Learning to Learn Words from Visual Scenes

Dídac Surís[1⋆], Dave Epstein[1⋆], Heng Ji[2], Shih-Fu Chang[1], and Carl Vondrick[1]

[1]Columbia University [2]UIUC
expert.cs.columbia.edu

In this supplementary material we present some additional information on the data, training procedure and results. In Section 1 we elaborate on the procedure we follow to create a train-test split that allows for testing of new words and compositions. In Section 2 we detail the different ways we have of controlling the information flow. In Section 3 we show qualitative results of our model on predicting new compositions, as well as extended tables of language model results when ablating model design choices. In Section 4, we provide further detail on our implementation procedure. We provide more visualizations for the models trained on the Flickr30k dataset in Section 5.

1 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 following their conventions. 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 will release this dataset splits for others to build on our this work.

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

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

⋆ Equal contribution

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

Since the Flickr30k dataset has a larger vocabulary than EPIC-Kitchens and thus a larger train-test split, we only list here the new verbs and nouns, and not the new compositions.

Flickr30k new nouns: 3d, a, A&M, ad, Adidas, Africa, AIDS, airborne, Airways, Alaska, America, american, Americans, Amsterdam, Angeles, Asia, ax, B, badminton, bale, barrier, Batman, batman, Bay, be, Beijing, Berlin, big, Birmingham, Blue, Boston, bottle, Brazil, Bridge, Britain, British, british, Bush, c, California, Calvin, Canada, canadian, Canyon, card, Carolina, case, catholic, cause, cd, cello, Celtics, Chicago, China, chinatown, Chinese, chinese, christmas, Circus, Claus, Clause, clipper, co, cocktail, colleague, Coney, content, convertible, courtyard, Cruz, cup, cycle, dance, David, Deere, desk, Dior, Disney, dj, Domino, dragster, drum, dryer, DS, dump, east, Easter, eastern, Eiffel, Elmo, elmo, England, Europe, fellow, Florida, Ford, France, Francisco, Gate, gi, Giants, glove, go, God, Golden, golden, graduate, Grand, Great, Haiti, halloween, hedge, Heineken, helicopter, hi, hide, highchair, hispanic, Hollister, Hollywood, hoop, Houston, iMac, India, Indian, Indians, information, ingredient, Iowa, Island, Israel, Italy, Jackson, jam, Japan, Jesus, Jim, Joe, John, Klein, knoll, La, Lakers, Las, latex, layup, legged, liberty, library, Lincoln, locomotive, London, loom, Los, Lynyrd, ma, mariachi, Mets, Mexico, Miami, Michael, Michigan, Mickey, mickey, Mike, Miller, mind, Morgan, Mr, Mrs, Music, muslim, Navy, New, new, NFL, Nintendo, no, nun, NY, NYC, o, Obama, officer, Oklahoma, old, op, oriental, ox, Oxford, p, Pabst, Pacific, Paris, Patrick, Paul, pavement, pc, Penn, pew, piercing, pig, plane, pot, punk, rafting, rain, razor, Red, Renaissance, repairman, research, robe, Rodgers, runway, RV, S, Salvation, San, saris, scrimmage, scrubs, Seattle, second, shooting, shrine, Skynyrd, something, South, Sox, Spain, Spanish, Square, SquarePants, St, St, start, States, Statue, Story, style, superman, surfs, T, t, tech, Texas, texas, the, Thomas, Times, today, tooth, Toronto, Toy, trinket, tv, type, U, UFC, UK, ultimate, Unicef, United, up, US, USA, v, Vegas, Verizon, Volvo, vw, W, Wall, Wars, Washington, Wells, West, White, Wii, wind, windsurfer, Winnie, winter, Wonder, wonder, x, Yankees, yard, yo, yong, York, york.

Flickr30k new verbs: address, am, amused, applaud, armed, baked, bar, be, bearded, blend, boat, bound, broken, build, button, cling, complect, confused,

cooked, costumed, covered, crashing, crowded, crumble, darkened, decorated, deflated, do, dyed, fallen, fenced, file, flying, go, goggle, graze, haircut, handicapped, inflated, injured, juggle, listening, lit, living, looking, magnify, measure, mixed, motorize, mounted, muzzle, muzzled, numbered, oncoming, opposing, organized, patterned, pierced, populated, proclaim, puzzle, restrain, rim, saddle, scratch, seated, secure, sell, shaved, skinned, slump, solder, spotted, sprawl, streaked, strike, stuffed, suited, sweeping, tanned, tattooed, tile, tiled, train, uniformed, up, wheeled, woode, wooded.

2 Information Flow

To use the episode, 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: $H^{z+1} = (S \odot M)V$ where M_{ij} is a binary mask to indicate whether information can flow from element j to i. Several masks M are possible. Figure **??** visualizes them.

(a) 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 equivalent to running each example separately through the model, and optimizing the model with a metric learning loss.

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

(c) 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.

(d) Attention via vision: We can also constrain the attention to only transfer information through vision. Here, $M_{ij} = 1$ if i and j are from the same example, and also $M_{ij} = 1$ if i and j are both images and i is a target element. Otherwise, $M_{ij} = 0$. Information is first propagated from reference text nodes to visual nodes, then propagated from the visual nodes to the target text node.

Some attention mechanisms are more computationally efficient because they do not require computing representations for all pairwise relationships. For full attention, computation scales quadratically with the number of examples. However, for the other attention mechanisms, computation scales linearly with the number of examples, allowing us to efficiently operate on large episodes.

3 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 3.

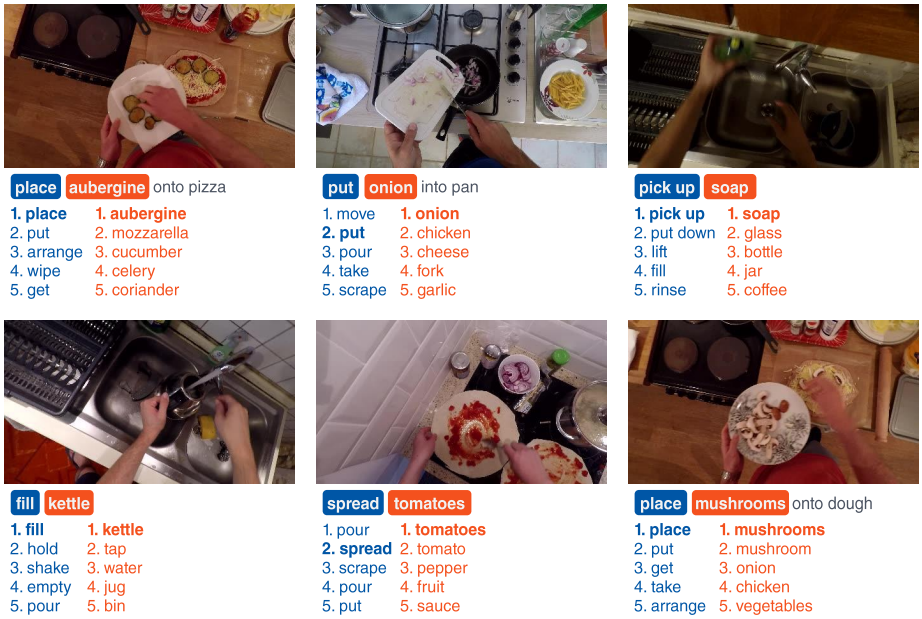

Fig. 1: **Predictions of new compositions:** We show some examples of our model's ability to generalize to new compositions, given only the images shown (and their bounding boxes), as well as the unmasked words in the sentence. We show the top five predictions for each word. As in the paper, we indicate masked words with the dark box, and their predictions below it.

In addition, we provide detailed language modeling performance figures in Tables 1 and 2, breaking accuracy down by model variant.

4 Implementation Details

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 3×10^{-5} and $\epsilon = 0.0001$. In our experiments, optimization typically takes one week on a single GPU.

In training, we mask out text tokens $\frac{1}{3}$ of the time and image tokens $\frac{1}{6}$ of the time. Following [2], 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

	Method	Verbs	Nouns	All PoS
	Chance	0.1	0.1	0.1
	BERT (scratch) [2]	68.2	48.9	57.9
	BERT (pretrained) [2]	71.4	51.5	59.8
	BERT with Vision [1]	77.3	63.2	65.6
EXPERT	Isolated attn	81.2	74.2	66.8
EXPERT	Target-to-ref attn	80.9	69.6	69.8
EXPERT	Via-vision attn	81.9	73.0	74.9
EXPERT	+ Input pointing	80.1	73.2	67.0
EXPERT	Full attn	79.4	68.7	67.3

Table 1: **Acquiring Familiar Words on EPIC-Kitchens:** We report top-5 accuracy at predicting words seen in training for new visual instances.

text contains the token. We then randomly add between 0 and $k_{\text{-}} = 2$ 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]` $v_1^1, \ldots, v_{I_1}^1$ `[SEP]` ... `[SEP]` $v_1^k, \ldots, v_{I_k}^k$ `[TXT]` $w_1^1, \ldots, w_{J_1}^1$ `[SEP]` ... `[SEP]` $w_1^k, \ldots, w_{J_k}^k$ `[SEP]`. We denote example index with the superscript and $k = \text{rand}(0, k_{\text{-}}) + \text{rand}(0, k_{+}) + 1$ 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.

We use the PyTorch framework [3] to implement our model. We base much of our code on a modified version of the Hugging Face PyTorch transformer repository. [1] In particular, we implement our own model class that extends `BertPreTrainedModel`. We write our own input encoding class that extends `BertEmbeddings` and adds visual embedding functionality as well as all the ϕ functions that are added to the word and image region embeddings. We use `hidden_size=384, intermediate_size=1536, num_attention_heads=4, num_hidden_layers=4, max_position_embeddings=512, type_vocab_size=64, vocab_size=32000` (and all other parameters default) to configure all models except the pretrained BERT one, which uses the original weights from `BERT-small`.

When we give an example to the model (whether individually or as part of an episode), we include all bounding boxes provided in the EPIC-Kitchens and Flickr30k datasets as well as the whole scene image, though our method can generalize to all formats of image input (*e.g.* multiple whole image frames, to provide temporal information and help action disambiguation). In practice, we filter out all bounding boxes of width or height < 10px since they do not provide useful information to the model, and resize both the entire image and bounding boxes to 112×112.

[1] https://huggingface.co/transformers/

	Method	Seen	New	Difference
	Chance	~0	~0	-
	BERT (scratch) [2]	34.3	17.7	16.6
	BERT (pretrained) [2]	39.8	20.7	19.1
	BERT with Vision [1]	56.1	37.6	18.5
EXPERT	Isolated attn	65.0	51.6	13.4
EXPERT	Target-to-ref attn	61.7	45.7	16.0
EXPERT	Via-vision attn	63.5	53.0	10.5
EXPERT	+ Input pointing	62.7	48.3	15.4
EXPERT	Full attn	59.2	44.4	14.8

Table 2: **Compositionality on EPIC-Kitchens:** We show top-5 accuracy at predicting masked compositions of seen nouns and verbs – both the verb and the noun must be correctly predicted.

Test	BERT with Vision	EXPERT
Acquiring Familiar Words		
Seen verbs	68.5	71.7
Seen nouns	44.6	59.3
Seen compositions	35.7	40.9
New compositions	30.0	34.4
Acquiring New Words		
Nouns 1:1	56.2	78.6
Verbs 1:1	60.5	72.1
Nouns 2:1	50.7	71.5
Verbs 2:1	44.1	64.5

Table 3: **Results on EPIC-Kitchens without Bounding Boxes:** We show accuracy on all evaluation metrics used in the paper, but withhold ground truth object bounding boxes at test time. EXPERT continues to outperform the competitive vision and language baseline.

4.1 Does EXPERT require bounding boxes?

We test EXPERT on EPIC-Kitchens without any additional image regions (i.e., only with the full image) to quantify the importance of providing grouped image regions. This model experiences a slight performance decrease in language acquisition, performing 4% worse with a 1:1 distractor ratio, and 2% worse with a 2:1 ratio. However, the baseline BERT with Vision model has larger decreases on all metrics when tested without additional image regions, so our model still outperforms it. When testing on masked language modeling (as in Sections 4.4 and 4.5 in the main paper), EXPERT performs 10% worse on verbs, 14% worse on nouns, 24% on seen compositions, and 29% on new compositions. However, the baseline model again has larger decreases in performance. These experiments show that while EXPERT improves when provided more visual information, it can acquire new language even when provided just with one image and perform stronger than baselines.

On Flickr30k, when testing with only the full image as model input in addition to text, performance decreases by *at most 7%*. Therefore, EXPERT does not require using image regions at test time. Performance decreases by a similar amount on vision and language baselines, such that EXPERT still outperforms them.

5 Flickr Visualizations

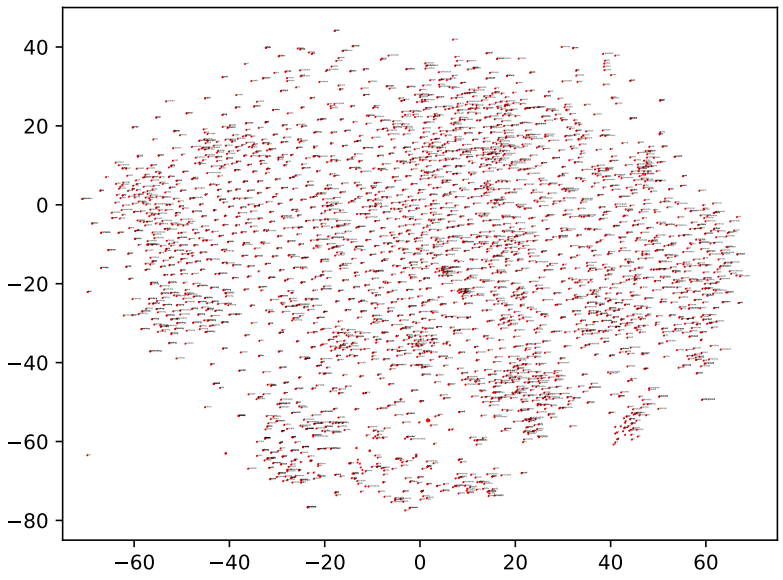

Fig. 2: **t-SNE 2D projection of Flickr embeddings:** We show a 2D projection computed using the t-SNE algorithm [4] of the word embedding matrix for EXPERT trained on the Flickr dataset. Each dot corresponds to the word shown to its top right. Please zoom in to view in detail.

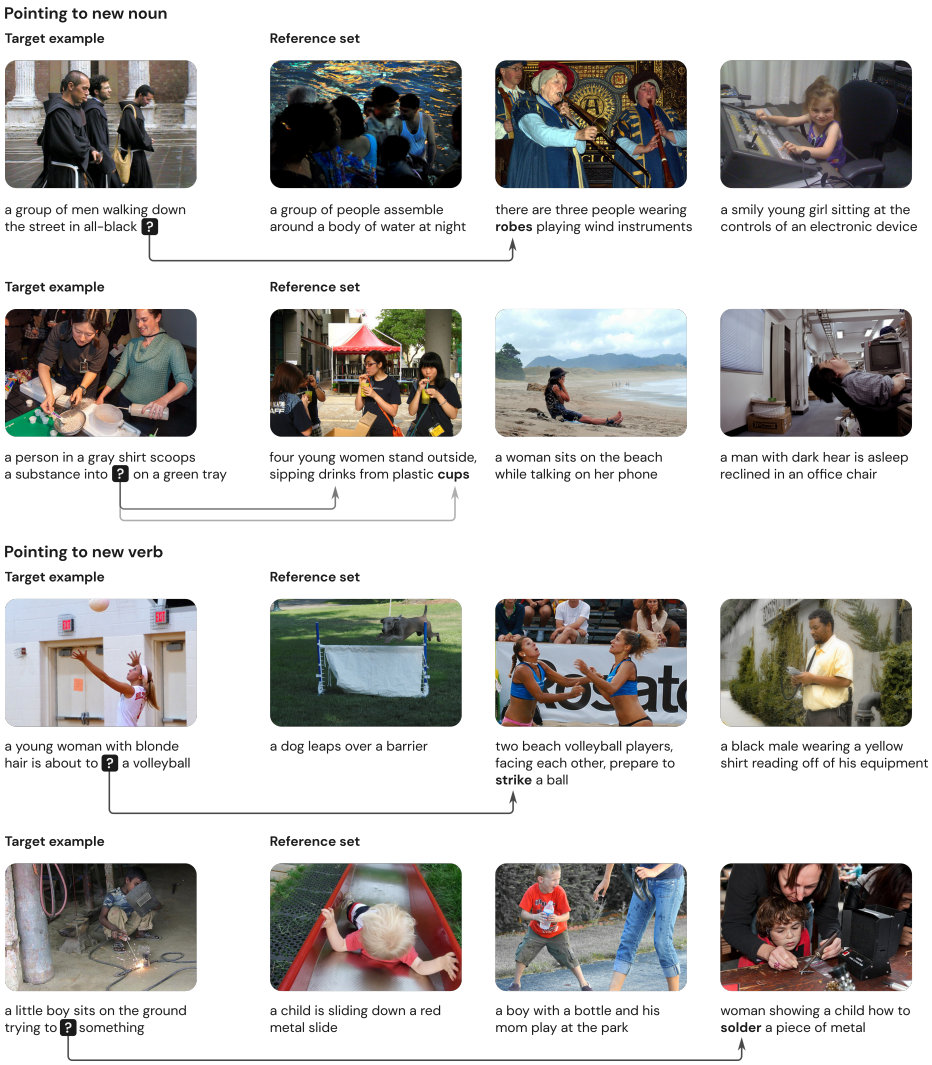

Fig. 3: **Acquiring New Words on Flickr30k:** We show examples where the model acquires new words. ? in the target example indicates the masked out new word. **Bold** words in the reference set are ground truth. The model makes predictions by pointing into the reference set, and the weight of each pointer is visualized by the shade of the arrows shown (weight $< 3\%$ is omitted).

References

1. Chen, Y.C., Li, L., Yu, L., Kholy, A.E., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: UNITER: Learning UNiversal Image-TExt Representations. Tech. rep. (2019)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Tech. rep., `https://github.com/tensorflow/tensor2tensor`
3. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
4. van der Maaten, L., Hinton, G.: Visualizing high-dimensional data using t-sne. Journal of Machine Learning Research **9**, 2579–2605 (2008)