VideoBERT: A Joint Model for Video and Language Representation Learning

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid

Google Research

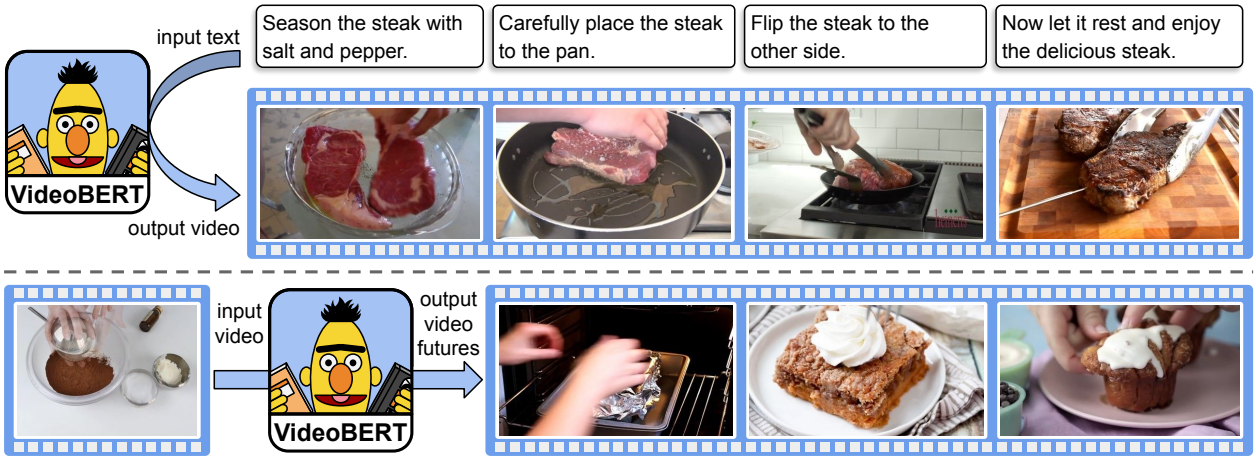


Figure 1: **VideoBERT text-to-video generation and future forecasting.** (Above) Given some recipe text divided into sentences, $y = y_{1:T}$, we generate a sequence of video tokens $x = x_{1:T}$ by computing $x_t^* = \arg\max_k p(x_t = k|y)$ using VideoBERT. (Below) Given a video token, we show the top three future tokens forecasted by VideoBERT at different time scales. In this case, VideoBERT predicts that a bowl of flour and cocoa powder may be baked in an oven, and may become a brownie or cupcake. We visualize video tokens using the images from the training set closest to centroids in feature space.

Abstract

Self-supervised learning has become increasingly important to leverage the abundance of unlabeled data available on platforms like YouTube. Whereas most existing approaches learn low-level representations, we propose a joint visual-linguistic model to learn high-level features without any explicit supervision. In particular, inspired by its recent success in language modeling, we build upon the BERT model to learn bidirectional joint distributions over sequences of visual and linguistic tokens, derived from vector quantization of video data and off-the-shelf speech recognition outputs, respectively. We use VideoBERT in numerous tasks, including action classification and video captioning. We show that it can be applied directly to open-vocabulary classification, and confirm that large amounts of training data and cross-modal information are critical to performance. Furthermore, we outperform the state-of-the-art on video captioning, and quantitative results verify that the model learns high-level semantic features.

1. Introduction

Deep learning can benefit a lot from labeled data [24], but this is hard to acquire at scale. Consequently there has been a lot of recent interest in "self supervised learning", where we train a model on various "proxy tasks", which we hope will result in the discovery of features or representations that can be used in downstream tasks. A wide variety of such proxy tasks have been proposed in the image and video domains. However, most of these methods focus on low level features (e.g., textures) and short temporal scales (e.g., motion patterns that last a second or less). We are interested in discovering high-level semantic features which correspond to actions and events that unfold over longer time scales (e.g. minutes), since such representations would be useful for various video understanding tasks.

In this paper, we exploit the key insight that human language has evolved words to describe high-level objects and events, and thus provides a natural source of "self" supervision. In particular, we present a simple way to model the relationship between the visual domain and the

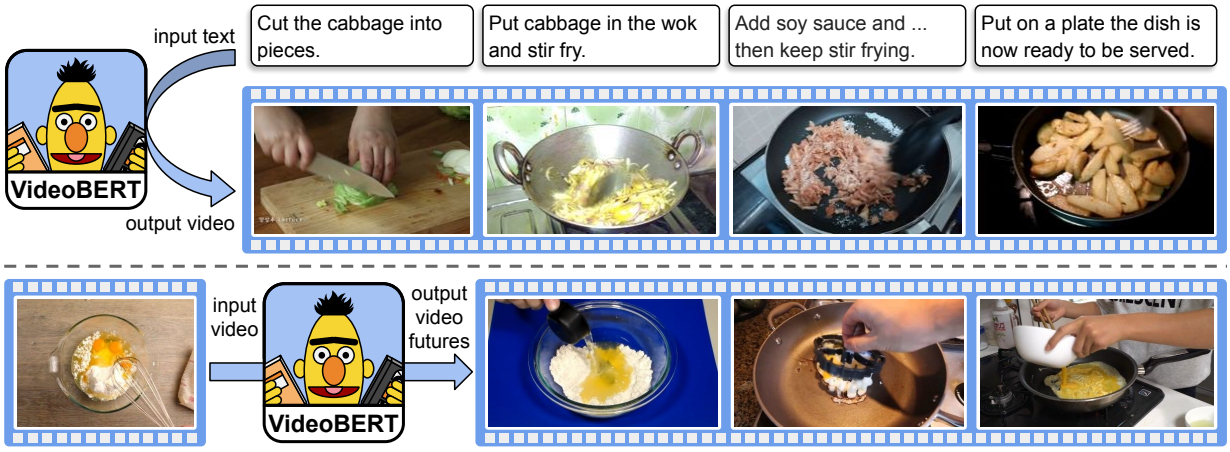


Figure 2: Additional text-to-video generation and future forecasting examples from VideoBERT, see Figure 1 for details.

linguistic domain by combining three off-the-shelf methods: an automatic speech recognition (ASR) system to convert speech into text; vector quantization (VQ) applied to low-level spatio-temporal visual features derived from pre-trained video classfication models; and the recently proposed BERT model [6] for learning joint distributions over sequences of discrete tokens.

More precisely, our approach is to apply BERT to learn a model of the form $p(x, y)$, where x is a sequence of "visual words", and y is a sequence of spoken words. Given such a joint model, we can easily tackle a variety of interesting tasks. For example, we can perform text-to-video prediction, which can be used to automatically illustrate a set of instructions (such as a recipe), as shown in the top examples of Figure 1 and 2. We can also perform the more traditional video-to-text task of dense video captioning [10] as shown in Figure 6. In Section 4.6, we show that our approach to video captioning significantly outperforms the previous state-of-the-art [39] on the YouCook II dataset [38].

We can also use our model in a "unimodal" fashion. For example, the implied marginal distribution $p(x)$ is a language model for visual words, which we can use for long-range forecasting. This is illustrated in the bottom examples of Figure 1 and 2. Of course, there is uncertainty about the future, but the model can generate plausible guesses at a much higher level of abstraction than other deep generative models for video, such as those based on VAEs or GANs (see e.g., [4, 5, 13, 27]), which tend to predict small changes to low level aspects of the scene, such as the location or pose of a small number of objects.

In summary, our main contribution in this paper is a simple way to learn high level video representations that capture semantically meaningful and temporally long-range structure. The remainder of this paper describes this contribution in detail. In particular, Section 2 briefly reviews related work; Section 3 describes how we adapt the recent progress in natural language modeling to the video domain; Section 4 presents results on activity recognition and video captioning tasks; and Section 5 concludes.

2. Related Work

Supervised learning. Some of the most successful approaches for video representation learning have leveraged large labeled datasets (e.g., [9, 19, 36, 7]) to train convolutional neural networks for video classification. However, it is very expensive to collect such labeled data, and the corresponding label vocabularies are often small and not capable of representing the nuances of many kinds of actions (e.g., "sipping" is slightly different than "drinking" which is slightly different than "gulping"). In addition, these approaches are designed for representing short video clips, typically a few seconds long. The main difference to our work is that we focus on the long-term evolution of events in video, and we do not use manually provided labels.

Unsupervised learning. Recently, a variety of approaches for learning density models from video have been proposed. Some use a single static stochastic variable, which is then "decoded" into a sequence using an RNN, either using a VAE-style loss [32, 35] or a GAN-style loss [31, 17]. More recent work uses temporal stochastic variables, e.g., the SV2P model of [4] and the SVGLP model of [5]. There are also various GAN-based approaches, such as the SAVP approach of [13] and the MoCoGAN approach of [27]. We differ from this work in that we use the BERT model, without any explicit stochastic latent variables, applied to visual tokens derived from the video. Thus our model is not a generative model of pixels, but it is a generative model of features derived from pixels, which is an approach that has been used in other work (*e.g.*, [30]).

Self-supervised learning. To avoid the difficulties of learning a joint model $p(x_{1:T})$, it has become popular to learn conditional models of the form $p(x_{t+1:T}|x_{1:t})$, where we partition the signal into two or more blocks, such as gray

scale and color, or previous frame and next frame (e.g., [18]), and try to predict one from the other (see e.g., [23] for an overview). Our approach is similar, except we use quantized visual words instead of pixels. Furthermore, although we learn a set conditional distributions, our model is a proper joint generative model, as explained in Section 3.

Cross-modal learning. The multi-modal nature of video has also been an extensive source of supervision for learning video representations, which our paper builds on. Since most videos contain synchronized audio and visual signals, the two modalities can supervise each other to learn strong self-supervised video representations [3, 20, 21]. In this work, we use speech (provided by ASR) rather than low-level sounds as a source of cross-modal supervision.

Natural language models. We build upon recent progress in the NLP community, where large-scale language models such as ELMO [22] and BERT [6] have shown state-of-the-art results for various NLP tasks, both at the word level (e.g., POS tagging) and sentence level (e.g., semantic classification). The BERT model is then extended to pre-train on multi-lingual data [12]. Our paper builds on the BERT model to capture structure in both the linguistic and visual domains.

Image and video captioning. There has been much recent work on image captioning (see e.g., [11, 8, 15]), which is a model of the form $p(y|x)$, where y is the manually provided caption and x is the image. There has also been some work on video captioning, using either manually provided temporal segmentation or estimated segmentations (see e.g., [10, 39]). We use our joint $p(x, y)$ model and apply it to video captioning, and achieve state-of-the-art results, as we discuss in Section 4.6.

Instructional videos. Various papers (e.g., [16, 2, 10, 38, 39]) have trained models to analyse instructional videos, such as cooking. We differ from this work in that we do not use any manual labeling, and we learn a large-scale generative model of both words and (discretized) visual signals.

3. Models

In this section, we briefly summarize the BERT model, and then describe how we extend it to jointly model video and language data.

3.1. The BERT model

BERT [6] proposes to learn language representations by using a "masked language model" training objective. In more detail, let $x = \{x_1, \ldots, x_L\}$ be a set of discrete tokens, $x_l \in \mathcal{X}$. We can define a joint probability distribution over this set as follows:

$$p(x|\theta) = \frac{1}{Z(\theta)} \prod_{l=1}^{L} \phi_l(x|\theta) \propto \exp\left(\sum_{l=1}^{L} \log \phi_l(x|\theta)\right)$$

where $\phi_l(x)$ is the l'th potential function, with parameters θ, and Z is the partition function.

The above model is permutation invariant. In order to capture order information, we can "tag" each word with its position in the sentence. The BERT model learns an embedding for each of the word tokens, as well as for these tags, and then sums the embedding vectors to get a continuous representation for each token. The log potential (energy) functions for each location are defined by

$$\log \phi_l(x|\theta) = x_l^T f_\theta(x_{\setminus l})$$

where x_l is a one-hot vector for the l'th token (and its tag), and

$$x_{\setminus l} = (x_1, \ldots, x_{l-1}, \text{MASK}, x_{l+1}, \ldots, x_L)$$

The function $f(x_{\setminus l})$ is a multi-layer bidirectional transformer model [28] that takes an $L \times D_1$ tensor, containing the D_1-dimensional embedding vectors corresponding to $x_{\setminus l}$, and returns an $L \times D_2$ tensor, where D_2 is the size of the output of each transformer node. See [6] for details. The model is trained to approximately maximize the pseudo log-likelihood

$$L(\theta) = E_{x \sim D} \sum_{l=1}^{L} \log p(x_l | x_{\setminus l}; \theta)$$

In practice, we can stochastically optimize the logloss (computed from the softmax predicted by the f function) by sampling locations as well as training sentences.

BERT can be extended to model two sentences by concatenating them together. However, we are often not only interested in simply modeling the extended sequence, but rather relationships between the two sentences (e.g., is this a pair of consecutive or randomly selected sentences). BERT accomplishes this by prepending every sequence with a special classification token, `[CLS]`, and by joining sentences with a special separator token, `[SEP]`. The final hidden state corresponding to the `[CLS]` token is used as the aggregate sequence representation from which we predict a label for classification tasks, or which may otherwise be ignored. In addition to differentiating sentences with the `[SEP]` token, BERT also optionally tags each token by the sentence it comes from. The corresponding joint model can be written as $p(x, y, c)$, where x is the first sentence, y is the second, and $c = \{0, 1\}$ is a label indicating whether the sentences were separate or consecutive in the source document.

For consistency with the original paper, we also add a `[SEP]` token to the end of the sequence, even though it is not strictly needed. So, a typical masked-out training sentence pair may look like this: `[CLS] let's make a traditional [MASK] cuisine [SEP] orange chicken with [MASK] sauce [SEP].` The corresponding class label in this case would be $c = 1$, indicating that x and y are consecutive.

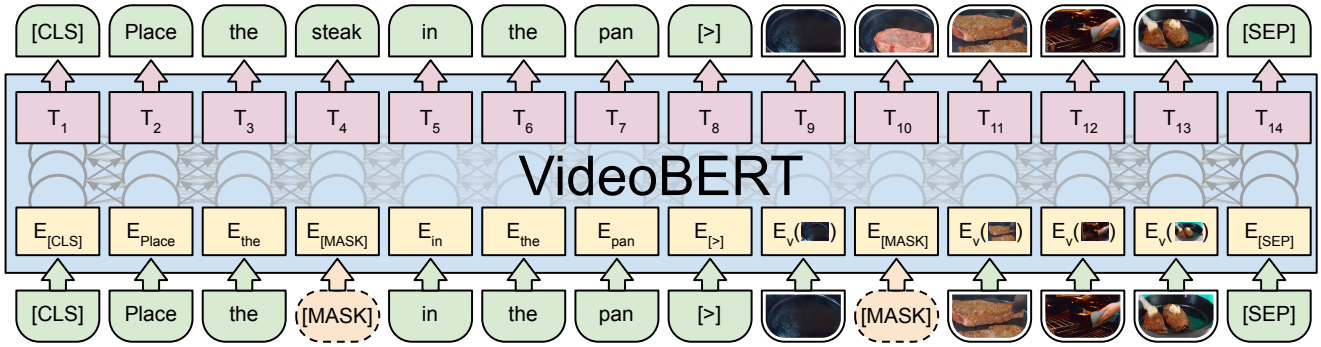


Figure 3: Illustration of VideoBERT in the context of a video and text masked token prediction, or *cloze*, task. This task also allows for training with text-only and video-only data, and VideoBERT can furthermore be trained using a linguistic-visual alignment classification objective (not shown here, see text for details).

3.2. The VideoBERT model

To extend BERT to video, in such a way that we may still leverage pretrained language models and scalable implementations for inference and learning, we decided to make minimal changes, and transform the raw visual data into a discrete sequence of tokens. To this end, we propose to generate a sequence of "visual words" by applying hierarchical vector quantization to features derived from the video using a pretrained model. See Section 4.2 for details. Besides its simplicity, this approach encourages the model to focus on high level semantics and longer-range temporal dynamics in the video. This is in contrast to most existing self-supervised approaches to video representation learning, which learn low-level properties such as local textures and motions, as discussed in Section 2.

We can combine the linguistic sentence (derived from the video using ASR) with the visual sentence to generate data such as this: `[CLS] orange chicken with [MASK] sauce [>] v01 [MASK] v08 v72 [SEP]`, where `v01` and `v08` are visual tokens, and `[>]` is a special token we introduce to combine text and video sentences. See Figure 3 for an illustration.

While this *cloze* task extends naturally to sequences of linguistic and visual tokens, applying a next sentence prediction task, as used by BERT, is less straightforward. We propose a linguistic-visual alignment task, where we use the final hidden state of the `[CLS]` token to predict whether the linguistic sentence is temporally aligned with the visual sentence. Note that this is a noisy indicator of semantic relatedness, since even in instructional videos, the speaker may be referring to something that is not visually present.

To combat this, we first randomly concatenate neighboring sentences into a single long sentence, to allow the model to learn semantic correspondence even if the two are not well aligned temporally. Second, since the pace of state transitions for even the same action can vary greatly between different videos, we randomly pick a subsampling rate of 1 to 5 steps for the video tokens. This not only helps the model be more robust to variations in video speeds, but also allows the model to capture temporal dynamics over greater time horizons and learn longer-term state transitions. We leave investigation into other ways of combining video and text to future work.

Overall, we have three training regimes corresponding to the different input data modalities: text-only, video-only and video-text. For text-only and video-only, the standard mask-completion objectives are used for training the model. For text-video, we use the linguistic-visual alignment classification objective described above. The overall training objective is a weighted sum of the individual objectives. The text objective forces VideoBERT to do well at language modeling; the video objective forces it to learn a "language model for video", which can be used for learning dynamics and forecasting; and the text-video objective forces it to learn a correspondence between the two domains.

Once we have trained the model, we can use it in a variety of downstream tasks, and in this work we quantitatively evaluate two applications. In the first application, we treat it as a probabilistic model, and ask it to predict or impute the symbols that have been MASKed out. We illustrate this in Section 4.4, where we perform "zero-shot" classification. In the second application, we extract the predicted representation (derived from the internal activations of the model) for the `[CLS]` token, and use that dense vector as a representation of the entire input. This can be combined with other features derived from the input to be used in a downstream supervised learning task. We demonstrate this in Section 4.6, where we perform video captioning.

4. Experiments and Analysis

In this section we describe our experimental setup, and show quantitative and qualitative results.

4.1. Dataset

Deep learning models, in both language and vision domains, have consistently demonstrated dramatic gains in performance with increasingly large datasets. For example, the "large" BERT model (which we use) was pretrained on the concatenation of the BooksCorpus (800M words) and English Wikipedia (2,500M words).

Therefore, we would like to train VideoBERT with a comparably large-scale video dataset. Since we are interested in the connection between language and vision, we would like to find videos where the spoken words are more likely to refer to visual content. Intuitively, this is often the case for instructional videos, and we focus on cooking videos specifically, since it is a well studied domain with existing annotated datasets available for evaluation. Unfortunately, such datasets are relatively small, so we turn to YouTube to collect a large-scale video dataset for training.

We extract a set of publicly available cooking videos from YouTube using the YouTube video annotation system to retrieve videos with topics related to "cooking" and "recipe". We also filter videos by their duration, removing videos longer than 15 minutes, resulting in a set of 312K videos. The total duration of this dataset is 23,186 hours, or roughly 966 days. For reference, this is more than two orders of magnitude larger than the next largest cooking video dataset, YouCook II, which consists of 2K videos with a total duration of 176 hours [38].

To obtain text from the videos, we utilize YouTube's automatic speech recognition (ASR) toolkit provided by the YouTube Data API [1] to retrieve timestamped speech information. The API returns word sequences and the predicted language type. Among the 312K videos, 180K have ASR that can be retrieved by the API, and 120K of these are predicted to be in English. In our experiments, while we use all videos for the video-only objective, we only use text from English ASR for VideoBERT's text-only and video-text objectives.

We evaluate VideoBERT on the YouCook II dataset [38], which contains 2000 YouTube videos averaging 5.26 minutes in duration, for a total of 176 hours. The videos have manually annotated segmentation boundaries and captions. On average there are 7.7 segments per video, and 8.8 words per caption. We use the provided dataset split, with 1333 videos for training and 457 for validation. To avoid potential bias during pretraining, we also remove any videos which appear in YouCook II from our pretraining set.

4.2. Video and Language Preprocessing

For each input video, we sample frames at 20 fps, and create clips from 30-frame (1.5 seconds) non-overlapping windows over the video. For each 30-frame clip, we apply a pretrained video ConvNet to extract its features. In this work, we use the S3D [34] which adds separable temporal convolutions to an Inception network [25] backbone. We take the feature activations before the final linear classifier and apply 3D average pooling to obtain a 1024-dimension feature vector. We pretrain the S3D network on the Kinetics [9] dataset, which covers a wide spectrum of actions from YouTube videos, and serves as a generic representation for each individual clip.

We tokenize the visual features using hierarchical k-means. We adjust the number of hierarchy levels d and the number of clusters per level k by visually inspecting the coherence and representativeness of the clusters. We set $d=4$ and $k = 12$, which yields $12^4 = 20736$ clusters in total. Figure 4 illustrates the result of this "vector quantization" process.

For each ASR word sequence, we break the stream of words into sentences by adding punctuation using an off-the-shelf LSTM-based language model. For each sentence, we follow the standard text preprocessing steps from BERT [6] and tokenize the text into WordPieces [33]. We use the same vocabulary provided by the authors of BERT, which contains 30,000 tokens.

Unlike language which can be naturally broken into sentences, it is unclear how to break videos into semantically coherent segments. We use a simple heuristic to address this problem: when an ASR sentence is available, it is associated with starting and ending timestamps, and we treat video tokens that fall into that time period as a segment. When ASR is not available, we simply treat 16 tokens as a segment.

4.3. Model Pre-training

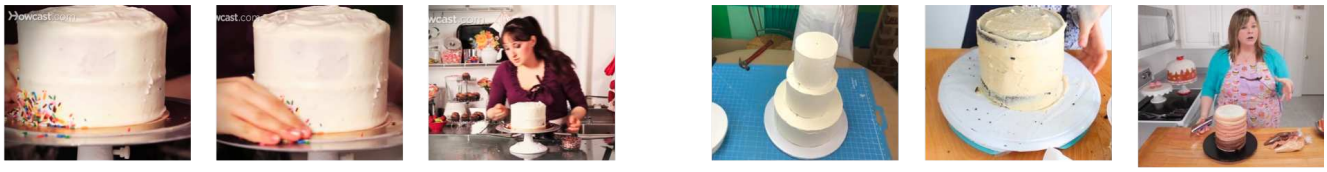
We initialize the BERT weights from a text pre-trained checkpoint. Specifically, we use the $\text{BERT}_{\text{LARGE}}$ model released by the authors of [6], using the same backbone architecture: it has 24 layers of Transformer blocks, where each block has 1024 hidden units and 16 self-attention heads.

We add support for video tokens by appending 20,736 entries to the word embedding lookup table for each of our new "visual words". We initialize these entries with the S3D features from their corresponding cluster centroids. The input embeddings are frozen during pretraining.

Our model training process largely follows the setup of BERT: we use 4 Cloud TPUs in the Pod configuration with a total batch size of 128, and we train the model for 0.5 million iterations, or roughly 8 epochs. We use the Adam optimizer with an initial learning rate of 1e-5, and a linear decay learning rate schedule. The training process takes around 2 days.

4.4. Zero-shot action classification

Once pretrained, the VideoBERT model can be used for "zero-shot" classification on novel datasets, such as YouCook II (By "zero-shot" we mean the model is not



"but in the meantime, you're just kind of moving around your cake board and you can keep reusing make sure you're working on a clean service so you can just get these all out of your way but it's just a really fun thing to do especially for a birthday party."

"apply a little bit of butter on one side and place a portion of the stuffing and spread evenly cover with another slice of the bread and apply some more butter on top since we're gonna grill the sandwiches."

Figure 4: Examples of video sentence pairs from the pretraining videos. We quantize each video segment into a token, and then represent it by the corresponding visual centroid. For each row, we show the original frames (left) and visual centroids (right). We can see that the tokenization process preserves semantic information rather than low-level visual appearance.

trained on YouCook II data nor with the same label ontology used in YouCook II). More precisely, we want to compute $p(y|x)$ where x is the sequence visual tokens, and y is a sequence of words. Since the model is trained to predict sentences, we define y to be the fixed sentence, "`now let me show you how to [MASK] the [MASK]`," and extract the verb and noun labels from the tokens predicted in the first and second masked slots, respectively. See Figure 5 for some qualitative results.

For quantitative evaluation, we use the YouCook II dataset. In [37], the authors collected ground truth bounding boxes for the 63 most common objects for the validation set of YouCook II. However, there are no ground truth labels for actions, and many other common objects are not labeled. So, we collect action and object labels, derived from the ground truth captions, to address this shortcoming. We run an off-the-shelf part-of-speech tagger on the ground truth captions to retrieve the 100 most common nouns and 45 most common verbs, and use these to derive ground truth labels. While VideoBERT's word piece vocabulary gives it the power to effectively perform open-vocabulary classification, it is thus more likely to make semantically correct predictions that do not exactly match the more limited ground truth. So, we report both top-1 and top-5 classification accuracy metrics, where the latter is intended to mitigate this issue, and we leave more sophisticated evaluation techniques for future work. Lastly, if there is more than one verb or noun associated with a video clip, we deem a prediction correct if it matches any of those. We report the performance on the validation set of YouCook II.

Table 1 shows the top-1 and top-5 accuracies of VideoBERT and its ablations. To verify that VideoBERT actually makes use of video inputs, we first remove the video inputs to VideoBERT, and use just the language

Top verbs: make, assemble, prepare
Top nouns: pizza, sauce, pasta

Top verbs: make, do, pour
Top nouns: cocktail, drink, glass

Top verbs: make, prepare, bake
Top nouns: cake, crust, dough

Figure 5: Using VideoBERT to predict nouns and verbs given a video clip. See text for details. The video clip is first converted into video tokens (two are shown here for each example), and then visualized using their centroids.

Method	Supervision	verb top-1 (%)	verb top-5 (%)	object top-1 (%)	object top-5 (%)
S3D [34]	yes	16.1	46.9	13.2	30.9
BERT (language prior)	no	0.0	0.0	0.0	0.0
VideoBERT (language prior)	no	0.4	6.9	7.7	15.3
VideoBERT (cross modal)	no	3.2	43.3	13.1	33.7

Table 1: Action classification performance on YouCook II dataset. See text for details.

Method	Data size	verb top-1 (%)	verb top-5 (%)	object top-1 (%)	object top-5 (%)
VideoBERT	10K	0.4	15.5	2.9	17.8
VideoBERT	50K	1.1	15.7	8.7	27.3
VideoBERT	100K	2.9	24.5	11.2	30.6
VideoBERT	300K	3.2	43.3	13.1	33.7

Table 2: Action classification performance on YouCook II dataset as a function of pre-training data size.

model $p(y)$ to perform prediction. We also use the language prior from the text-only BERT model, that was not fine-tuned on cooking videos. We can see that VideoBERT significantly outperforms both baselines. As expected, the language prior of VideoBERT is adapted to cooking sentences, and is better than the vanilla BERT model.

We then compare with a fully supervised classifier that was trained using the training split of YouCook II. We use the pre-computed S3D features (same as the inputs to VideoBERT), applying average pooling over time, followed by a linear classifier. Table 1 shows the results. As we can see, the supervised framework outperforms VideoBERT in top-1 verb accuracy, which is not surprising given that VideoBERT has an effectively open vocabulary. (See Figure 5 for an illustration of the ambiguity of the action labels.) However, the top-5 accuracy metric reveals that VideoBERT achieves comparable performance to the fully supervised S3D baseline, without using any supervision from YouCook II, indicating that the model is able to perform competitively in this "zero-shot" setting.

4.5. Benefits of large training sets

We also studied the impact of the size of the pretraining dataset. For this experiment, we take random subsets of 10K, 50K and 100K videos from the pretraining set, and pretrain VideoBERT using the same setup as above, for the same number of epochs. Table 2 shows the performance. We can see that the accuracy grows monotonically as the amount of data increases, showing no signs of saturation. This indicates that VideoBERT may benefit from even larger pretraining datasets.

4.6. Transfer learning for captioning

We further demonstrate the effectiveness of VideoBERT when used as a feature extractor. To extract features given only video inputs, we again use a simple fill-in-the-blank task, by appending the video tokens to a template sentence "`now let's [MASK] the [MASK] to the [MASK], and then [MASK] the [MASK].`" We extract the features for the video tokens and the masked out text tokens, take their average and concatenate the two together, to be used by a supervised model in a downstream task.

We evaluate the extracted features on video captioning, following the setup from [39], where the ground truth video segmentations are used to train a supervised model mapping video segments to captions. We use the same model that they do, namely a transformer encoder-decoder, but we replace the inputs to the encoder with the features derived from VideoBERT described above. We also concatenate the VideoBERT features with average-pooled S3D features; as a baseline, we also consider using just S3D features without VideoBERT. We set the number of Transformer block layers to 2, the hidden unit size to 128, and Dropout probability to 0.4. We use a 5-fold cross validation on the training split to set the hyper-parameters, and report performance on the validation set. We train the model for 40K iterations with batch size of 128. We use the same Adam optimizer as in VideoBERT pre-training, and set the initial learning rate to 1e-3 with a linear decay schedule.

Table 3 shows the results. We follow the standard practice in machine translation and compute BLEU and METEOR scores micro-averaged at corpus level, and also report ROUGE-L [14] and CIDEr [29] scores. For the baseline method [39], we recompute the metrics using the predictions provided by the authors. We can see that VideoBERT consistently outperforms the S3D baseline, especially for CIDEr. We can also see that cross-modal pretraining outperforms the video-only version. Furthermore, by concatenating the features from VideoBERT and S3D, the model achieves the best performance across all metrics[1].

Figure 6 shows some qualitative results. We note that the predicted word sequence is rarely exactly equal to the ground truth, which explains why the metrics in Table 3 (which measure n-gram overlap) are all low in absolute value. However, semantically the results seem reasonable.

[1]The metrics used by [39] are macro-averaged at video level and may suffer from undesirable sparsity artifacts. Using their provided evaluation code, VideoBERT + S3D has B@4 of 1.79, and METEOR of 10.80.

Method	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
Zhou *et al*. [39]	7.53	3.84	11.55	27.44	0.38
S3D [34]	6.12	3.24	9.52	26.09	0.31
VideoBERT (video only)	6.33	3.81	10.81	27.14	0.47
VideoBERT	6.80	4.04	11.01	27.50	0.49
VideoBERT + S3D	**7.59**	**4.33**	**11.94**	**28.80**	**0.55**

Table 3: Video captioning performance on YouCook II. We follow the setup from [39] and report captioning performance on the validation set, given ground truth video segments. Higher numbers are better.

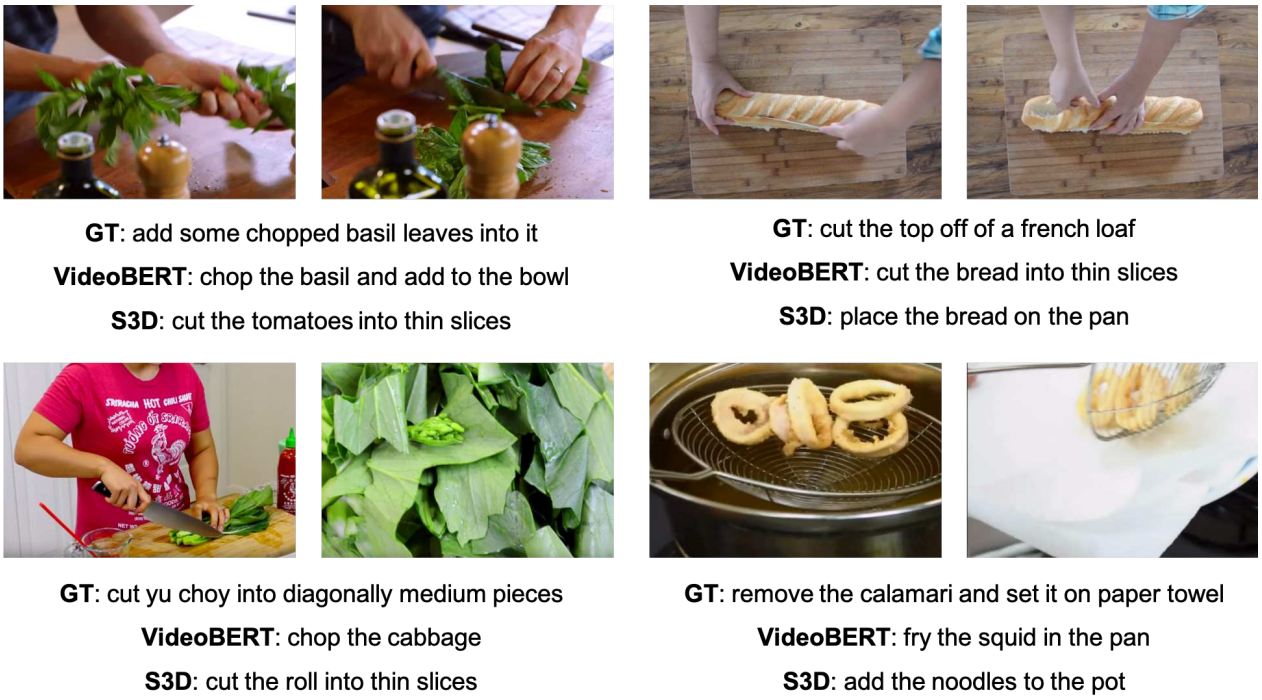


Figure 6: Examples of generated captions by VideoBERT and the S3D baseline. In the last example, VideoBERT fails to exploit the full temporal context, since it misses the paper towel frame.

5. Discussion and conclusion

This paper adapts the powerful BERT model to learn a joint visual-linguistic representation for video. Our experimental results demonstrate that we are able to learn high-level semantic representations, and we outperform the state-of-the-art for video captioning on the YouCook II dataset. We also show that this model can be used directly for open-vocabulary classification, and that its performance grows monotonically with the size of training set.

This work is a first step in the direction of learning such joint representations. For many applications, including cooking, it is important to use spatially fine-grained visual representations, instead of just working at the frame or clip level, so that we can distinguish individual objects and their attributes. We envision either using pretrained object detection and semantic segmentation models, or using unsupervised techniques for broader coverage. We also want to explicitly model visual patterns at multiple temporal scales, instead of our current approach, that skips frames but builds a single vocabulary.

Beyond improving the model, we plan to assess our approach on other video understanding tasks, and on other domains besides cooking. (For example, we may use the recently released COIN dataset of manually labeled instructional videos [26].) We believe the future prospects for large scale representation learning from video and language look quite promising.

Acknowledgements. We would like to thank Jack Hessel, Bo Pang, Radu Soricut, Baris Sumengen, Zhenhai Zhu, and the BERT team for sharing amazing tools that greatly facilitated our experiments; Justin Gilmer, Abhishek Kumar, David Ross, and Rahul Sukthankar for helpful discussions. Chen would like to thank Y. M. for inspiration.

References

[1] YouTube Data API. https://developers.google.com/youtube/v3/docs/captions. 5

[2] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016. 3

[3] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *NeurIPS*, 2016. 3

[4] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. In *ICLR*, 2018. 2

[5] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *ICML*, 2018. 2

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3, 5

[7] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 2

[8] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 3

[9] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 5

[10] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-Captioning events in videos. In *ICCV*, 2017. 2, 3

[11] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Baby talk: Understanding and generating image descriptions. In *CVPR*, 2011. 3

[12] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019. 3

[13] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv:1804.01523*, 2018. 2

[14] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004. 7

[15] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *CVPR*, 2018. 3

[16] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. In *NAACL*, Mar. 2015. 3

[17] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016. 2

[18] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. 3

[19] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Yan Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *TPAMI*, 2019. 2

[20] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *CVPR*, 2016. 3

[21] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016. 3

[22] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, 2018. 3

[23] Marc Aurelio Ranzato and Alex Graves. Deep unsupervised learning. NIPS Tutorial, 2018. 3

[24] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017. 1

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 5

[26] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. COIN: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, 2019. 8

[27] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *CVPR*, 2018. 2

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 3

[29] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 7

[30] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016. 2

[31] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016. 2

[32] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016. 2

[33] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016. 5

[34] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. In *ECCV*, 2018. 5, 7, 8

[35] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*, 2016. 2

[36] Hang Zhao, Zhicheng Yan, Heng Wang, Lorenzo Torresani, and Antonio Torralba. Slac: A sparsely labeled dataset for action classification and localization. *arXiv preprint arXiv:1712.09374*, 2017. 2

[37] Luowei Zhou, Nathan Louis, and Jason J Corso. Weakly-supervised video object grounding from text by loss weighting and object interaction. In *BMVC*, 2018. 6

[38] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 2, 3, 5

[39] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *CVPR*, 2018. 2, 3, 7, 8