Learning to Learn Words from Narrated Video

Dídac Surís\textsuperscript{1*} Dave Epstein\textsuperscript{1*} Heng Ji\textsuperscript{2} Shih-Fu Chang\textsuperscript{1} Carl Vondrick\textsuperscript{1}
\textsuperscript{1}Columbia University \textsuperscript{2}UIUC
expert.cs.columbia.edu

Abstract

When we travel, we often encounter new scenarios we have never experienced before, with new sights and new words that describe them. We can use our language-learning ability to quickly learn these new words and correlate them with the visual world. In contrast, language models often do not robustly generalize to novel words and compositions. We propose a framework that learns how to learn text representations from visual context. Experiments show that our approach significantly outperforms the state-of-the-art in visual language modeling for acquiring new words and predicting new compositions. Model ablations and visualizations suggest that the visual modality helps our approach more robustly generalize at these tasks.

1. Introduction

The first time you travel to a new country, you are inundated with unfamiliar sights and words. However, the more you travel, the better you become at acquiring words and phrases to describe visual surroundings. Figure 1 shows one example of our ability to do this. Although you may not be familiar with the words “ghee” or “roti,” you are able to learn them by using the structure of the scene and knowledge of other words.

However, the relationship between vision and language is often subtle, and robustly learning their correlation is challenging. Vision and language models frequently learn representations by embedding images and words into a common space, such as learning to embed the word “apple” close to the embedding of an image for “apple”. This approach has been hugely successful on a number of vision language benchmarks \cite{13, 26, 3, 46, 53, 27, 44, 33}. However, when models are trained to maximize the likelihood over their training set, they necessarily assign low probability to rare examples. Consequently, models that learn the correlation between vision and language are often unable to generalize to new words and new compositions that were not encountered during training \cite{16}.

In this paper, we propose a method that learns to learn the correlations between the modalities. Our main idea is to meta-learn a policy to learn how to acquire information from visual-linguistic pairs. Rather than learning that an image of an apple is correlated to the word apple, we instead learn how to learn this relationship from visual structure and known contextual words.

Our experiments suggest that our framework learns a strong policy for acquiring new words from visual context. We train and evaluate our approach on the EPIC-Kitchens dataset \cite{8}, which has a large diversity of natural scenes and a long-tail distribution of nouns and verbs. After learning the policy, the model can receive a stream of images and corresponding short phrases containing unfamiliar words. Our model is able to learn the novel words and point to them to describe other scenes. Visualizations of the model suggest strong cross-modal interaction from language to visual inputs and vice versa.

A key advantage of our approach is that it is able to effectively generalize outside of the training set, e.g. to unseen compositions of nouns and verbs. Our experiments obtain significant improvement for compositional generalization, outperforming the state-of-the-art in visual language models by over fifteen percent when the compositions are new. Our results also suggest the underlying language model is more robust for new visual instances of familiar words.

1“ghee” is the butter on the knife, and “roti” is the bread in the pan
* indicates equal contribution

Figure 1: Can you figure out what “ghee” and “roti” are? The answer is in the footnote. Although the words “ghee” and “roti” may be unfamiliar to you, you are able to leverage the structure of the visual world and knowledge of other words to learn new words. In this paper, we propose a model that learns how to learn words from visual context.
Our primary contribution is a framework that learns how to learn the correlation between vision and language, which is able to robustly generalize to both new words and new compositions. In Section 2, we review related work. In Section 3, we present in detail our approach to learn how to learn words from visual episodes. In Section 4, we analyze the performance of our approach and ablate various components with a set of qualitative and quantitative experiments to study its effectiveness. We release all code and models to facilitate future research on the website.

2. Related Work

Visual language modeling: Recent advances in natural language processing have yielded neural language models able to learn from large amounts of text, such as BERT [10], ELMo [35] and GPT [37], which have achieved state-of-the-art results on a variety of linguistic tasks. Recently, vision has been tightly integrated with these language models in a series of papers [26, 44, 29, 38, 27, 53, 48, 7, 45, 46, 3] which are then evaluated on tasks such as visual question answering [2]. Such models often use paired vision and language data to learn the correlation between modalities. In this paper, we build upon this line of research, but instead learn to learn a visual language model. We compare against both BERT and the best performing visual language models as highly competitive baselines, which we outperform.

Generalizing to compositions: Due to the diversity of the visual world, there has been extensive work in computer vision on learning compositional representations for objects and attributes [32, 22, 31, 34] as well as for objects and actions [24, 34]. In natural language processing, high-level compositions have also been studied [9, 12]. The work most related to ours is [25], which also develops a meta-learning framework for compositional generalization. However, unlike [25], our approach is designed for realistic language and challenging natural images.

Generalization to new words: Previous work has considered generalizing with little or no labeled data in a variety of tasks, including image classification [52, 47, 43], translating between a language pair never seen explicitly during training [23] or understanding text from a completely new language [5], among others. We focus on modeling new out-of-vocabulary (OOV) words or compositions unseen during training, but that are seen once during testing. A simple approach to model OOV words involves fine-tuning in a few-shot setting [1, 19]. [20] proposes a meta-learning approach, but unlike us, they perform gradient updates on the new words, do not use visual information, and constrain the model by explicitly designing it for the few-shot scenario. We incorporate OOV words not just as an input to the system [19, 41, 36], but also as output.

Learning to learn: Meta-learning is a rapidly growing area of investigation. Different approaches include learning to quickly learn new tasks by finding a good initialization [14, 28], learning efficient optimization policies [42, 6, 4, 39, 28], learning to select the correct policy or oracle in what is also known as hierarchical learning [15, 21], and others [11, 30]. In this paper, we apply meta-learning to acquire new words and compositions from narrated video.

3. Learning to Learn Words

We present an approach that learns how to learn word representations from visual context. In this section, we formulate the learning task and describe a neural network architecture that instantiates the visual language model.

3.1. Episodes for Generalization

Most visual language models learn representations by estimating the correlation between image and language pairs. While this learns rich word and visual representations, the trained model is unable to robustly generalize to situations outside its training set because maximum likelihood training necessarily assigns low probability to rare events.

We propose a meta-learning approach that learns to learn a visual language model for generalization. We construct
training episodes containing multiple examples, where each example is a text-image pair for a different scene. Figure 3 illustrates some episodes. To build an episode, we first sample a target example, which is an image and text pair, and mask some of its word tokens. We then pick reference examples from other pairs, some of which contain tokens masked in the target. For notational convenience, we write the episode as the set \( \{v_1, \ldots, v_i, w_{i+1}, \ldots, w_j\} \) where \( v_i \) is an image and \( w_i \) is a word token in the episode. During learning, we input an episode to the model, and train the model to reconstruct one or more masked words \( w_i \).

We build episodes that have substantial generalization gaps, allowing us to explicitly train the model for generalization. Some episodes may contain new words, requiring the model to learn a policy for acquiring the word from reference examples and using it to describe the target scene in the episode. Other episodes may contain familiar words but novel compositions in the target. In both cases, the model will need to generalize to target examples by using the reference examples in the episode. Since we train our model on a distribution of episodes instead of a distribution of examples, and each episode contains new instances, words, and compositions, the learned policy will be robust at generalizing to testing episodes from the same distribution.

### 3.2. Model

To create our model, we use a neural network that consumes an entire episode, and predicts the words for the target image in the episode. By propagating the gradient from the target scene back to other examples in the episode, we can directly train the model for generalization.

The neural network architecture (Figure 4) needs to be able to receive multiple images and text pairs, and produce predictions for each input. We parameterize the neural network with a stack of self-attention-based transformers [50]. Transformers are able to capture pairwise relationships between each element in the episode, enabling rich contextual representations [10]. The input to the model is the episode set \( \{v_1, \ldots, w_j\} \), and the stack of transformers will produce hidden representations \( \{h_1, \ldots, h_j\} \) for each image and word in the episode.

**Input Encoding:** We first encode each word and image into a fixed-length vector representation before we input them into the transformer. To embed input words, we use an \( N \times d \) word embedding matrix \( \phi_w \), where \( N \) is the size of the vocabulary considered by the tokenizer. To embed visual regions, we use a convolutional network \( \phi_v(\cdot) \) over image regions (we use ResNet-18 pre-trained on ImageNet [40, 18]). Visual regions can be the entire image in addition to any objects detected in the image. To augment the input encoding with both information about the modality and the positional information (word index for text, relative bounding box coordinates for image), we also translate the encoding by a learned vector:

\[
\phi_{\text{img}}(v_i) = \phi_v(v_i) + \phi_{\text{bbox}}(v_i) + \phi_{\text{mod}}(\text{IMG}) + \phi_d(v_i) \\
\phi_{\text{txt}}(w_j) = \phi_v^T w_j + \phi_{\text{pos}}(w_j) + \phi_{\text{mod}}(\text{TXT}) + \phi_d(w_j)
\]

where \( \phi_{\text{bbox}} \) encodes the bounding box of \( v_i \), \( \phi_{\text{pos}} \) encodes the word position of \( w_j \), \( \phi_{\text{mod}} \) encodes the modality and \( \phi_d \) encodes the index of the example.

**Transformer Architecture:** We input these word and visual encodings into the transformer stack. One transformer consists of a multi-head attention block followed by a linear projection, which outputs a hidden representation at each location, and is passed in series to the next transformer layer. Let \( H^z = \mathbb{R}^{d \times j} \) be the \( d \)-dimensional hidden vectors at layer \( z \). The transformer first computes vectors for queries \( Q = W_q^z H^z \), keys \( K = W_k^z H^z \), and values \( V = W_v^z H^z \) where each \( W_a \in \mathbb{R}^{d \times d} \) is a matrix of learned parameters. Using these queries, keys, and values, the transformer computes the next layer representation by attending to all elements in the previous layer:

\[
H^{z+1} = SV \quad \text{where} \quad S = \text{softmax} \left( \frac{QK^T}{\sqrt{d}} \right).
\]

In practice, the transformer uses multi-head attention, which repeats Equation 2 once for each head, and concate-
3.3. Learning Objectives

To train the model, we mask input elements from the episode, and train the model to reconstruct them. We use three different complementary loss terms for learning.

**Pointing to Words:** We train the model to “point” to other words within the same episode. Let \( w_i \) be the target word that we wish to predict, which is masked out. Furthermore, let \( w_{i'} \) be the same word which appears in a reference example in the episode \( (i' \neq i) \). To fill in the masked position \( w_i \), we would like the model to point to \( w_{i'} \), and not any other word in the reference set.

We estimate similarity between the \( i \)th input and the \( j \)th input, which we define by the matrix \( A \):

\[
\log A_{ij} = f(h_i)^T f(h_j)
\]

where \( f(h_i) \in \mathbb{R}^d \) is a linear projection of the hidden representation for the \( i \)th element.

Pointing to the right word within the episode corresponds to maximizing the similarity between the masked position and the true reference position, which we implement as a cross-entropy loss:

\[
L_{\text{point}} = -\log \left( \frac{A_{i'i}}{\sum_k A_{ik}} \right)
\]

Minimizing the above loss over a large number of episodes will cause the neural network to produce a representation of the masked word \( h_i \) such that it is closer to its reference word \( h_{i'} \) than the negative examples in the same episode.

Other similarity matrices are possible. The similarity matrix in Equation 3 will cause the model to fill in a masked word by pointing to another contextual representation. However, we can also define a similarity matrix that points to the input word embedding instead. To do this, the matrix is defined as \( \log A_{ij} = f(h_i)^T \phi_w(w_j) \). This prevents the model from solely relying on the context and forces it to specifically attend to the reference word, which our experiments will show helps generalizing to new words.

**Word Cloze:** We additionally train the model to reconstruct words by directly predicting them. Given the contextual representation of the masked word \( h_i \), the model predicts the missing word by multiplying its contextual representation with the word embedding matrix, \( \hat{w}_i = \phi_w^T h_i \).

We then train with cross-entropy loss between the predicted word \( \hat{w}_i \) and true word \( w_i \), which we write as \( L_{\text{close}} \). This objective is the same as in the original BERT [10].

**Visual Cloze:** In addition to training the word representations, we train the visual representations on a cloze task. However, whereas the word cloze task requires predicting the missing word, generating missing examples is challenging. Instead, we impose a metric loss such that a linear projection of \( h_i \) is closer to \( \phi_v(v_i) \) than \( \phi_v(v_{k \neq i}) \). We use the triplet loss [51] with cosine similarity and a margin of one. We write this loss as \( L_{\text{vision}} \). This loss is similar to the visual loss used in state-of-the-art visual language models [7].

**Combination:** Since each objective is complementary, we train the model by optimizing the neural network parameters to minimize the sum of losses:

\[
\min_{\Omega} \mathbb{E} \left[ L_{\text{point}} + \alpha L_{\text{close}} + \beta L_{\text{vision}} \right]
\]

where \( \alpha \in \mathbb{R} \) and \( \beta \in \mathbb{R} \) are scalar hyper-parameters to balance each loss term, and \( \Omega \) are all the learned parameters. We sample an episode, compute the gradients with back-propagation, and update the model parameters by stochastic gradient descent.

3.4. Information Flow

For the model to use reference examples in the episode to describe the target example, information needs to flow from reference examples to the target example. Since the transformer computes attention between elements, we can control how information flows in the model by constraining the attention. We implement this as a mask on the attention: \( H^{t+1} = (S \circ M) V \) where \( M_{ij} \) is a binary mask to indicate whether information can flow from element \( j \) to \( i \). Several masks \( M \) are possible.

**Isolated attention:** By setting \( M_{ij} = 1 \) iff \( i \) and \( j \) belong to the same example in the episode, examples can only attend within themselves. This is identical to running each example separately through the model, which disables meta-learning from episodes.

**Full attention:** By unconditionally setting \( M_{ij} = 1 \), attention is fully connected and every element can attend to all other elements.

**Target-to-reference attention:** We can constrain the attention to only allow the target elements to attend to the reference elements, and prevent the reference elements from communicating across each other. To do this, \( M_{ij} = 1 \) iff \( i \) and \( j \) are from the same example or \( i \) is a target element.
In our experiments, optimization typically takes one week, with a learning rate of $3 \times 10^{-5}$ and $\epsilon = 0.0001$. For all our experiments we use four transformers ($Z = 4$) and four heads. All the models are trained with the Adam optimizer, with a learning rate of $3 \times 10^{-5}$ and $\epsilon = 0.0001$. In our experiments, optimization typically takes one week on a single GPU.

In training, we mask out text tokens $\frac{1}{8}$ of the time and image tokens $\frac{1}{6}$ of the time. Following [10], a masked text token gets assigned a special [MASK] token 80% of the time, a random word token 10%, and remains unchanged 10%. Similarly, we zero out image tokens 90% of the time and leave them unaltered 10%.

We construct episodes by first randomly sampling a target example from the training set. We then randomly select between 0 and $k_+ = 2$ text tokens as targets, which will be masked with probability 1. For each one of these tokens, we randomly add to the episode another example in the training set whose text contains the token. We then randomly add between 0 and $k_- = 2$ negative examples (distractors) to the episode, which do not contain any of the target tokens in their text.

We randomly shuffle examples in an episode before feeding them to the model. Then, we combine them with indicator tokens to demarcate examples: $[IMG] v^1_1, \ldots, v^1_{J_1} [SEP] \ldots [SEP] v^k_1, \ldots, v^k_{J_k} [TXT] \; w^1_1, \ldots, w^1_{J_1} [SEP] \ldots [SEP] w^k_1, \ldots, w^k_{J_k} [SEP]$. We denote example index with the superscript and $k = \text{rand}(0, k_+) + \text{rand}(0, k_-) + 1$ is the number of examples in the episode. $I_i$ and $J_i$ are the number of image and text tokens in the $i$th example of the episode.

4. Experiments

The goal of our experiments is to analyze our model’s ability to learn language from scratch and generalize to new instances, new compositions, and new words. Importantly, we train our model only on our video dataset, without any language pretraining. We present several quantitative and qualitative experiments using realistic visual data. We call our approach EXPERT.2

4.1. Dataset

We evaluate our approach using realistic, natural images and their corresponding text narrations. A common evaluation issue in vision and language tasks is that one modality is unnecessary [17]. We therefore evaluate our approach on tasks where both modalities are necessary.

**EPIC-Kitchens** is a large dataset consisting of 39,594 video clips across 32 kitchens. Each clip has a short text narration, which spans 314 verbs and 678 nouns, as well as other word types. EPIC-Kitchens contains natural language, but because of its non-exhaustive description of the scene, visual information is also needed to model and predict missing words. EPIC-Kitchens is challenging due to the complexities of unscripted video. Although the images are also annotated with ground-truth object and action cate-

---

2Episodic Cross-Modal Pointing for Encoder Representations from Transformers
4.2. Baselines

We compare our approach against state-of-the-art approaches in visual and language models.

BERT is a large language model trained on more than 3,000M words and recently obtained state-of-the-art performance across several natural language processing tasks [10]. We consider two variants: firstly, we download the pre-trained model and fine-tune it on our training set. Secondly, we also train BERT from scratch on our data. We use BERT as strong language-only baseline.

BERT with Vision refers to the family of visually grounded language models [26, 3, 46, 53, 27, 44, 33, 7], which add visual pre-training to BERT. We experimented with several of them on our tasks, and we report the one that performs the best [7]. Same as our model, this baseline does not use language pretraining.

We also breakdown performance of our model by removing different components. Tgt-to-ref attn, Via-vision attn, and Full attn indicate the choice of attention mask; the base one is the isolated attention. Input pointing indicates the choice of pointing to the input encodings in addition to contextual encodings. Unless otherwise noted, EXPERT refers to the variant trained with via-vision attention.

4.3. New Instances

We first analyze how our model learns to generalize to new image instances, keeping the vocabulary and compositions the same between training and testing. We evaluate our system using the cloze test [49], also known as a “fill in the blank” test. We mask out one word, and report the top-5 classification accuracy to predict the correct missing word. To obtain predictions from our model, we input an episode that only contains a target image and a partially masked text narration without a reference example.

Table 1 shows that our approach learns a more robust model than baselines. While visual information helps, how to use vision information matters. Our approach, which learns to learn a visual language model from episodes, outperforms other visual-language models in predicting nouns and verbs by 9% and 4% respectively. Our approach also obtains strong performance for all parts of speech (PoS).

4.4. New Compositions

We next analyze how well our model generalizes to compositions. In this task, the vocabulary is the same between train and test, but compositions of words may be new. We evaluate both seen compositions, which were encountered during training, and unseen compositions, which were not. To quantify performance, we mask out both a noun and a verb, and consider a prediction correct only if both the predicted noun and verb are correct. To obtain predictions from our model, we follow the same setup as before where we input an empty reference set and one target example.

Table 2 shows that our model improves visual language modeling for both seen and new compositions. Our approach outperforms text-only baselines by up to 32%, showing the importance of vision for this task. Our approach also outperforms state-of-the-art visual language models by up 15%, suggesting that learning how to learn the relationship between vision and text produces better representations.

Moreover, for all the baselines, performance significantly degrades on novel compositions, with drop in performance of 19%, revealing poor generalization outside of the training set. The performance of our approach, however, remains relatively stabler, dropping only 10%, suggesting that our approach is more robust at generalization.

4.5. New Words

Our experiments so far have suggested that “learning to learn” words from vision leads to better generalization. We now analyze how well our model is able to learn to acquire words from one scene, and use them to describe another scene. As word embeddings for new words are randomly distributed, directly predicting them is unreasonable. Instead, our model produces the new words by pointing to them in the reference set. Since the new word embeddings are random vectors, our model needs to leverage the re-
Figure 6: **Pointing to New Words:** We show examples where the model encounters new words in the episode. The dark green in the target example indicates the ground-truth new word, which is masked out. The model predicts that word by pointing to words in the reference set, and the weight of each pointer is visualized by the shade of the box around each word (weight < 3% is omitted). In bottom right, we show an error where the model predicts the plate is being placed, not grabbed.

![Table 3: Acquiring New Words](image)

Table 3: **Acquiring New Words:** We test our model’s ability to acquire new words at test time. The difficulty of this task varies with the number of distractor examples in the reference set. We show top-1 accuracy results on both 1:1 and 2:1 ratios of distractors to positives.

**Figure 7:** **New Word Accuracy versus Distractor Ratio:** As more distractors are added, the problem becomes more difficult, causing performance for all models to go down. However, EXPERT decreases at a lower rate than baselines.

motion information would help EXPERT. Figure 7 shows that, even as the number of distractor examples increases, the performance of our model is relatively strong. As expected, EXPERT accuracy decreases with the number of negatives, but it does so at a slower rate than the baselines. Specifically, EXPERT outperforms baselines by 18% with one distractor example, and by 36% with ten. This shows that our model remains relatively robust compared to baselines.

Figure 8 shows some examples of how the model handles predicting words which it has seen neither in training nor in the reference set. In this case, since the word cannot be found in the reference set, the model cannot point to it, so we examine language model predictions of the masked new words. The model still makes sensible predictions. For example, when the model has never seen the word “oven”, it describes the oven as a “drawer” instead.

**4.6. Analysis**

In this section, we analyze why EXPERT obtains better performance than competitive baselines.

**Does EXPERT use vision?** We take our complete
model, trained with both text and images, and withhold images at test time. Performance drops to nearly chance, suggesting that EXPERT uses visual information to predict words and disambiguate between similar language contexts.

**What visual information does EXPERT use?** To study this, we withhold one visual region at a time from the episode and find the regions that cause the largest decrease in prediction confidence. Figure 9 visualizes these regions, showing that removing the object that corresponds to the target word causes the largest drop in performance. This suggests that the model is correlating these words with the right visual region, without direct supervision.

**How does information flow through EXPERT?** Our model makes predictions by attending to other elements within examples and episodes. To analyze the learned attention, we take the variant of EXPERT trained with full pairwise attention and measure changes in accuracy as we disable query-key interactions one by one. Figure 10 shows which connections are most important for performance. This analysis reveals a strong dependence on cross-modal attention, where information flows from text to image in the first layer back to text in the last layer.

**How does EXPERT disambiguate multiple new words?** To answer this, we evaluate our model on episodes that contain five new words in the reference set, only one of which matches the target token. Our model obtains an accuracy of 56% in this scenario, while randomly picking one of the novel words would give 20%. This shows that our model is able to discriminate between many new words in an episode. We also evaluate the fine-tuned BERT model in this same setting, where it obtains a 37% accuracy, significantly worse than our model. This suggests that vision is important for disambiguating new words.

5. **Discussion**

Vision and language models typically learn the correlation between the two modalities, but the relationship between vision and language is often subtle. In this paper, we instead learn how to learn this correlation, which our experiments show improves performance for new instances, words, and compositions. We believe our approach would benefit from motion information, specifically in modeling actions and verbs. Our results suggest that, by creating episodes specifically for generalization, one can improve other types of abstraction.

**Acknowledgements:** We thank Alireza Zareian, Bobby Wu, Spencer Whitehead, Parita Pooj and Boyuan Chen for helpful discussion. Funding for this research was provided by DARPA GAILA HR00111990058. We thank NVidia for GPU donations.
Appendix

A. EPIC-Kitchens Train-Test Split

The EPIC-Kitchens dataset does not provide action narrations for its test set, which we need for evaluation. We therefore create our own train-test split from the dataset. We aim to generate an 80-20 train-test split. We select a small subset of verbs and nouns to remove entirely from the training set, as well as verb-noun compositions. We remove around 4% of nouns and verbs and withhold them for testing, and around 10% of all compositions. These are not uniformly distributed, and removing a verb or noun entirely from the training set often results in withholding a disproportionate amount of data. Therefore, with these parameters, we end up with around a 81-19 split. We release the complete dataset split, as well as code for training with it, to facilitate future research in this area.

Below, we list the words and compositions that are withheld from the training set, but present during testing.

List of new nouns:
1. avocado
2. counter
3. hand
4. ladle
5. nesquik
6. nut
7. oven
8. peeler
9. salmon
10. sandwich
11. seed
12. onion
13. corn

List of new verbs:
1. fry
2. gather
3. grab
4. mash
5. skin
6. watch

List of new verb/noun compositions:
Since there are around 400 new compositions, we include only the 20 most common here.
1. close oven
2. cut peach
3. dry hand
4. fry in pan
5. grab plate
6. open cupboard
7. open oven
8. pick up sponge
9. put onion
10. put in oven
11. put spoon
12. remove garlic
13. rinse hand
14. skin carrot
15. stir pasta
16. take plate
17. wash hand
18. wash knife
19. wipe counter
20. wipe hand

B. Language Model Results

We show EXPERT’s outputs when given a sentence containing a new composition of verb and noun. The verb and noun are masked, and we ask EXPERT to make language model predictions at these locations (as in the standard BERT cloze task setting). Results are shown in Figure 11.

References
[37] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2


[46] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A Joint Model for Video and Language Representation Learning. apr 2019. 1, 2, 6


