. Figure reproduced from Dosovitskiy et al. (2021)

for positive caption z_t^+ matched with image z_i, while $\{z_t^-\}$ are the negative captions from the batch $\mathcal{B}$.

Part of this approach is to use a very large batch, so that there are many incorrect pairings to learn from. Radford et al. (2021) use a batch size of 32768. The text encoder is a Transformer (Vaswani et al., 2017), and their best model uses a Vision Transformer (Dosovitskiy et al., 2021) as the image encoder.

The model enables zero-shot transfer to many downstream computer vision classification tasks by predicting the most probable (image, text) pair when given an image and a set of text prompts with each class embedded in the prompt achieving performance comparable to or surpassing the previous state of the art by finetuned models. The representations learned by the contrastive pre-training objective have wide applicability to a range of VLMs and video language models (vidLMs), particularly as frozen features from which to add smaller modules on top for adapting to vision and language tasks (Alayrac et al., 2022; Lin et al., 2022; Luo et al., 2022). We discuss some of these models in Section 1.4, and consider the limitations and possible expansions of the contrastive pre-training method in Section 2.1.

1.3 Vision and Language Models

One criticism of large language models (LLMs) is that the representations learned by training a model to predict the next word fail to learn any kind of meaning

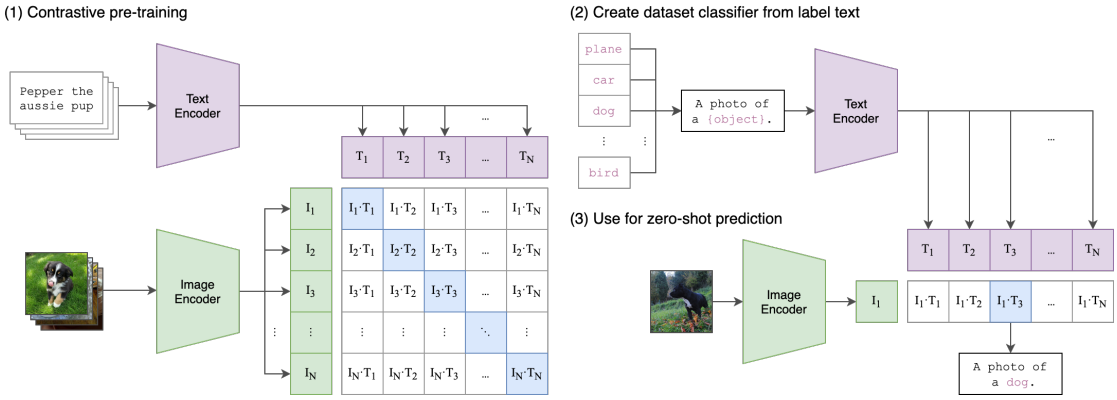


Figure 1.2: CLIP. Given a batch of image-text pairs, the pre-training objective matches correct pairs, while minimising similarity of non-matching pairs. This representation can be used for a range of downstream tasks. From Radford et al. (2021)

without reference to the real world (Bender and Koller, 2020). Models trained in this way learn connections between surface forms, but no grounded meaning between the form and intent portrayed through the form. A way to create grounded representations may be to combine the two modalities of language and vision through multimodal embeddings. A key challenge in recent years has been to find suitable methods for creating these shared representations.

One approach follows the strong performance of BERT (Devlin et al., 2019) in NLP tasks. Li et al. (2019) extend BERT to include visual features extracted from a CNN as well as text tokens as input to a Transformer encoder, implicitly discovering a joint representation between the two modalities. The authors use the self-attention mechanism to align elements of the input text and regions in the input image, and pre-train on two visually-grounded language model objectives. The authors finetune and evaluate on a range of vision and language applications, including visual question answering (VQA, see Section 1.3.1) and visual commonsense reasoning, which extends VQA to ask the model to also predict a rationale given a question and answer pair. This simple method provided encouraging results on these tasks.

Other works use novel ways of fusing visual information into the language model. Yu et al. (2022) use cross-attention layers in a text decoder to train a captioning loss with pooled features from the image encoder, combined with a contrastive loss between image and text features. Li et al. (2023a) bootstrap a VLM from separate frozen models for vision and for language, with a trainable Querying Transformer that extracts a fixed number of visual features from the vision encoder and optimises to extract features that most align with the text

caption associated with the image. The output of the Querying Transformer is then trained alongside a frozen LLM to return interpretable visual features as a prefix to a language model, acting as an information bottleneck for downstream use by the LLM. One of the downstream tasks is visual question answering.

1.3.1 Visual Question Answering

Visual question answering (VQA) is the task of answering questions given text and an image. There are a range of datasets that test different aspects of image and text understanding (e.g. Johnson et al. (2017); Hudson and Manning (2019); Antol et al. (2015)). A challenge in creating datasets is to ensure that there are few statistical biases or shortcuts in the answer distribution that a model can exploit without a true understanding of the scene. For example, models trained on the VQA dataset (Antol et al., 2015) were found to make predictions based on overly strong language priors without considering the associated image (Zhang et al., 2016) (a green banana may trip up a model) and failed to show complete question understanding, settling on an answer before receiving the full question (Agrawal et al., 2016). Questions generally required little reasoning or compositionality, with many answers achievable solely by object recognition (Hudson and Manning, 2019). The GQA dataset (Hudson and Manning, 2019) is one dataset that aims to limit these issues by generating questions with linguistic diversity and a large vocabulary, and balancing the answer distribution through sampling.

Visual question answering has also been extended to question answering over videos. On top of scene understanding, video question answering requires event understanding to understand causal and temporal relationships within the context of the video. A particular challenge in developing models for this task is combining and aligning the modalities of text, vision, and audio as well. Several datasets have been proposed for this task (Xu et al., 2016; Wu et al., 2021; Xiao et al., 2021; Lei et al., 2020). We discuss two which we study and evaluate in our experiments, STAR (Wu et al., 2021) and NExT-QA (Xiao et al., 2021), in Chapter 3.

1.4 Video Language Models

Much as the task of visual question answering has been extended to the domain of video, so too have models been created for video language tasks. Video language models are models used to solve problems related to video understanding tasks. This introduces the added complexity of temporal modeling to understand relationships between successive frames in the video, as well as the possibility of modeling audio where the data allows it. There have been two main approaches to solving these tasks. The first is to adapt pre-trained vision and language models as

seen in Section 1.3 to the new domain without any specific training or finetuning on a video dataset. This can benefit from the large amount of research into these models, with huge pre-trained and highly performant models readily available, although there is a challenge to adapt to the domain shift and new challenges posed by videos. Alternatively, we can train on a video dataset, either from scratch, or finetuning from a pre-trained vision and language model. This section explores both methods.

1.4.1 Adapted from Vision and Language Models

Pre-trained generative large language models (LLMs) have shown strong capabilities for in-context learning (Brown et al., 2020). In-context learning provides a number of examples of a task (for few-shot learning – zero-shot learning provides only a task description) as the start of a prompt to a language model, where a typical example contains the context of the task and its desired completion. The language model must then provide the correct completion when presented with just the example context. This idea, and extensions, have been shown to be effective for a wide range of tasks, particularly those which require advanced reasoning (Wei et al., 2022; Kojima et al., 2022).

By providing generative LLMs with access to image features in its prompt, it is possible to leverage pre-trained LLMs for video tasks. Wang et al. (2022) use a vision and language model to label objects, events and attributes, as well as captions, for each sampled frame in a video. These features are then composed in a template for few-shot learning of video tasks. Temporal relationships between frames are modeled in the template using textual indicators ('first', 'then', 'finally'). Crucially, no finetuning of language or vision and language models is performed, so high quality pre-trained models can be plugged in and changed easily. This process is highly dependent on strong visual feature extraction, meaning that key low-level features may be lost if the vision models are not strong enough. Concurrent work by Zeng et al. (2023) finds that using stronger vision and language models correlates with better performance when combining VLMs and language models in a zero-shot manner for egocentric perception, where videos are shot from a first-person perspective.

Portillo-Quintero et al. (2021) use CLIP features with an aggregation function across frames to adapt to the video domain for retrieval tasks. The best aggregation function tested was to simply average frame-level features, which beat previous best recall@1 scores on the MSR-VTT (Xu et al., 2016) dataset, despite the lack of relative temporal awareness by mean pooling features from multiple frames. The authors found that using a single frame from around one second in to the video as the aggregation function gives significantly worse recall performance than other aggregation functions which take into account multiple frames.

By contrast, Huang et al. (2018) found that temporal understanding plays just a small role in multiple video datasets. On two action recognition datasets, the impact of motion accounts for just 6 percentage points of 79% accuracy on UCF101 (Soomro et al., 2012), and 5 points of 47% accuracy on Kinetics (Carreira et al., 2018), and 40% and 35% of classes do not require any temporal understanding for the two datasets respectively. Buch et al. (2022) extend this finding for video language tasks, with single frame understanding performing strongly compared to state of the art models, "even in settings intended for complex multi-frame event understanding". The key distinction between these works and Portillo-Quintero et al. (2021) is that the model selects a highly informative frame based on its task. Similarly, Lei et al. (2023) find that training on a randomly chosen single-frame and only providing visual features from multiple frames at inference time achieves strong performance on existing datasets, which have a bias towards static appearance. Buch et al. (2022) propose that their design, the atemporal probe (ATP), be used to design better datasets that better test efficacy of a benchmark for causal and temporal understanding. They find a subset of NExT-QA (Xiao et al., 2021) questions that "truly necessitate video-level understanding compared with the original dataset". We test on both NExT-QA and this subset, denoted ATP_{hard}, in our experiments.

1.4.2 Training on Videos

This section mainly explores models pre-trained with a contrastive objective, since we are predominantly concerned with how models trained with this choice of objective function, popular for vidLMs, are able to learn temporal reasoning. We note, however, that other models (Lei et al., 2021; Xu et al., 2021a; Alayrac et al., 2022) have achieved comparable downstream performance when trained with other objectives (masked language modeling, image-text matching, cross-modal attention).

The obvious approach for training video language models is to train on videos. Luo et al. (2022) extend vision and language models to video retrieval in a simple way by mapping sequences of image representations learned from CLIP into a fixed video representation, and computing similarity between the CLIP text encoding and the learned video encoding. They find that training on a medium-sized video dataset starting from the CLIP encodings improves zero-shot and finetuning results on multiple downstream datasets, and that attempts to model the temporal dependency between frames (using 3D linear projections for video features, and using similarity measurements that model sequentiality for the video and text similarity measure) from the base CLIP model trained only on image and text pairs do not produce better results on video tasks.

Bain et al. (2021) learn a separate visual encoder for images and videos, and a

text encoder for captions for video retrieval. Visual features are used as input to a space-time Transformer encoder, which, when projected into a common video-text space, is contrastively compared to the encoded text features. The authors use curriculum learning to learn the temporal information of videos by increasing the number of frames provided to the model during training. They find that training on a single frame is not enough for retrieval, and that progressively increasing the number of frames (up to 8 frames) during pre-training can result in better performance than training with more frames to start with.

Xu et al. (2021b) train a video understanding model using a contrastive objective with loosely temporally aligned videos and text transcriptions. The authors note that strict alignment between transcript and video clips can result in low relevance pairings. The authors empirically find that loosening the constraint between video and text clip timestamps to have loosely overlapping pairs provides a better association between video and text pairs. The authors also create batches based on semantically similar video/text embeddings, creating hard negative examples to strengthen the learned embeddings. The retrieval process is intertwined with the training process, so that as the joint embedding space is learned harder batch videos can be retrieved.

Finally, Zellers et al. (2022) uses a contrastive span objective, where videos, speech and their subtitles are aligned in short time spans, and the model must predict masked out text and audio spans given a frame and the surrounding context. It is able to scale to loosely aligned datasets, and outperforms even some finetuned models in a zero-shot setting on the STAR (Wu et al., 2021) benchmark. We use this model as the basis for our experiments, and discuss the full architecture in Section 3.3.

1.5 Temporal Reasoning

We finish this chapter with an overview of how temporal reasoning is defined in video and in language, and datasets used to explore the abilities of models in this direction.

1.5.1 In Video

Most computer vision has studied how to model concepts and relationships between them in the world, e.g. through object detection and segmentation in images. To go one step further into the video domain, we need to study how to model event knowledge. That is, how do we model recurring and meaningful patterns and sequences of behaviour? A model must understand activities and their components, as well as the temporal ordering of these activities to find causal dynamics

between them (Elman and McRae, 2019). New datasets and models have been proposed in recent years that aim to find computational models capable of this event knowledge through temporal reasoning.

As discussed in Section 1.4.1, some models can still perform well on video datasets with just a single frame given to the model. This suggests a requirement for more challenging datasets and tasks to learn temporal ordering of events. Grauman et al. (2022) create Ego4D, a dataset of over 3000 hours of egocentric (first-person) video, with several associated tasks requiring understanding of how objects change state over time, remembering temporal windows for objects appearing in scenes, and prediction of future actions in videos, requiring causal understanding of actions and events. For example, a cooking video may predict the subsequent steps to making a pizza when presented with the first steps of rolling and kneading dough. The authors identify normalised pointwise mutual information as a means to inform the temporal structure of sequences of actions over time, with certain action sequences favoured over others. Learning this structure of action pairs is key to a model's performance on these tasks.

1.5.2 In Language

There is a long history of studying temporal expressions in linguistics. Moens and Steedman (1988) claim that *when*-clauses (e.g. "When they built the 39th Street bridge, they solved most of their traffic problems") are not primarily temporal, but "establish a temporal focus" between two events, contingent on e.g. a causal link, as in the unnatural use of *when* in "*When my car broke down, the sun set." They argue that any representation looking to accurately model temporal descriptions must therefore, for *when*-clauses and similar phenomena, model contingency, a dependency between events in time, rather than just the sequential ordering in time of such events.

Allen (1983) suggests a model of temporal reasoning based on intervals and relations among them. Given two events, the temporal relations between them can be expressed in many ways based on the time intervals of the events occurring. We explore the use of this temporal representation further in Section 5.1. Zhou et al. (2021) propose a dataset for natural language inference of temporal relations for before and after relations. They find that current models struggled to predict temporal relationships between explicit and implicit events, and that a neuro-symbolic method improved reasoning ability by estimating event durations to infer implicit end times.

1.5.3 In Video and Language

Different models have different approaches to modelling temporal awareness for videos, even within the contrastive pre-training approach. Merlot Reserve (Zellers et al., 2022) relies on an alignment between subtitles, audio, and video frames to keep consistent temporal awareness with the contrastive span objective, but its architecture lacks a specific module for temporal reasoning. Similarly, VideoCLIP (Xu et al., 2021b) relies on temporally overlapping video-text pairs to train its contrastive objective function with an otherwise simple architecture.

In contrast, other models explicitly include temporal modules in the architecture, either through learnt temporal positional encodings (Alayrac et al., 2022), 3D linear projections of video features with patches to the ViT video encoder a 3D kernel of multiple frames , cross-attention between frames (Li et al., 2023b) or combinations thereof (Lin et al., 2022; Bain et al., 2021). Lin et al. (2022) find that the effect of temporal information varies greatly depending on the dataset, and analysis on Something-Something-V2 (Goyal et al., 2017), which relies more heavily on temporal information, shows that including a single temporal module improves accuracy by over 10%, while combining sources of temporal information provides marginal extra performance gain.

1.5.4 Probing Video Datasets

Sevilla-Lara et al. (2021) create a perceptual test to discover action classes in videos that require temporal information to identify. The authors shuffle frames in time from action classification datasets, and present human annotators with the shuffled or control videos, where there is no shuffling. Action classes are then identified by the largest average performance degradation of action classification between the two groups. They train video models on a temporal and static dataset, the 50 classes where human accuracy decreases most and least respectively, and find that training on the temporal dataset produces features that are more sensitive to temporal ordering, and therefore are stronger temporal features. We extend this finding and explore the performance of various models with frames shuffled on video question answering datasets in Chapter 4, and develop a novel method for training video language models on a temporal-aware dataset in Chapter 5.

2. Related Work

This chapter looks at previous work on probing vision and language models, and techniques for improving reasoning in various directions in vision and language models. We use and extend the approaches explored to try and improve the temporal reasoning abilities of video language models.

2.1 Contrastive Training in VLMs

Some previous studies have looked at the effect of contrastive pre-training in vision and language models, and introduce the idea of post-pretraining VLMs with hard negatives (Yuksekgonul et al., 2023; Momeni et al., 2023; Bagad et al., 2023). Post-pretraining is a continuation of self-supervised pre-training on a smaller dataset with desired properties that aid the learning process of the model, mitigating the cost of expensive general pre-training while allowing for specialisation of a model. This can be used for, e.g. transferring VLMs to the video domain with a small video dataset, as in Luo et al. (2022), discussed in Section 1.4.2. Or, as we discuss in this chapter, instilling better understanding of concepts and relationships with targeted hard negatives in a contrastive objective. Yuksekgonul et al. (2023) explore compositional relationships in vision and language models by testing existing VLMs on a dataset with perturbations exploring attributive understanding of adjectives to nouns, relational understanding for prepositions and verbs, and sensitivity to word order in image captions. When presented with an original caption and its transformation(s), models must predict which caption is more likely. The authors find that most models are deficient in relational understanding tasks (e.g. choosing between 'the horse is eating the grass' and 'the grass is eating the horse'), but are better at attribution of properties to objects, as in 'the paved road and the white house' vs 'the white road and the paved house'. Models also performed close to chance on the word order sensitivity test, where multiple extra captions were created with shuffled nouns/adjectives, shuffled trigrams, and shuffled words within each trigram, indicating the VLMs behave like bags-of-words.

The authors claim that this may be down to the contrastive pre-training objective in VLMs such as CLIP (Radford et al., 2021), where the retrieval nature of the objective leads to a bias towards object recognition without considering compositionality, and that in datasets without carefully constructed caption alternatives, order information is not required to solve the objective. An incentive, in the form of additional hard negatives in both alternative images and generated targeted captions, is therefore proposed (Fig. 2.1), which improves performance on the testing benchmarks for attribution (from 62% to 71%), relation (63% to

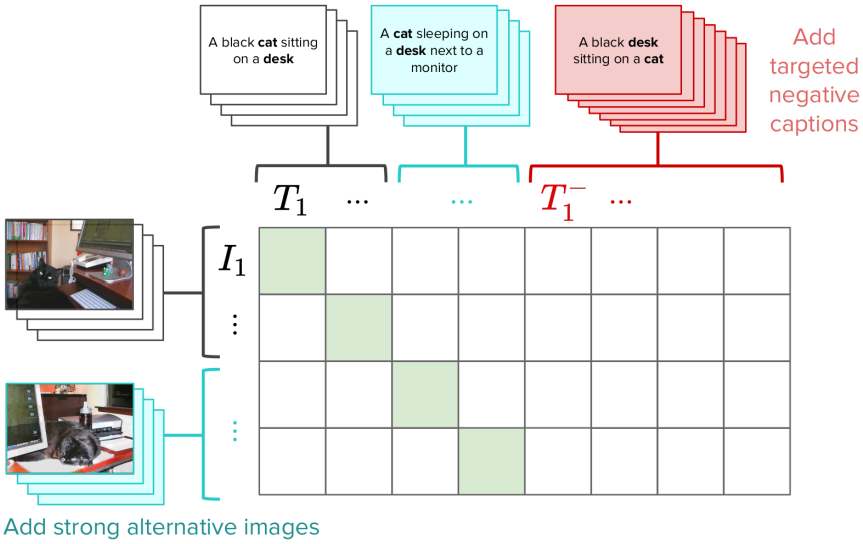


Figure 2.1: Hard negatives for contrastive learning, with generated negative captions and retrieved alternative images. Captions are generated by swapping various linguistic features, while images are sampled from k-nearest neighbours. From Yuksekgonul et al. (2023).

81%), and order (46% to 86% and 59% to 91%) substantially, while not degrading performance in other downstream tasks.

2.2 Understanding in Video Language Models

Here we discuss two papers that look at improving understanding in video language models (vidLMs) by extending the contrastive objectives with hard negatives for verb understanding (Momeni et al., 2023), and in before/after relations (Bagad et al., 2023).

2.2.1 Verbs in Action

Momeni et al. (2023) look at vidLMs trained with a contrastive loss function, and find similar issues with verb understanding to those identified by Yuksekgonul et al. (2023). They propose to generate hard negatives with modified verb phrases using pre-trained large language models (LLMs), as well as introducing an additional verb phrase alignment loss which contrastively compare a verb phrase from the positive caption to other verb phrases in the batch to provide an additional focus on verbs to the model (see Fig. 2.2). As in Yuksekgonul et al. (2023), models

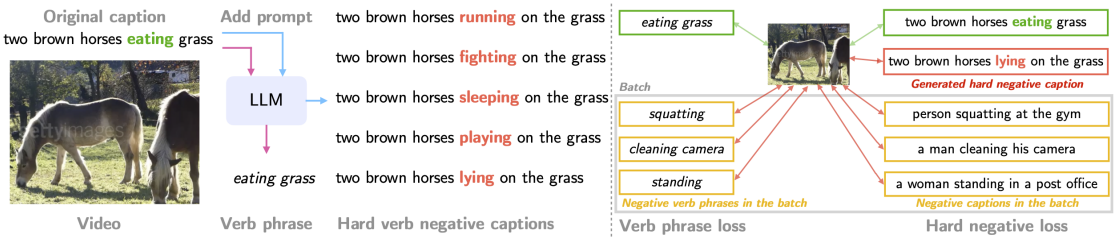


Figure 2.2: Verb-Focused Contrastive learning. Generated negative captions are added as hard negatives to the contrastive loss objective. From Momeni et al. (2023).

trained with targeted alternatives improve performance for datasets that require understanding of the targeted domain in both zero-shot and finetuning setups.

2.2.2 Test of Time

The most similar paper to our work is Bagad et al. (2023). The authors look at before/after relations in videos by using a synthetic dataset to probe existing models. They construct videos containing pairs of events such as "a red circle appears before a yellow circle", and create distractor annotations by reversing the order of events but keeping the temporal relation the same. On this time-order probing task, they find that existing models perform no better than chance for the task of associating the correct annotation to the video. They describe a post-pretraining strategy, Temporal Adaptation by Consistency of Time-order (TACT), for improving the understanding of before/after relations in vidLMs, where non-overlapping video clips are stitched together and paired with a text description that consists of the clip captions and a temporal relation, either before or after, to match the order of the combined video clip (see Fig. 2.3). A reversal function is then applied to create hard negative examples by reversing the order of text captions or video clips, which are included as negatives in the contrastive post-pretraining objective.

They find that this approach improves performance on the time-order probing task, with models much more likely to match the correct caption to the stitched video clips. On downstream tasks, they find a mixed result. On video retrieval tasks, on which they claim existing datasets have more of a bias towards spatial understanding than temporal reasoning (see Section 1.4.1; Buch et al. (2022); Lei et al. (2023); Luo et al. (2022)), the model performs slightly worse in general than without using the TACT approach. On video question answering, with temporally challenging datasets, there are generally slight improvements. For example on the NExT-QA ATP_{hard} subset, the TACT model trained on clips from TEMPO (Hen-

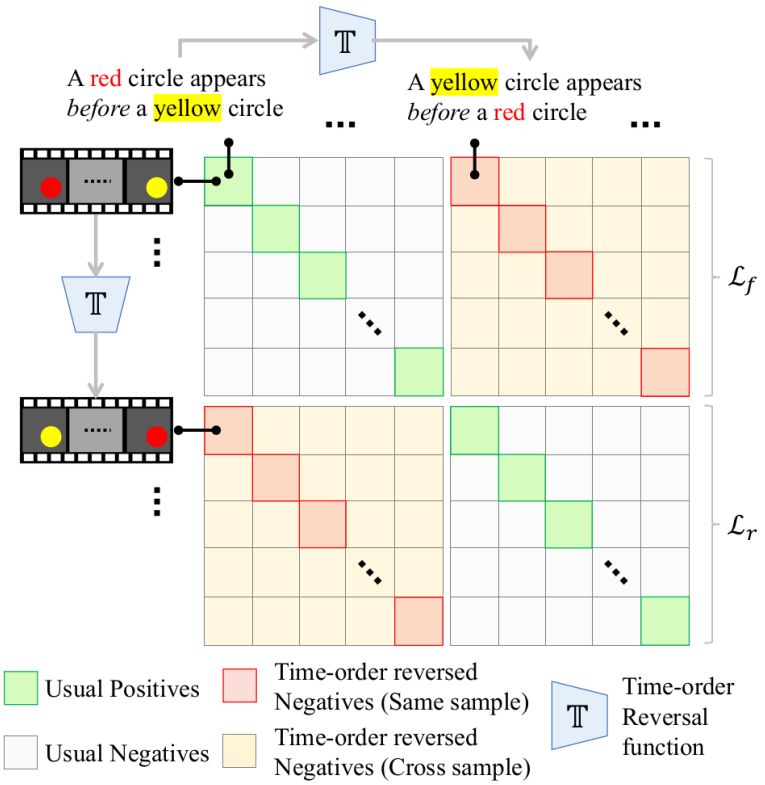


Figure 2.3: Temporal Adaptation by Consistency of Time-order. Extra negatives are included by reversing time-order in annotations and videos. Figure from Bagad et al. (2023).

dricks et al., 2018) achieves a zero-shot accuracy of 27.6, compared to 25.0 on the baseline model. However, on clips from the Charades dataset (Sigurdsson et al., 2016) the TACT performance is worse than the Charades baseline model (25.2 vs 26.0). They find that the TACT model generally improves performance on action recognition benchmark subsets which have been identified as requiring temporal information.

In comparison to Bagad et al. (2023), we explore different ways of probing temporal understanding, use a wider range of temporal relations with full videos, and aim to gain a stronger relationship between frame and action with the contrastive span objective. We provide a full comparison of our approaches and results in Section 6.5. We now go on to describe the datasets and models that we use in our approach.

3. Datasets and Models

This chapter describes the existing datasets and models used in our experiments in detail. We first look at two video question answering datasets, STAR (Wu et al., 2021) and NExT-QA (Xiao et al., 2021), and then discuss the model we adapt for use in our experiments, Merlot Reserve (Zellers et al., 2022). STAR is tested zero-shot in Merlot Reserve, and achieves state of the art performance. In Chapters 5 and 6 we test how we can modify this model to improve its temporal reasoning ability, while keeping strong performance on the STAR dataset. We further use NExT-QA to test the generalisability of our model.

3.1 STAR

STAR (Wu et al., 2021) is a dataset for situated reasoning in real-world videos. It uses videos taken from the Charades (Sigurdsson et al., 2016) dataset, which describe daily life actions or activities in indoor scenes. A video is annotated with actions and timestamps. STAR builds a detailed scene annotation from these videos. A situation is a description of entities, events, movements, and environments. An example is shown in Fig. 3.1.

There are four types of question: interaction, sequence, prediction, and feasibility. Based on the type of question, a situation will include complete action segments or, for prediction and feasibility questions, involve actions involved in the questions and an incomplete action segment about answers. Answers are generated to provide three different distractors along with the correct answer. The compositional distractor satisfies verb-object compositionality and is generated so as to be feasible in the same situation. The random option is selected from other instances, with the constraint that compositionality is satisfied, while the frequent option selects the most frequently occurring answer in each type of question group to deceive models that look for shortcuts in this way.

With respect to temporal reasoning, of particular interest to us are the sequence questions. These are questions which evaluate the temporal reasoning of systems when facing consecutive actions in dynamic situations, and ask about relationships between people and objects through the actions they perform in a situation. In the example shown in Fig. 3.1, given the sequence question "What object did the person take after the person put down the bottle?" and four multiple choice options, a model must predict the correct answer, "The book".

The best baseline model achieves an average accuracy of 36.7% across all question types. In their baseline results, the authors note that visual perception has a significant impact on situated reasoning. Existing vision models struggle to reason

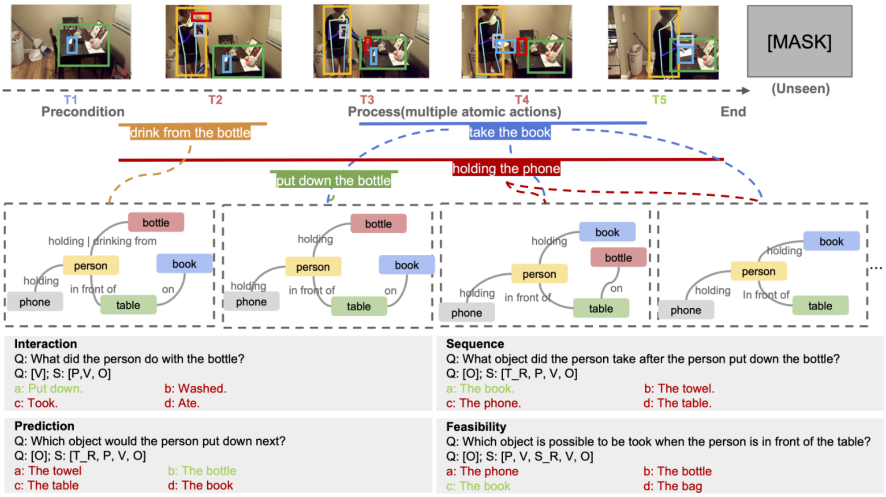


Figure 3.1: An example instance from the STAR dataset, with the four question types. Image from Wu et al. (2021)

well in real-world situations, and models struggle more to identify relationships between objects than objects themselves. It is therefore the task of an improved video and language model to better realise these relationships. The authors note that the dataset requires multimodal modeling and cannot be solved by a language-only model, with BERT achieving an average of 31.5% accuracy. The relatively low scores with a chance level of 25% (based on four multiple choice questions) indicates that there is a lot of room for models to improve on this benchmark.

3.2 NExT-QA

NExT-QA (Xiao et al., 2021) is another video question answering dataset with a focus on a wider range of temporal actions. Where STAR asks questions that test only before/after temporal relations, questions in NExT-QA challenge the model to reason about causal actions as well as temporal relations such as 'when' (Fig. 3.2). Videos are sourced from YFCC-100M (Thomee et al., 2016), which contains diverse videos portraying real-life actions and events from Flickr. Charades videos, mainly filmed inside and depicting one person doing a single activity, are limited in their domain. To test the generalisability of our approach to different domains, YFCC-100M videos provide a greater variety of scenes and actors in the dataset, alongside the greater range of temporal relations in NExT-QA questions. Buch et al. (2022) introduce a subset of NExT-QA, dubbed ATP_{hard}, which is NExT-QA questions that are challenging in the temporal dimension. We test on this subset as well.

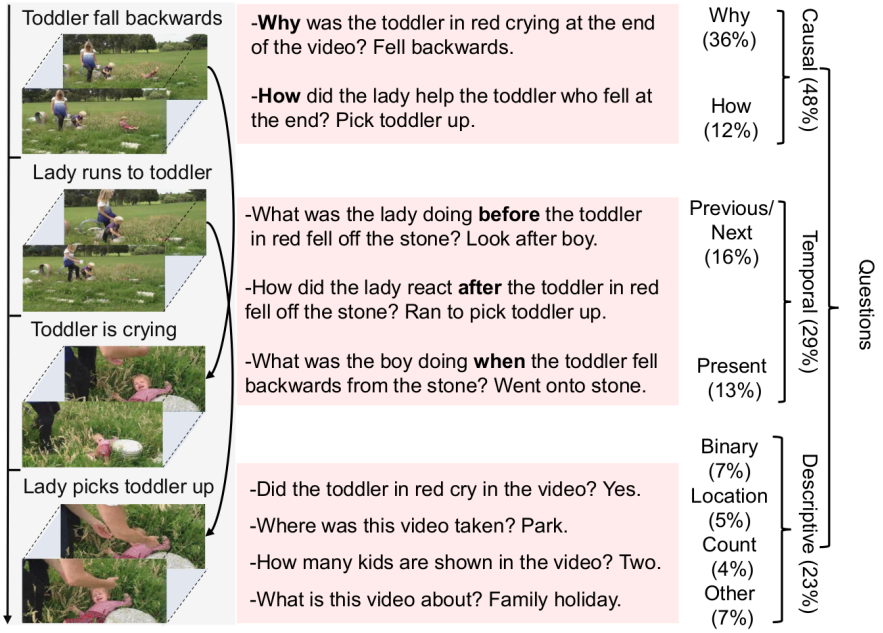
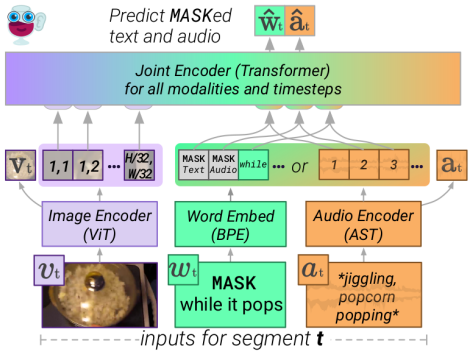


Figure 3.2: Example NExT-QA video and question types. From Xiao et al. (2021)

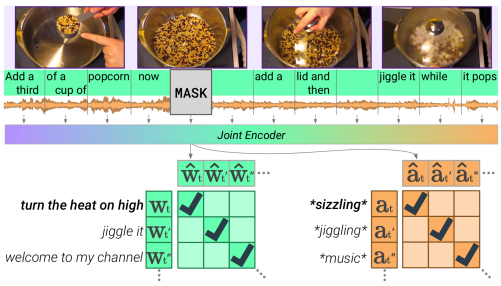
3.3 Merlot Reserve

Merlot Reserve (Zellers et al., 2022) is a pre-trained video language model which uses a contrastive objective that learns from aligned audio, subtitles and video frames. The authors collect a diverse dataset of 1 billion frames from YouTube videos. Videos are filtered to be high quality, favouring instructional videos so that there would be a visual grounding to the subtitles and audio. Subtitles include timing information for each utterance, and so it is possible to align segments of video, text, and audio to short timespans for all videos, with some later modifications to improve imperfect alignment. The architecture consists of independent encodings of each modality, via a Vision Transformer (Dosovitskiy et al., 2021), an Audio Spectrogram Transformer (Gong et al., 2021), and a Transformer span encoder, which computes targets from an embedding of a candidate text span (Fig. 3.3a). This is then fed into a joint Transformer encoder for all modalities and timesteps.

The specific pre-training objective is called contrastive span training (Fig. 3.3b). For each segment consisting of an aligned image, text, and audio encoding, a region of text and/or audio is masked out with 25% chance. The model must maximise its similarity only to an independent encoding of the text and audio. In pre-training the model learns to predict spans of text and audio in two cases: predicting audio where frames and text is provided; and predicting text where frames and audio are provided. The pre-training task learns both independent encoders of the three



(a) Merlot Reserve Architecture. Modalities are independently encoded before being jointly encoded to predict masked text and audio spans.



(b) Contrastive Span Training. The model maximises similarity of correct text and audio encodings to a joint encoding masked subsegment, while maximising dissimilarity of in-batch negatives.

Figure 3.3: Merlot Reserve Details. Figures taken from Zellers et al. (2022)

modalities as well as a joint encoder Transformer.

Once pre-trained, the model can be used by finetuning on a dataset, or zero-shot for a range of video and language tasks. The authors achieved state of the art results on visual commonsense reasoning (Zellers et al., 2019), TVQA (Antol et al., 2015), another video question answering dataset, and Kinetics-600 (Carreira et al., 2018), for activity understanding. Further, zero-shot experiments showed performance competing with those of supervised models on a range of video question answering datasets, and even slightly exceeding the supervised state of the art on STAR.

3.4 VideoCLIP

We use VideoCLIP (Xu et al., 2021b) as a probing model to test the existing temporal reasoning ability of existing video language models in Chapter 4. VideoCLIP trains on video-text clip pairs with a contrastive objective. It learns fine-grained associations between the modalities by using loosely temporally overlapping video and text clips as positive pairs, while sampling progressively harder negative pairs from the shared embedding space to the batch as training progresses and better representations are learned. In the case of an instructional video, the domain of the pretraining dataset used (Miech et al., 2019), a sentence such as "I am going to show you how to cook fried rice" may not be exactly aligned with the visual semantic content relating to the sentence. Using loosely overlapping video-text pairs can provide more relevant positive pairs for the contrastive loss, and in higher quantity,

than exactly temporally aligned pairs.

They extract 512-d video features every 30 frames, using a pre-trained video encoder S3D (Xie et al., 2018) on HowTo100M (Miech et al., 2019). S3D is a 3D CNN that extracts spatio-temporal features over time. The result is dense representations of many frames, which other models such as Merlot Reserve, Frozen (Bain et al., 2021), and CLIP4CLIP (Luo et al., 2022) do not have. For a 30 second video, VideoCLIP has visual input from 30 frames, compared to up to 12 in other models.

The authors do zero-shot evaluation on downstream tasks including text-to-video retrieval and video QA, and similarly to Merlot Reserve find that zero-shot evaluation outperforms even supervised previous work on select datasets. Since Zellers et al. (2022) and Xu et al. (2021b) use different datasets or splits to measure performance, we report results on the MSR-VTT QA dataset, using splits from Xu et al. (2017), where the model must choose from five candidate answers. Merlot Reserve achieves a zero-shot accuracy of 73.34% compared to 73.51% for VideoCLIP. So the two models are comparable in terms of headline downstream task performance on video question answering.

4. Probing Temporal Ability

In this chapter we probe the existing temporal reasoning abilities of two models: Merlot Reserve (Zellers et al., 2022) and VideoCLIP (Xu et al., 2021b). These models are chosen because of their use of a contrastive pre-training objective and zero-shot ability to test on downstream datasets. We describe the required adaptation of STAR to allow for zero-shot testing of both models before detailing results of targeted data perturbation for temporal features. The perturbations aim to identify how a model performs with incorrect data. We would expect a model to be more uncertain and for accuracy to go down.

4.1 Zero-Shot Setup

As in Zellers et al. (2022), we convert questions into statements to more closely match the text seen by the model during pre-training. Statements are rewordings of questions with answers masked out, since YouTube captions do not typically render question marks. Since not all of the conversion templates used for the Merlot Reserve model are given, we develop our own based on examples. We use STAR's question-answer template program and modify each question type into a statement type. For example, for the question and answer pair

`Q: ‘‘What happened after the person put down the towel?’’`
`A: ‘‘Threw the clothes.’’`

we are given the (slightly modified) question-answer template from STAR

`Q: ‘‘What happened after the person [VBP] [NP]?’’`
`A: ‘‘[Answer]’’.`

The corresponding statement is

`S: ‘‘The person _ after they put down the towel.’’`
`A: ‘‘threw the clothes’’,`

or in templated form

`S: ‘‘The person _ after they [VBP] [NP].’’`
`A: ‘‘[answer]’’.`

where the task becomes correctly identifying the correct answer to replace the masked span denoted by "_". A full conversion from question type to statement is given in Table 4.1.

Given these statements we present the model with the masked statement for each instance. This is aligned with the first frame, and all other frames have no aligned text. The correct answer is the choice most likely to replace the masked token(s).

For VideoCLIP, we use a similar approach to the zero-shot video question

answering approach in Xu et al. (2021b). That is, we formulate the task as a video-text retrieval task, except the candidate answers are associated with each video and the answer selected is the one that is most relevant out of the four options. We use the same statement templates as above, with the masked section replaced by each of four multiple choice options, to again reduce the domain shift from the pre-training data. The model was pre-trained on HowTo100M (Miech et al., 2019), a large-scale dataset of instructional videos. We found that using these statements produced better results than keeping the question-answer format. Each candidate answer is ranked according to the InfoNCE contrastive loss (van den Oord et al., 2019).

Table 4.1: Templates for conversion of all question types to statement types in STAR. An "_" in a statement indicates the masked answer. POS and syntax tags are as described in the Penn Treebank (Marcus et al., 1993). Verbs in Sequence_T1 and Sequence_T2 are post-processed to ensure grammaticality.

Type	Question	Statement
Interaction_T1	Which object was [VBD] by the person?	The object [VBD] by the person was _.
Interaction_T2	What did the person do with [NP]?	The person _ [NP].
Interaction_T3	What did the person do while they were [VBG] the [NP]?	The person _ while they were [VBG] the [NP].
Interaction_T4	What did the person do while they were [VBG1] [NP1] and [VBG2] [NP2]?	The person _ while they were [VBG1] [NP1] and [VBG2] [NP2].
Sequence_T1	Which object did the person [VB1] after they [VBP2] [NP]?	The person [VBP1] _ after they [VBP2] [NP].
Sequence_T2	Which object did the person [VB1] before they [VBP2] [NP]?	The person [VBP1] _ before they [VBP2] [NP].
Sequence_T3	What happened after the person [VBP] [NP]?	The person _ after they [VBP] [NP].
Sequence_T4	What happened before the person [VBP] [NP]?	The person _ before they [VBP] [NP].
Sequence_T5	What did the person do to [NP2] after [VBP1] [NP1]?	The person _ [NP2] after [VBP1] [NP1].
Sequence_T6	What did the person do to [NP2] before [VBP1] [NP1]?	The person _ [NP2] before [VBP1] [NP1].
Prediction_T1	What will the person do next?	The person will _ next.
Prediction_T2	What will the person do next with [NP]?	The person will _ [NP] next.
Prediction_T3	Which object would the person [VB] next?	The person would [VB] _ next.
Prediction_T4	Which object would the person [VB2] next after they [VB1] [NP]?	The person would [VB2] _ next after they [VB1] [NP].
Feasibility_T1	Which other object is possible to be [VBD] by the person?	The other object possible to be [VBD] by the person is _.
Feasibility_T2	What else is the person able to do with [NP]?	The person is also able to _ with [NP].
Feasibility_T3	Which object is possible to be [VBP] when the person is [PREP] [NP]?	The object possible to be [VBP] when the person is [PREP] [NP] is _.
Feasibility_T4	What is the person able to do when they are [PREP] [NP]?	The person is able to _ when they are [PREP] [NP].
Feasibility_T5	Which object is the person able to [VB1] after [VBG2] [NP]?	The person is able to [VB1] _ after [VBG2] NP.
Feasibility_T6	What is the person able to do after [VBG] [NP]?	The person is able to _ after [VBG] [NP].

Table 4.2: Results of models (accuracy) on STAR dataset. I, S, P, F stands for Interaction, Sequence, Prediction, and Feasibility. We focus on sequence questions, which generally require the most temporal reasoning.

	Question Types				
	I	S	P	F	Mean
Chance			25.00		
Merlot Reserve (val)	43.12	42.33	43.27	47.14	**43.97**
VideoCLIP (val)	39.66	**42.86**	48.72	50.82	42.84
Merlot Reserve (test)	40.51	**44.76**	43.85	39.48	42.15
VideoCLIP (test)	39.77	43.60	42.60	47.13	**43.27**
Merlot Reserve Paper (test)	44.8	42.4	38.8	36.2	40.5

4.2 Zero-Shot Results

Table 4.2 shows the reported results on STAR for both models. Our results on Merlot Reserve differ from the authors' reported results slightly; this may be down to different conversions from questions to statements. The models are generally comparable to each other, except VideoCLIP appears to be better at feasibility questions (e.g. "Which object is possible to be taken when the person is in front of the table?") than Merlot Reserve.

Having established the baseline results, we go on to test how we can explore the temporal reasoning ability through a series of data perturbations.

4.3 Changing Temporal Indicators

The first perturbation explores how the models perform when temporal expressions are changed, so that a question asks for events occurring at different times in the video to what the question actually asks, while keeping the 'correct' answer the same. For models that are able to reason across time, we hypothesise that this should result in worse performance because of the uncertainty caused by the incorrect time ordering in the question. A model that does not have temporal reasoning ability would have its capabilities limited lightly to not at all by this perturbation.

We test on the sequence subset of questions, since every sequence question has a temporal expression (before/after) in it, and reverse temporal indicators for each question. For example, the statement "The person opened _ *after* they took the sandwich" becomes "The person opened _ *before* they took the sandwich", with

Table 4.3: Statement perturbation for Sequence question types on STAR. Swapped indicates that a question has had its temporal expression (before/after) reversed. Mask Temporal Expressions asks the model to predict, given the two actions, whether one happened before or after the other, by masking out the temporal expression. Since the test set is withheld, we report results for this only on the validation set.

	Sequence Questions		
Model	Correct	Swapped	Mask Temporal Expressions
Chance	25.00		50.00
Merlot Reserve (val)	42.33	42.92 (+0.59)	50.86
VideoCLIP (val)	42.86	41.91 (-0.95)	50.11
Merlot Reserve (test)	44.76	41.04 (-3.72)	—
VideoCLIP (test)	43.60	43.54 (-0.06)	—

the masked answer remaining *the same* for both statements.

We report results in Table 4.3. For both Merlot Reserve and VideoCLIP models, there is very little difference between either test. In fact, for Merlot Reserve, the accuracy on this subset actually increases slightly from the correct statements. This suggests that other factors than temporal reasoning are contributing to the model's performance in this task.

One reason may be the selection of the other multiple choice options. As described in Section 3.1, options are generated to provide distractors based on compositionality, randomness, and frequency, but not for temporality. So a model may still pick the most likely option based on its ability to identify the answer using object detection, where the other three options are not present in the video at all. To counteract this possibility, we set up a binary-choice experiment, whereby we change the masked answers of statements to mask the temporal indicator before or after, creating a binary choice, and replace the previously masked answer with the correct answer from the dataset. Since we do not have gold labels for the test data, results are only reported for validation data. As seen in the 'Masked Temporal Indicators' column in Table 4.3, both models perform only marginally better than chance in this instance, suggesting that there is no sense of before or after in either model.

Table 4.4: Probing video perturbations. Shuffled video features accuracy on STAR test set. VideoCLIP results include both shuffling frame-level features, and computing features on a shuffled video. Probing video features results in a slight decrease in performance, with a large decrease for shuffled video.

	Question Types				
Model	I	S	P	F	Mean
Original Merlot Reserve	40.51	44.76	43.85	39.48	42.15
Merlot Reserve	38.88	40.61	42.32	39.48	40.32
Original VideoCLIP	39.77	43.60	42.60	47.13	43.27
VideoCLIP (shuffle frame)	39.27	43.31	42.04	46.09	42.68
VideoCLIP (shuffle video)	33.67	35.91	36.17	34.96	35.18

4.4 Randomise Video Frames

In addition to probing language understanding of temporal expressions, we test how well the two models encode visual features across frames. Following Sevilla-Lara et al. (2021), who found that performance on action classification degraded in humans when presented with out-of-order video frames, we randomise the order of frames presented to the models, and test on the default created statements. Concretely, for Merlot Reserve this involves shuffling the order of the 8 frames presented to the model, and aligning the statement to the new first frame. For VideoCLIP we take the 512-d feature vectors computed from every 30 frames (see Section 3.4), and shuffle based on the time axis. The mean video time is 29.8 seconds, so this involves shuffling on average 30 feature vectors for each clip. We also experiment with extracting features from videos that have had all their frames randomly shuffled. Since Merlot Reserve samples at the frame level anyway, there is no difference in the two approaches for Merlot Reserve, whereas the computation of features is affected by a video that has been randomly shuffled for VideoCLIP. Results are shown in Table 4.4.

We would expect that models with shuffled video features perform worse than the correct video features. With the exception of the shuffled video for VideoCLIP, there is little difference in performance between the setups. Computing features from randomly shuffled videos results in significantly worse performance due to the weaker features that can be learned from a jumbled-up video, where features are extracted in the temporal dimension. Shuffling well-made features, or images in the case of Merlot Reserve, at the frame level is reasonably robust to these effects, although there is a small drop in performance.

4.5 Summary

In this chapter we have looked at the existing capabilities for temporal reasoning of two state of the art video language models, Merlot Reserve and VideoCLIP. We found that their competitive performance on a video QA dataset with questions that should require temporal understanding was not because of the learned abilities of the models through a targeted probing setup. By perturbations of the evaluation data, for both language and video, we have shown that the models are not sensitive to misdirecting inputs, and are unable to identify (with random chance) the correct ordering of two actions when asked to determine whether one action occurs before or after another. In the next chapter we propose a method for learning fine-grained temporal understanding in the Merlot Reserve model by additional training on targeted data.

5. Method and Setup

We continue training the Merlot Reserve (Zellers et al., 2022) model on videos from the Charades dataset (Sigurdsson et al., 2016). We call this process post-pretraining, rather than finetuning, since the objective function is very similar to the pre-training stage, but we train on a smaller dataset that is designed to improve the temporal reasoning of the Merlot Reserve model before being evaluated on downstream tasks. This chapter explains the creation process of this new dataset, as well as the post-pretraining process.

5.1 Dataset Creation

We use videos and annotations from Charades to create our new dataset. This is motivated by the desire to have videos annotated with descriptive actions, while also having annotated timespans associated with the actions that allows for creating segments that are well aligned with frames presented to the model. From these actions and timespans, we are able to create text labels that describe a variety of temporal relations between a pair of actions.

Since one of our evaluation benchmarks, STAR (Wu et al., 2021), uses the same dataset as its video source, we filter video ids that appear in the validation and test sets of STAR to avoid contamination of the evaluation data. We further filter based on videos that contain at least 2 actions. This provides 7204 videos, annotated with actions and their timespans.

For each video, we create relation based on annotated actions. Each relation is based on Allen's Interval Algebra (Allen, 1983), shown in Table 5.1. This simple calculus provides a powerful template for reasoning about all types of temporal relation. By going beyond just before and after, we are able to model actions that happen at any point in relation to one another, providing a much more powerful schema.

The dataset is specifically designed to provide annotations that fit the expected input for Merlot Reserve, but could trivially be generalised to work with other models that use a masked language modelling objective. That is, we create aligned segments of frames, text, and audio per instance. Using actions and timestamps allows for a close alignment between frames and annotation, which would not be possible if we were to use whole video descriptions. Following Merlot Reserve, we use 8 segments per instance.

Table 5.1: The Thirteen Possible Relationships. All relation types except for *equal* have a corresponding inverse relation type, which is used instead 50% of the time. To create segments, three frames are selected based on the timespans of actions X and Y. m(·), s(·), e(·), t(·) indicate the mid, start, end and one-third points of a timespan, respectively. A timespan is indicated either by the action X or Y, or by the interval between timepoints, expressed using a colon (:). Modified from Allen (1983).

Relation	Example	Frames Selected
X *before* Y	XXX YYY	[m(X) ; m(e(X):s(Y)) ; m(Y)]
X *meets* Y	XXXYYY	[m(X) ; m(e(X):s(Y)) ; m(Y)]
X *overlaps* Y	XXX YYY	[t(X) ; m(s(Y):e(X)) ; 2t(Y)]
X *starts* Y	XXX YYYYY	[m(X) ; e(X) ; 2t(Y)]
X *during* Y	XXX YYYYYY	[s(Y) ; m(X) ; e(Y)]
X *finishes* Y	XXX YYYYY	[t(Y) ; s(X) ; m(X)]
X *equals* Y	XXX YYY	[m(s(X):s(Y)) ; m(m(X):m(Y)) ; m(e(X):e(Y))]

5.1.1 Creating Segments

For each video, we have relations between pairs of actions X, Y. We create instances for each relation, such that one instance contains annotations for a relation between exactly two actions. We create up to $\binom{N}{2}$ instances per video, where N is the number of actions labeled in each video. The total number of instances is 32627. An instance is made up of the label for action X, a temporal expression τ, and the label for action Y. Each action pair has a timespan $(X_{start}, X_{end}), (Y_{start}, Y_{end})$ associated with the action. τ is chosen based on these start and end times, according to Allen's Interval Algebra, including a threshold of 1 second to define the range of time that constitutes the difference between different relation types.[1] For example, if action X has timespan (0.4, 6.7) and action Y has timespan (6.5, 10.0), this would create the instance X *meets* Y, since the overlap between the end of X and the start of Y is less than the threshold, and such a small overlap does not constitute creating an instance with the relation type *overlaps*.

Once the relation type for a pair of actions is identified, we create the triple (X, τ, Y), split into three segments as follows. We split the label into three, depending on the relation type, to create a complete annotation that aims to match the average length of text spans from Merlot Reserve.

We then select frames and audio spectrograms based on timestamps. For the three annotated segments, timestamps are chosen based on the relation type and action timespans (see Table 5.1). The other five timestamps are then selected uniformly from either side of the annotations such that remaining frames span the length of the whole video. Depending on when the actions occur in the video, different numbers of timestamps will be taken from before and after the annotated segments. Once we have eight timestamps corresponding to the eight segments, we select one frame for each timestamp, and create audio spectrograms following Merlot Reserve, with 5 second clips surrounding the timestamp. If this causes an overlap with the timespan of another spectrogram, the overlap is silenced, so as not to cause any information leakage that may otherwise occur when predicting masked segments.

5.1.2 Positive Labels

Each segment is then made up of a frame, an audio spectrogram, and an annotation. The annotation may be empty, or may be one of the three segment labels we create in Section 5.1.1. Since our temporal relation types do not at this stage include inverse relations (we have a relation type before, but not after), we map

[1] We find that 1 second provides a good compromise balance slight inaccuracies in annotated time stamps while selecting the correct relation type.

Table 5.2: Mappings from temporal relation types to temporal expressions and their inverses. For inverse temporal expressions, the order of actions `X` and `Y` is swapped. For *starts* and *finishes* relations, the final segment includes `Y` or `X` depending on whether a normal or inverse temporal expression is used, respectively.

Relation Type	Temporal Expression	Inverse Temporal Expression	Annotation
before	`before`	`after`	`[X;`ϵ`;`τ`+Y]`
meets	`immediately before`	`immediately after`	`[X[:-1];X[-1:]+`τ`+Y[:1];Y[1:]]`
overlaps	`overlaps with`	`is overlapped by`	`[X[:-1];X[-1:]+`τ`+Y[:1];Y[1:]]`
starts	`(at the same time as, then continues)`	`(at the same time as, then continues)`	`[X;`τ`[0]+Y[:-1]; Y[-1:]+`τ`[1]+(Y|X)]`
during	`during`	`interrupted by`	`[X[:-1];X[-1:]+`τ`+Y[:1];Y[1:]]`
finishes	`(before, while)`	`(while, after)`	`[Y;`τ`[0]+X[:-1]; X[-1:]+`τ`[1]+(X|Y)]`
equals	`and`	`and`	`[X[:-1];X[-1:]+`τ`+Y[:1];Y[1:]]`

temporal relation types to temporal expressions with 50% chance that the annotation is inverted, including inverting the order of the actions. For example, if we have the triple (tidying up a blanket, *before*, tidying something on the floor), this could create the segment labels [tidying up a; blanket before tidying; something on the floor], or, if inverted, [tidying something on the; floor after tidying; up a blanket]. This creates a balance between opposite temporal relation types.

Mappings from temporal relation types to temporal expressions are shown in Table 5.2

5.1.3 Contrastive Span Objective

We use a slightly modified contrastive span objective, as in Zellers et al. (2022). The difference between our implementation and the original comes with the masking strategy. We restrict possibly masked spans to segments which contain a temporal word. Since we continue training on the pre-trained Merlot Reserve model, we do not require further training of the general span objective, but focus explicitly on the learning of temporal reasoning between relations. We do this by using additional hard negatives focused on temporal words, along with batch negatives. The hard negatives act as a close match to the positive option in the contrastive setup, but are specifically wrong in the temporal dimension. This focuses the model on learning how to reason across time.

5.1.4 Creating Negative Spans

We create a list of negative spans for each relation type. Negative spans are spans that match the corresponding positive span, except for temporal markers in the span. The temporal marker is changed to an alternative temporal marker that

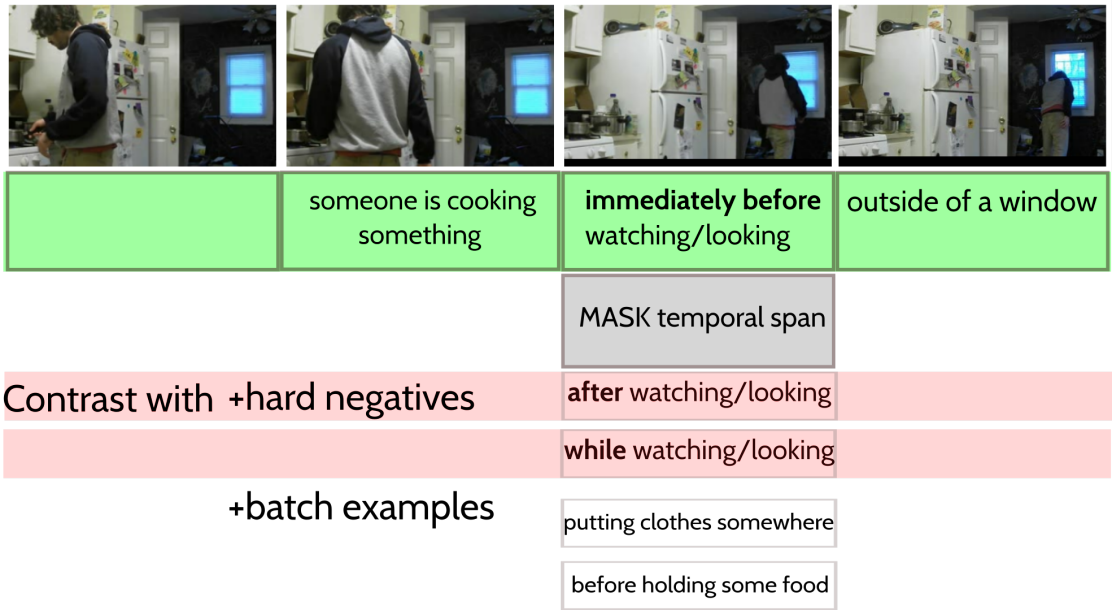


Figure 5.1: Example of an annotated instance, with a label across three segments. The middle segment contains a temporal word and is masked out. The contrastive span objective must identify the correct span out of the generated hard negatives and other batch spans. Note that audio and the other four segments are not shown here.

does not reflect the order of events as determined by the relation. For example, the relation type *before* is mapped to a set of negative temporal relation types *inv_before, equals, inv_meets, inv_overlaps, during, inv_starts, finishes*. Up to 5 temporal expressions are then selected from this set, allowing the contrastive objective to learn different gradations of temporality. Each negative span is the segment that contains the positive temporal marker, but substituted for a negative temporal marker. These are then provided to the model to use as additional hard negatives for the contrastive span objective. We show an example annotation in Fig. 5.1.

5.2 Merlot Reserve Post-Pretraining

Using this dataset, we post-pretrain Merlot Reserve with a similar pre-training objective and setup as described in Zellers et al. (2022). We emphasise the relevant points here.

5.2.1 Architecture

We continue training on the Merlot Reserve Base model (+audio), which achieved the highest downstream performance on STAR. The hyperparameters, unless otherwise specified, are the same as Table 8 in Zellers et al. (2022). The image encoder is a 12-layer ViT-B/16 Vision Transformer (Dosovitskiy et al., 2021), which encodes each frame independently. Images are scaled to 192×320 with a patch size of 16. Audio is encoded using an Audio Spectrogram Transformer (Gong et al., 2021), and we encode text spans into Byte Pair Encoding (BPE) tokens, using the same embedding table as Merlot Reserve. Differently to Merlot Reserve, we do not split up audio and text encodings into subsegments, since the length of the text spans in a segment are shorter in our dataset, and the average span length per segment is close to the desired length of 5 tokens.

As in Merlot Reserve, the three modalities are combined in a joint encoder over all input segments using a Transformer with 12 layers. Finally, a linear layer of size 768 projects the output of the final layer's hidden state for prediction of the masked segments. To learn the encodings of the targets for each modality, the final hidden state of a `CLS` token is used for the image and audio encoder, and a Transformer "span encoder" is learned to extract text targets "from a `CLS` [token] and embedded tokens of a candidate text span" (Zellers et al., 2022). The overview from Merlot Reserve is shown in Fig. 3.3a.

5.2.2 Objective Function

We use the contrastive span setup from Merlot Reserve, except we focus only on learning temporal relations, as described in Section 5.1.3. The objective function is therefore to minimise the cross-entropy between the masked-out representation of the temporal segment $\hat{\mathbf{w}}_t$ and the BPE encoding of the segment $\mathbf{w}_t$, along with encodings of spans from the batch $\mathcal{W}$ and additional hard negative spans $\mathcal{W}_{hard}$, as described in Section 5.1.4. We do not include any masked audio segments. The text span loss is therefore:

$$\mathcal{L}_{\text{text}} = \frac{1}{(\mid \mathcal{W} \mid + \mid \mathcal{W}_{\text{hard}} \mid)} \sum_{\mathbf{w}_t \in (\mathcal{W} \cup \mathcal{W}_{\text{hard}})} \left(\log \frac{\exp(\hat{\mathbf{w}}_t \cdot \mathbf{w}_t)}{\sum_{\mathbf{w} \in (\mathcal{W} \cup \mathcal{W}_{\text{hard}})} \exp(\hat{\mathbf{w}}_t \cdot \mathbf{w})} \right). \quad (5.1)$$

We also include the frame-matching objective $\mathcal{L}_{\text{frame}}$ from Merlot Reserve. The joint encoder encodes the entire annotation, and we maximise similarity between the ViT independent encoding of each frame to an extracted representation from each segment. The loss function is the same as Eq. (5.1), without the additional hard negatives $\mathcal{W}_{hard}$. The complete loss function is the sum of both objectives:

$$\mathcal{L} = \mathcal{L}_{\text{text}} + \mathcal{L}_{\text{frame}}. \quad (5.2)$$

We run our experiments on an NVIDIA GeForce RTX 3090 with a batch size of 8 for up to 3 epochs. We use AdamW (Loshchilov and Hutter, 2019) with the same optimizer parameters as in Merlot Reserve, except we use a learning rate of 5e-6 after linear warmup over 3750 steps. The next chapter details the downstream evaluation of this training process.

6. Results

In this chapter we evaluate the performance of post-pretraining the Merlot Reserve model on our proposed dataset from Chapter 5. We test downstream performance on STAR (Wu et al., 2021) and NExT-QA (Xiao et al., 2021), and analyse some of the choices we made in our dataset design process. We finish with a comparison to (Bagad et al., 2023).

6.1 Zero-Shot Downstream Results

6.1.1 STAR Results

We report zero-shot performance of our model compared to the original Merlot Reserve model in Table 6.1. We find that performance improves slightly for both validation and test splits. We especially observe a noticeable gain in performance on prediction questions. These have a temporal element to them, although we do not target this kind of hypothetical question type in our post-pretraining.

We go back to the probing experiments from Chapter 4, and ask how our model performs under the same conditions. Results are shown in Table 6.2. There is little change except in the masked temporal expressions task, which shows weaker performance in our trained model. Surprisingly, despite our dataset having an even split of inverse relations, we note a strong bias towards answering "before" for this masked temporal expressions task, and would be worthy of future investigation. We also note that in some other models we tested there would be a strong bias towards "after".

6.1.2 NExT-QA Results

To test the generalisability of our approach to a wider range of temporal relations, and a different domain, we also test zero-shot on NExT-QA. As in Section 4.1, we convert NExT-QA questions into statements to minimise distribution drift. Since each question is hand-written and does not follow a strict template for question types, we use a generative LLM to convert from questions to masked statements. We follow the approach in Zellers et al. (2022) for generating statements for MSRVTT-QA (Xu et al., 2016) questions, by providing a prompt for each question type. We use Mistral-7B-Instruct (Jiang et al., 2023), an LLM which has been finetuned to better respond to instructional prompts (see Ouyang et al. (2022)). For each question, we provide a different prompt depending on the question type, which includes an instruction of the task, three examples taken from the training set with hand-written statement conversions, and finally the question which is to

Table 6.1: Zero-shot STAR accuracy on Merlot Reserve and our post-pretrained model. Our model improves on Merlot Reserve for all question types.

	Question Types				
	I	S	P	F	Mean
Merlot Reserve (val)	43.12	42.33	43.27	47.14	43.01
Ours (val)	43.99	43.64	50.48	47.96	44.66
Merlot Reserve (test)	40.51	44.76	43.85	39.48	42.15
Ours (test)	41.49	44.88	46.09	40.70	43.29

Table 6.2: Probing Sequence Questions on validation data. Comparisons in brackets are compared to validation sequence values in Table 6.1. For swap and shuffle frames, a lower change is better.

Model	Swap (+/− ↓)	Shuffle Frames (+/− ↓)	Mask (↑)
Chance	25.00		50.00
Merlot Reserve	42.92 (+0.59)	42.58 (**+0.25**)	**50.86**
Ours	43.98 (**+0.34**)	44.67 (+1.03)	44.81

be converted. An example for Temporal Next (TN) questions, asking what will happen after an event, is shown below:

```
system: "You are a helpful assistant. The user will give an input
        question, and you will respond with the question in the
        form of a statement, giving space for an answer in the
        form of an underscore."
user: "what does the girl do after placing the mop down"
assistant: "the girl _ after placing the mop down"
user: "how does the child react after falling over"
assistant: "the child _ after falling over"
user: "what did the boy do after he stopped playing the drums the
      second time"
assistant: "the boy _ after he stopped playing the drums the
           second time"
user: ${question}
assistant:
```

We confirm that the output is valid by checking that there is one mask token per generated statement, and manually editing if there was not. We found 25

Table 6.3: NExT-QA Results on Merlot Reserve. On the ATP_{hard} subset, our model improves over Merlot Reserve, but neither model is robust at rejecting shuffled video features. Column headings are: Causal How, Causal Why, Temporal Current, Temporal Next, Temporal Previous, Descriptive Location, Descriptive Count, Descriptive Other.

	Question Types								
Method	CH	CW	TC	TN	TP	DL	DC	DO	Mean
Chance					20.0				
MReserve (val)	33.2	36.7	27.8	35.6	40.7	35.0	41.0	42.6	35.0
With Questions	35.3	34.0	29.8	31.4	30.8	15.8	34.8	25.9	32.2
ATP_{hard}	28.9	27.5	29.8	25.8	17.2				27.5
Shuffled Frames	29.7	27.7	30.0	24.7	24.1				27.6
Ours ATP	27.9	29.1	32.5	27.1	27.6				29.0
Shuffled Frames	26.3	30.4	33.6	28.2	31.0				29.7

examples that had to be manually edited, predominantly for questions such as asking to describe the colour of clothes worn by multiple people, where the answer is actually the same for both people.

We show our results in Table 6.3. We compare the performance on Merlot Reserve using our generated statements with using the existing questions and adding a mask token at the end of the question, simulating where an answer to the question would ordinarily go, and observe overall better performance using statements (35.0 vs 32.2), confirming that the generation process helps to mitigate distribution shift. We then evaluate our post-pretrained model on the ATP_{hard} subset. Similarly to STAR, we find that main task performance improves with our new model, although on the shuffled frames probe we find that the model is not robust to the perturbation, with even improved performance across most question types.

6.2 Qualitative Examples

Figure 6.1 shows an example where the post-pretrained model is able to reason more about before and after than the original Merlot Reserve model. Both Models get the original question right, but when we swap the temporal relation in the statement, our model becomes a lot more uncertain and selects a different answer, "happy". Note there is no ground truth for the swapped statement. From the

Figure 6.1: NExT-QA zero-shot example. Frames are ordered top row, then bottom row. Statement: THE GIRL IN BLACK WAS _ BEFORE SHE STOOD UP NEAR THE END
Swapped: THE GIRL IN BLACK WAS _ AFTER SHE STOOD UP NEAR THE END
Our model predicts the correct answer, "blowing" when presented with the correct statement, and predicts a different answer, "happy", when the statement is swapped. Merlot Reserve predicts "blowing" for both statements.

provided frames, it is quite possible that neither predicted answer is correct for the swapped statement. Using open-ended question answering may provide more insight into how a model approaches a task with incorrect data.

We also observe that the model can be limited in its ability to predict the correct answer just based on the frames selected. Figure 6.2 shows an example where the correct answer, "raised his hand to the camera" is not shown in any of the frames provided to the model, and so it is very hard for the model to select the correct answer from its inputs. A simple solution is to provide more frames as input, use ensembles of models with different frame selection processes, or use selection methods such as Buch et al. (2022); Lei et al. (2023) for selecting informative frames from a video.

6.3 Expanding Temporal Relation Types

We look at the effect of training with only inverse relation types as negative span candidates, as opposed to a range of negatives to provide a more fine-grained temporal understanding. We find that using more relation types improves performance on sequence questions, with an accuracy of 43.64 compared to 40.18 using only inverse relations.

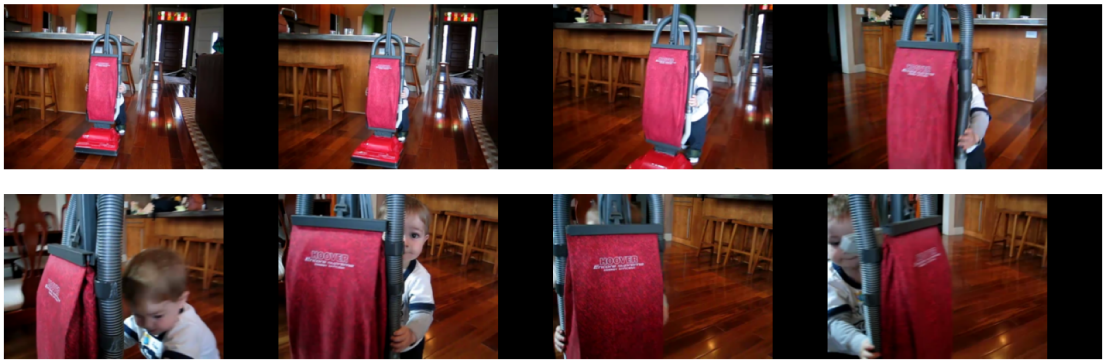


Figure 6.2: Frame selection means the model can have very little chance of selecting the correct answer.
Statement: THE BABY _ AFTER HE APPROACHED NEAR THE CAMERA
Prediction is "suck his thumb", ground truth is "raised his hand to the camera"

Table 6.4: Annotation Method results on STAR (validation set). We test with spans that contain only temporal words compared to including part of action annotations in the masked temporal span.

	Sequence	Mean (All)
Combination	43.64	44.66
Only temporal	34.69	33.94

6.4 Selecting Annotation Method

We also explore alternative approaches for creating the segments. Remember from Section 5.1.1 that we split labels across three segments, combining action annotations across segment boundaries depending on relation type. We experiment with creating segments formed distinctly of the action annotation and temporal expression in different segments, i.e. [X;τ;Y] for actions X, Y, and temporal relation type τ. We find that this significantly degrades performance (Table 6.4), and hypothesise that this is due to the shorter length of the temporal relation, often just a single token, which differs significantly from the length of segments found in pre-training.

6.5 Comparison to Test of Time

Finally, we compare our results to Test of Time (Bagad et al., 2023). We run their TACT model, trained on TEMPO, on our perturbation tests from Chap-

Table 6.5: Test of Time (TEMPO TACT) accuracy on STAR validation set. Overall performance is lower than the base model VideoCLIP, but there is a slightly wider performance gap in the perturbations.

	Question Types				
	I	S	P	F	Mean
VideoCLIP	39.66	42.86	48.72	50.82	42.84
Swapped before/after		41.91			
Masked temporal expressions		50.11			
Shuffled videos	34.61	36.31	36.70	40.82	36.08
Shuffled frames	39.99	43.06	46.47	49.59	43.39
TEMPO TACT	39.49	39.88	47.44	46.33	40.86
Swapped before/after		37.73			
Masked temporal expressions		57.53			
Shuffled videos	34.15	33.27	39.26	37.14	34.36
Shuffled frames	38.49	38.09	44.55	42.25	39.08

ter 4 (Table 6.5). We find that TACT performs slightly worse than VideoCLIP overall, although the probes achieve a greater loss, suggesting that the model has increased uncertainty for those tricky cases. Note the sequence column, which sees a drop of just under 1% on the base VideoCLIP model, but over 2% when post-pretrained with TACT, on the swapped before/after test.

As Bagad et al. (2023) mention, their approach was most successful on VideoCLIP. On other models (Frozen (Bain et al., 2021), VindLU (Cheng et al., 2023), CLIP4CLIP (Luo et al., 2022)), their performance is not as strong. They hypothesise that this is due to the number of frames provided as input to the model, with 32 provided to VideoCLIP compared to a maximum of 12 in others. Merlot Reserve only provides 8, and as our findings in Section 6.2 suggest, this may be a limiting factor on further improving performance in this direction.

In comparison to Test of Time, we develop a dataset that trains on full videos, rather than stitched together clips. This allows us to use a wider range of temporal relations, based on Allen's Interval Algebra, that results in improved downstream performance on sequence questions (Section 6.3).

6.6 Summary

We have looked at the performance of our proposed post-pretraining regime on the modified Charades dataset. There is an improvement on downstream video

QA tasks, suggesting more capable models have been learned, although there is little evidence to say that the temporal reasoning ability of models has improved, particularly in robustness. We compared architectural dataset decisions, in terms of the number and type of hard negatives to include, and found that using more temporal relation types increased downstream performance.

Conclusion

In this thesis, we investigated the ability of vision and language models (VLMs) to reason across time. We found that current models, trained with a contrastive learning objective, do not use temporal indicators, even for questions that ought to require them. When predicting whether one action happens before or after another, models perform at chance at best. Following previous work, we attempted to instill a sense of temporal reasoning into one specific model, Merlot Reserve. We use hard negatives focussed on temporal expressions to make models more sensitive to temporal cues, with the expectation that performance on tasks that require temporal information would improve, and that models would become more robust to misleading temporal information, such as the examples shown in Chapter 4.

We proposed a new dataset based on Charades (Sigurdsson et al., 2016) for the Merlot Reserve model (Zellers et al., 2022). This dataset added focussed hard negatives to segments which include a temporal expression. Training on this dataset, with hard negatives included as additions to the contrastive loss function, we found that our model improved on zero-shot downstream video QA tasks, but did not show clear signs of improvement in our probing tests. We did find some qualitative suggestions of more model uncertainty, however.

This may be down to the multiple choice nature of the datasets, which often do not provide distractor options that are incorrect in the sense of time. Future work may consider this as an added dimension to challenge models in when considering multiple-choice video question answering. Open-ended video QA was not considered here, since the models we evaluate are not able to generate text, but it may be an interesting study to compare our probing methods with the generated answers given for open-ended video QA datasets.

Finally, there are technical changes and optimisations that could be made. As noted in Section 1.2.2, contrastive learning often requires large batches to work effectively. The largest batch size we were able to train with was 8, which is at least an order of magnitude away from batch sizes considered by previous work. We also acknowledged that the frame selection process at inference time could be improved, and suggested an approach for selecting more informative frames from a video.

Bibliography

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1203. URL https://aclanthology.org/D16-1203.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, November 1983. ISSN 0001-0782. doi: 10.1145/182.358434. URL https://doi.org/10.1145/182.358434.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

Piyush Bagad, Makarand Tapaswi, and Cees G. M. Snoek. Test of Time: Instilling Video-Language Models with a Sense of Time. In *CVPR*, 2023.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision*, 2021.

Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.463. URL https://aclanthology.org/2020.acl-main.463.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, mar 2003. ISSN 1532-4435.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the "video" in video-language understanding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2907–2917, 2022. doi: 10.1109/CVPR52688.2022.00293.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, page 213–229, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58451-1. doi: 10.1007/978-3-030-58452-8_13. URL `https://doi.org/10.1007/978-3-030-58452-8_13`.

Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600, 2018, 1808.01340.

Feng Cheng, Xizi Wang, Jie Lei, David Crandall, Mohit Bansal, and Gedas Bertasius. Vindlu: A recipe for effective video-and-language pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10739–10750, June 2023.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

Jeffrey L Elman and Ken McRae. A model of event knowledge. *Psychological Review*, 126(2):252, 2019.

Yuan Gong, Yu-An Chung, and James Glass. AST: Audio spectrogram transformer, 2021, 2104.01778.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5843–5851, 2017. doi: 10.1109/ICCV.2017.622.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning

for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with temporal language. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1380–1390, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1168. URL https://aclanthology.org/D18-1168.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=iBBcRUlOAPR.

De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7366–7375, 2018. doi: 10.1109/CVPR.2018.00769.

Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, 2019. doi: 10.1109/CVPR.2019.00686.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023, 2310.06825.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compo-

sitional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, 12 1989, https://direct.mit.edu/neco/article-pdf/1/4/541/811941/neco.1989.1.4.541.pdf. ISSN 0899-7667. doi: 10.1162/neco.1989.1.4.541. URL https://doi.org/10.1162/neco.1989.1.4.541.

Jie Lei, Licheng Yu, Tamara Berg, and Mohit Bansal. TVQA+: Spatio-temporal grounding for video question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8211–8225, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.730. URL https://aclanthology.org/2020.acl-main.730.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7331–7341, June 2021.

Jie Lei, Tamara Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 487–507, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.29. URL https://aclanthology.org/2023.acl-long.29.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and gen-

eration. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023a.

Linjie Li, Zhe Gan, Kevin Lin, Chung-Ching Lin, Zicheng Liu, Ce Liu, and Lijuan Wang. Lavender: Unifying video-language understanding as masked language modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23119–23129, June 2023b.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language, 2019, 1908.03557.

Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, page 388–404, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-19832-8. doi: 10.1007/978-3-031-19833-5_23. URL https://doi.org/10.1007/978-3-031-19833-5_23.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach, 2020. URL https://openreview.net/forum?id=SyxS0T4tvS.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019, 1711.05101.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2022.07.028. URL https://www.sciencedirect.com/science/article/pii/S0925231222008876.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://aclanthology.org/J93-2004.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by

watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Marc Moens and Mark Steedman. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15–28, 1988. URL `https://aclanthology.org/J88-2003`.

Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. Verbs in action: Improving verb understanding in video-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15579–15591, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Jesús Andrés Portillo-Quintero, José Carlos Ortiz-Bayliss, and Hugo Terashima-Marín. A straightforward framework for video retrieval using clip. In Edgar Roman-Rangel, Ángel Fernando Kuri-Morales, José Francisco Martínez-Trinidad, Jesús Ariel Carrasco-Ochoa, and José Arturo Olvera-López, editors, *Pattern Recognition*, pages 3–12, Cham, 2021. Springer International Publishing. ISBN 978-3-030-77004-4.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani. Only time can tell: Discovering temporal data for temporal modeling. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 535–544, 2021. doi: 10.1109/WACV48630.2021.00058.

Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ivan Laptev, Ali Farhadi, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. *ArXiv e-prints*, 2016, 1604.01753. URL `http://arxiv.org/abs/1604.01753`.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CRCV-TR-12-01*, 2012.

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. 2019, 1807.03748.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Zhenhailong Wang, Manling Li, Ruochen Xu, Luowei Zhou, Jie Lei, Xudong Lin, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Derek Hoiem, Shih-Fu Chang, Mohit Bansal, and Heng Ji. Language models with image descriptors are strong few-shot video-language learners. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=_LceCyuVcH`.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf`.

Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. STAR: A benchmark for situated reasoning in real-world videos. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9777–9786, June 2021.

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, page 318–335, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-01266-3. doi: 10.1007/978-3-030-01267-0_19. URL `https://doi.org/10.1007/978-3-030-01267-0_19`.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*, 2017.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Prahal Arora, Masoumeh Aminzadeh, Christoph Feichtenhofer, Florian Metze, and Luke Zettlemoyer. VLM: Task-agnostic video-language model pre-training for video understanding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4227–4239, Online, August 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.370. URL https://aclanthology.org/2021.findings-acl.370.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. VideoCLIP: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6787–6800, Online and Punta Cana, Dominican Republic, November 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.544. URL https://aclanthology.org/2021.emnlp-main.544.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. URL https://www.microsoft.com/en-us/research/publication/msr-vtt-a-large-video-description-dataset-for-bridging-video-and-language/.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, Aug 2022, 2022. URL https://arxiv.org/abs/2205.01917.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KRLUvxh8uaX.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *2019 IEEE/CVF Conference on*

Computer Vision and Pattern Recognition (CVPR), pages 6713–6724, 2019. doi: 10.1109/CVPR.2019.00688.

Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16375–16387, 2022.

Andy Zeng, Maria Attarian, brian ichter, Krzysztof Marcin Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael S Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=G2Q2Mh3avow`.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12104–12113, June 2022.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5014–5022, 2016.

Ben Zhou, Kyle Richardson, Qiang Ning, Tushar Khot, Ashish Sabharwal, and Dan Roth. Temporal reasoning on implicit events from distant supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1361–1371, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.107. URL `https://aclanthology.org/2021.naacl-main.107`.

List of Figures

List of Tables

List of Abbreviations

CLIP contrastive language-image pre-training. 3, 8, 12, 13, 17

CNN convolutional neural network. 7, 8, 10

LLM large language model. 9, 10, 12, 18, 39

LSTM long short-term memory network. 6

NLP natural language processing. 7, 10

RNN recurrent neural network. 5, 6

video QA video question answering. 2, 3, 19, 25, 32, 44, 45

vidLM video language model. 2–4, 9, 13, 16–19, 24, 32

ViT Vision Transformer. 8, 9, 15, 37, 38

VLM vision and language model. 2, 3, 6, 9, 10, 12, 13, 17, 45

VQA visual question answering. 3, 10, 11