A Case Study on Combining ASR and Visual Features for Generating Instructional Video Captions

Jack Hessel
Cornell University
jhessel@cs.cornell.edu

Bo Pang **Zhenhai Zhu** **Radu Soricut**
Google
{bopang,zhenhai,rsoricut}@google.com

Abstract

Instructional videos get high-traffic on video sharing platforms, and prior work suggests that providing time-stamped, subtask annotations (e.g., "heat the oil in the pan") improves user experiences. However, current automatic annotation methods based on visual features alone perform only slightly better than constant prediction. Taking cues from prior work, we show that we can improve performance significantly by considering automatic speech recognition (ASR) tokens as input. Furthermore, jointly modeling ASR tokens and visual features results in higher performance compared to training individually on either modality. We find that unstated background information is better explained by visual features, whereas fine-grained distinctions (e.g., "add oil" vs. "add olive oil") are disambiguated more easily via ASR tokens.

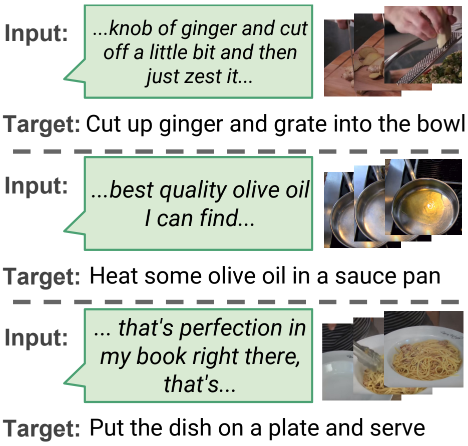

Input:
...knob of ginger and cut off a little bit and then just zest it...
Target: Cut up ginger and grate into the bowl
Input:
...best quality olive oil I can find...
Target: Heat some olive oil in a sauce pan
Input:
... that's perfection in my book right there, that's...
Target: Put the dish on a plate and serve

Figure 1: Illustration of a multimodal dense instructional video captioning task. Models are given access to both video frames and ASR tokens, and must generate a recipe instruction step for each video segment. The speaker in the video *sometimes* (but not always) references literal objects and actions.

1 Introduction

Instructional videos increasingly dominate user attention on online video platforms. For example, 86% of YouTube users report using the platform often to learn new things, and 70% of users report using videos to solve problems related to work, school, or hobbies (O'Neil-Hart, 2018).

Prior work in user experience has investigated the best way of presenting instructional videos to users. Kim et al. (2014), for example, compare two options; first: presenting users with the video alone, and second: presenting the video with an additional *structured* representation, including a timeline populated with task subgoals. Users interacting with the structured video representation reported higher satisfaction, and external judges rated the work they completed using the videos as having higher quality. Margulieux et al. (2012) and Weir et al. (2015) similarly find that presenting explicit subgoals alongside how-to videos improves user experiences. Thus, presenting instructional videos with additional structured annotations is likely to benefit users.

These studies rely on human annotation of time-stamped subtask goals, e.g., timed captions created through crowdsourcing. However, human-in-the-loop annotation is infeasible to deploy for popular video sharing platforms like YouTube that receive hundreds of hours of uploads per minute. In this work, we address the task of *automatically* producing captions for instructional videos at the level of video segments. Ideally, generated captions provide a literal, imperative description of the procedural step occurring for a given video segment, e.g., in the cooking context we consider, "add the oil to the pan."

Producing segment-level captions is a sub-task of dense video captioning, where prior work has mostly focused on visual-only models. Dense captioning is a difficult task, particularly in the

Proceedings of the 23rd Conference on Computational Natural Language Learning, pages 419–429
Hong Kong, China, November 3-4, 2019. ©2019 Association for Computational Linguistics

instructional video domain, as fine-grained distinctions may be difficult or impossible to make with visual features alone. Visual information can be ambiguous (e.g., distinguishing between "olive oil" vs. "vegetable oil") or incomplete (e.g., preparation steps may occur off-camera). In our study, a first important finding is that, for the dataset considered, current state-of-the-art, visual-features–only models only slightly outperform a constant prediction baseline, e.g., by 1.5 BLEU/METEOR points.

To improve performance in this difficult setting, we consider the *automatic speech recognition* (ASR) tokens generated by YouTube. These publicly available tokens are an ASR model's attempts to map words spoken in videos into text. However, while a promising potential source for signal, it is not always trivial to transform even accurate ASR into the desired imperative target: while there are cases of clear correspondence between the literal actions in the video and the ASR tokens, in other cases, the mapping is imperfect (Fig. 1). For example, when finishing a dish, a user says "that's perfection in my book right there" rather than "put the dish on a plate and serve." There are also cases where no ASR tokens are available at all. Despite these potential difficulties, previous work has demonstrated that ASR can be informative in a variety of instructional video understanding tasks (Naim et al., 2014, 2015; Malmaud et al., 2015; Sener et al., 2015; Alayrac et al., 2016; Huang et al., 2017); though less work has focused on instructional caption *generation,* which is known to be difficult and sensitive to input perturbations (Chen et al., 2018).

We find that incorporating ASR-token–based features significantly improves performance over visual-features–only models (e.g., CIDEr improves 0.53 ⇒ 1.0, BLEU-4 improves 4.3 ⇒ 8.5). We also show that *combining* ASR tokens and visual features results in the highest performing models, suggesting that the modalities contain complementary information.

We conclude by asking: what information is captured by the visual features that *is not* captured by the ASR tokens (and vice versa)? Auxiliary experiments examining performance of models in predicting the presence/absence of individual word types suggest that visual signals are superior for identifying unspoken, implicit aspects of scenes; for instance, in order to mix ingredients, they must be placed in a bowl — and although bowls are often visually present in the scene, "bowl" is often not explicitly mentioned by the speaker. Conversely, ASR features readily disambiguate between fine-grained entities, e.g., "olive oil" vs."vegetable oil", a task that is difficult (and sometimes impossible) for visual features alone.

2 Related Work

Narrated instructional videos. While several works have matched audio and video signals in an unconstrained setting (Arandjelovic and Zisserman, 2017; Tian et al., 2018), our work builds upon previous efforts to utilize accompanying speech signals to understand online *instructional* videos, specifically. Several works focus on learning video-instruction alignments, and match a fixed set of instructions to temporal video segments (Regneri et al., 2013; Naim et al., 2015; Malmaud et al., 2015; Hendricks et al., 2017; Kuehne et al., 2017). Another line of previous work uses speech to extract and align language fragments, e.g., verb-noun pairs, with instructional videos (Gupta and Mooney, 2010; Motwani and Mooney, 2012; Alayrac et al., 2016; Huang et al., 2017, 2018; Hahn et al., 2018). Sener et al. (2015), as part of their parsing pipeline, train a 3-gram language model on segmented ASR token inputs to produce recipe steps.

Dense Video Captioning. Recent work in computer vision addresses dense video captioning (Krishna et al., 2017; Li et al., 2018; Wang et al., 2018), a supervised task that involves (i) segmenting the input video, and, (ii) generating a natural language description for each segment. Here, we focus on the second subtask of generating descriptions given a ground-truth segmentation; this setting isolates the language generation part of the modeling process.[1] Most related to the present work are several dense captioning approaches that have been applied to instructional videos (Zhou et al., 2018b,c). Zhou et al. (2018c) achieve state-of-the-art performance on the dataset we consider; their model is video-only, and combines a region proposal network (Ren et al., 2015) and a Transformer (Vaswani et al., 2017) decoder.

Multimodal Video Captioning. Several works

[1] We find that state-of-the-art models perform poorly even for just this subtask (see § 3.2), so we reserve the full task for future work.

have employed multimodal signals to caption the MSR-VTT dataset (Xu et al., 2016), which consists of 2K video clips from 20 general categories (e.g., "news", "sports") with an average duration of 10 seconds per clip. In particular, Ramanishka et al. (2016); Xu et al. (2017); Hori et al. (2017); Shen et al. (2017); Chuang et al. (2017); Hao et al. (2018) all report small performance gains when incorporating audio features on top of visual features. However — we suspect that instructional video domain is significantly different than MSR-VTT (where the audio information does not necessarily correspond to human speech), as we find that ASR-only models significantly surpass the state-of-the-art video model in our case. Palaskar et al. (2019) and Shi et al. (2019), contemporaneous with the submission of the present work, also examine ASR as a source of signal for generating how-to video captions.

3 Dataset

We focus on YouCook2 (Zhou et al., 2018b), the largest human-captioned dataset of instructional videos publicly available.[2] It contains 2000 YouTube cooking videos, for a total of 176 hours, and spans 89 different recipes. Each video averages at 5.26 minutes, and is annotated with an average of 7.7 temporal segments (i.e., start/end points) corresponding to semantically distinct recipe steps. Each segment is associated with an imperative caption, e.g., "add the oil to the pan", for an average of 8.8 words per caption.

At the time of analysis (June 2018), over 25% of the YouCook2 videos had been removed from YouTube, and therefore we do not consider them. As a result, all our experiments operate on a *subset* of the YouCook2 data. While this makes direct comparison with previous and future work more difficult, our performance metrics can be viewed as lower bounds, as they are trained on less data compared to, e.g., (Zhou et al., 2018c). Unless noted otherwise, our analyses are conducted over 1.4K videos and the 10.6K annotated segments contained therein.

3.1 A Closer Look at ASR tokens

We collected the ASR tokens automatically generated by YouTube (available through the YouTube Data API[3] with trackKind = ASR), which are then mapped to their temporally corresponding video segments. We start by asking the following questions: How much narration do users provide for instructional videos? And: can YouTube's ASR system detect that speech?

Not surprisingly, speakers in videos tend to be more verbose than the annotated groundtruth captions: we find the length distribution of ASR tokens per segment to be roughly log-normal, with mean/median length being 42/28 tokens respectively (compared to a mean of 9 tokens/segment for captions). Over the 10.6K available segments, only 1.6% of them have zero associated tokens. Furthermore, based on automatic language identification provided by the YouTube API and some manual verification, we estimated that less than 1% of videos contain completely non-English speech (but we do not discard them from our experiments).

We also investigate the words-per-minute (WPM) ratio, based on the video segment length. The mean value of 134 WPM is slightly lower than, but comparable to, previously reported figures of English speaking rates (Yuan et al., 2006), which indicates that, for this set of video segments, words are being detected at rates comparable to everyday English speech.

3.2 A Closer Look at the Generation Task

To better understand the generation task, we computed lower and upper bounds for generation performance using a constant-prediction baseline and human performance, respectively.

Lower bound: constant. For all segments at test time, we predict "heat some oil in a pan and add salt and pepper to the pan and stir." This sentence is constructed by examining the most common n-grams in the corpus and pasting them together.

Upper bound: human estimate. We conducted a small-scale experiment to estimate human performance for the segment-level captioning task. Two of the authors of this paper, after being trained on segment-level captions from three videos, attempted to mirror that style of annotation for the segments of 20 randomly sampled videos, totalling over 140 segment annotations each.[4] Both human annotators report low-confidence with the

[2] How2 (Sanabria et al., 2018) tackles the different task of predicting video uploader-provided descriptions/captions, which are not always appropriate summarizations.

[3] https://developers.google.comyoutube/v3/docs/captions

[4] These preliminary experiments are not meant to provide a definitive, exact measure of inter-annotator agreement.

task, in particular, they found it difficult to maintain a consistent level of specificity in terms of how many factual details to include (e.g., "mix together" vs. "mix the peppers and mushrooms together.")

Results: We compute corpus-level performance statistics using four standard generation evaluation metrics: ROUGE-L (Lin, 2004), CIDEr (Vedantam et al., 2015), BLEU-4 (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) (higher is better in all cases).

Note that our evaluation is micro-averaged at the segment level, and differs slightly from prior work on this dataset, which has mostly reported metrics macro-averaged at the video level. We switched the evaluation because some metrics like BLEU-4 exhibit undesirable sparsity artifacts when macro-averaging, e.g., any video without a correct 4-gram gets a zero BLEU score, even if there are many 1/2/3-grams correct. Segment-level averaging, the standard evaluation practice in fields like machine translation, is insensitive to this sparsity concern, and (we believe) provides a more robust perspective on performance.

	BLEU-4	METEOR	ROUGE-L	CIDEr
Constant Prediction	2.70	10.3	21.7	.15
Zhou et al. (2018c)	3.84	11.6	27.4	.38
Sun et al. (2019b)	4.07	11.0	27.5	.50
Sun et al. (2019a)	4.31	11.9	29.5	.53
Human Estimate	15.2	25.9	45.1	3.8

Table 1: The performance of several state-of-the-art, video-only models, with lower (constant prediction) and upper (human estimate) bounds.

This comparison highlights the gap that remains between the simplest possible baseline, several computer vision based models, and (roughly) how well humans perform at this task. Given that Sun et al. (2019a) is a highly tuned computer vision model transfer learned from a corpus of over 300K cooking videos, from the perspective of building video captioning systems in practice, we suspect that incorporating additional modalities like ASR is more likely to result in performance gains versus building better computer vision models.

4 Models

In addition to the constant prediction baseline, we explore a series of ASR-based baseline methods:

ASR as the Caption (ASC) This baseline returns the test-time ASR token sequence as the caption. While the result is not a coherent, imperative step, performance of this method offers insight into the extent of word overlap between the ASR sequence and the target groundtruth, as measured by the captioning metrics.

Filtered ASR (FASC) Given that the ASR token sequences are much longer than groundtruth captions (§ 3.1), the performance of ASC incurs a length (or precision-based) penalty for several metrics. The FASC baseline strengthens ASC by removing word types that are less likely to appear in groundtruth captions, e.g., "ah", "he", "hello," or "wish". Specifically, we only keep words with high $\frac{P(w \mid GT)}{P(w \mid ASR)}$ values, i.e., words that would be indicative of the groundtruth class if we were to build a Naive-Bayes classifier with add-one smoothing; probabilities are computed only over the training set to reduce the risk of overfitting. This baseline produces outputs that are shorter compared to ASC, but it is unlikely to yield fluent, readable text.

ASR-based Retrieval (RET) This retrieval baseline memorizes the recipe steps in the training set, and represents them each as tf-idf vectors. At test-time, the ASR sequence is converted into a tf-idf vector and compared to each training-set caption via cosine similarity.[5] The training caption that is most similar to the test-time ASR according to this metric is returned as the "generated" caption. Note that, although a memorization-based technique, this baseline method produces de-facto captions as outputs.

4.1 Transformer-based Neural Models

We explore neural encoder-decoder models based on Transformer Networks (Vaswani et al., 2017). In contrast to RNNs, Transformers abandon recurrence in favor of a mix of different types of feed-forward layers, e.g., in the case of the Transformer decoder, self-attention layers, cross-attention layers (attending to the encoder outputs), and fully connected feed-forward layers. We explore two variants of the Transformer, corresponding to different hypotheses about what information might be useful for captioning instructional videos.

ASR Transformer (AT) This model learns to map ASR-token sequences directly to captions using

[5] We tried several variants of this method, e.g., comparing test ASR to train ASR, but found that comparing test ASR to train captions performed the best.

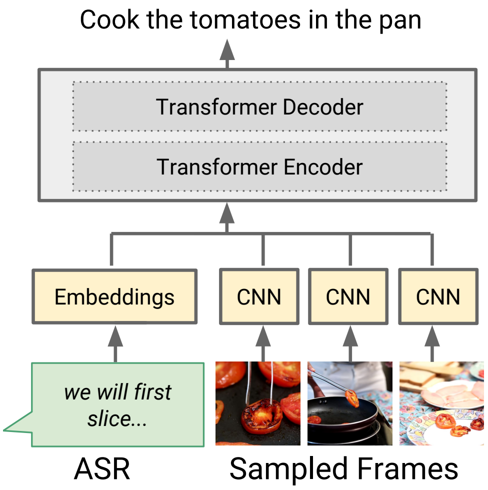


Figure 2: The AT+Video model. Both the encoder and decoder layers perform cross-modal attention.

	BLEU-4	METEOR	ROUGE-L	CIDEr
CNST	2.70	10.03	21.69	0.15
Sun et al. (2019a)	4.31	11.91	29.47	0.53
ASC	1.68	14.86	19.24	0.20
FASC	4.32	18.47	30.07	0.59
RET	5.68	14.29	28.06	0.80
AT	8.55	16.93	35.54	1.06
AT+Video	9.01	17.77	**36.65**	**1.12**

Table 2: Caption generation performance: AT+Video is a multimodal model that adds visual frame features to AT. A bolded value in a column indicates a statistically-significant improvement, whereas an underline indicates a statistical tie for best ($p < .01$).

a standard sequence-to-sequence Transformer architecture. The model's parameters are optimized to maximize the probability of the ground-truth instructions, conditioned on the input ASR sequences.

Multimodal model (AT+Video) We incorporate video features into the ASR transformer (Fig 2). For ease of comparison with prior and future work, we use features extracted from ResNet34 (He et al., 2016) pretrained on the ImageNet classification task; these features are provided in the YouCook2 data release. Each video is initially uniformly sampled at 512 frames, with an average of 30 frames per captioned-segment.

To represent each video segment, first, k frames are randomly sampled with replacement. The sampled frames are temporally sorted to preserve ordering information, and their corresponding ResNet34 feature vectors are projected to the Transformer encoder hidden dimension via a width-1 1D convolution. We use $k = 10$ for all our experiments. The encoder self-attention layers perform *cross-modal attention* operations between the visual features and the ASR-token–based features. For each output token, the decoder attends to previously predicted tokens, and encoder outputs for all input frames / ASR tokens.

5 Experiments

We perform 10-fold cross-validation with randomly sampled 80/10/10 train/dev/test splits (split at the video-level), using the same splits for all models. After discarding the videos that were deleted at the time of data collection, each split contains roughly 1.1K training videos (averaging 8.3K training segments). We report mean performance over these splits according to four standard captioning accuracy metrics, introduced in §3.2. ROUGE-L, CIDEr, BLEU-4, and METEOR. We perform both Wilcoxon signed-rank tests (Demšar, 2006) and two-sided corrected resampled t-tests (Nadeau and Bengio, 2000) to estimate statistical significance. To be conservative and reduce the chance of Type I error, we take whichever p-value is larger between these two tests.

Transformer-based model details. For each cross-validation split, we use a batch size of 128, tie the Transformer model's feed forward and model dimensions $d_{ffn} = d_{model}$, and optimize regularized cross-entropy loss using Adam (Kingma and Ba, 2015) with $lr = .001$. We train models for 100K steps, storing checkpoint files periodically. For each split, we train 8 model variants, conducting a grid search over model dimension, number of encoder/decoder layers, and L2 regularization: we consider all model parameter settings in $(d_{model}, N_{layer}, \lambda_{reg}) \in \{128, 256\} \times \{2, 3\} \times \{.0005, .001\}$ for each cross-validation split independently, and select the highest performing, checkpointed model according to ROUGE-L over the development set for that fold. Transformer models are implemented using `tensor2tensor` (Vaswani et al., 2018) and `Tensorflow` (Abadi et al., 2015). The vocabulary (average size 800) is determined separately using the training data for each cross-validation split. Words are considered if they occur at least 5 times in the ground-truth of the current training set.[6] This leads to an OOV rate of ~60% in the input. We truncate inputs at 80 tokens (~10-15%

[6] Different vocabulary creation schemes, e.g., sub-word tokenization, led to small performance decreases.

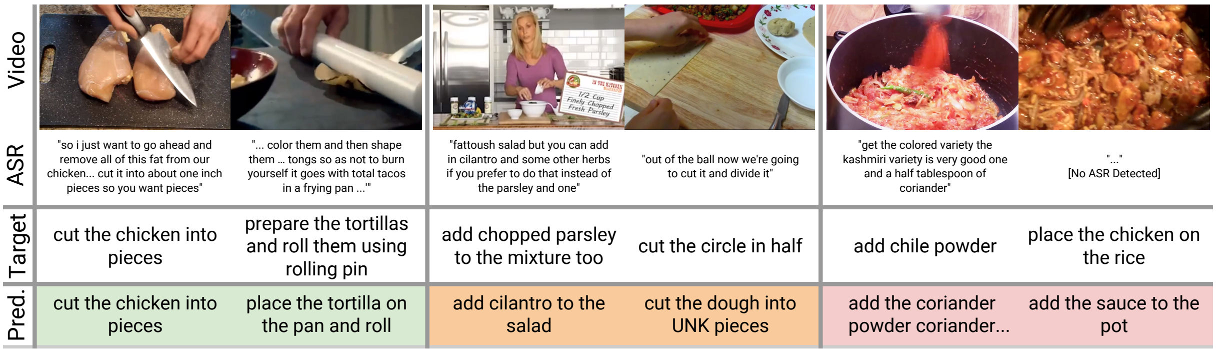

Video
ASR
"so i just want to go ahead and remove all of this fat from our chicken... cut it into about one inch pieces so you want pieces"
"... color them and then shape them ... tongs so as not to burn yourself it goes with total tacos in a frying pan ..."
"fattoush salad but you can add in cilantro and some other herbs if you prefer to do that instead of the parsley and one"
"out of the ball now we're going to cut it and divide it"
"get the colored variety the kashmiri variety is very good one and a half tablespoon of coriander"
"..." [No ASR Detected]
Target
cut the chicken into pieces
prepare the tortillas and roll them using rolling pin
add chopped parsley to the mixture too
cut the circle in half
add chile powder
place the chicken on the rice
Pred.
cut the chicken into pieces
place the tortilla on the pan and roll
add cilantro to the salad
cut the dough into UNK pieces
add the coriander powder coriander...
add the sauce to the pot

Figure 3: Example generations from AT+Video in cases where it performs **well**, **okay**, and **poorly**.

of transcripts are truncated in this process). For simplicity, decoding is done greedily in all cases.

Generation Experiment Results. Table 2 reports the performance of each model. For unimodal models, simple baselines like FASC (filtered ASR) and RET (training-caption retrieval) outperform the state-of-the-art video-only model of Sun et al. (2019a), according to the four automatic evaluation metrics. Overall, AT yields the best unimodal performance. Combining ASR and visual signals into a multimodal representation performs even better: the AT+Video model tends to outperform AT (and Sun et al. (2019a)), according to ROUGE-L, CIDEr, and METEOR (p <.01). Since AT and AT+Video have identical architectures and differ only in the available inputs, this result provides strong evidence that it is indeed the *multimodality* of AT+Video that leads to the (statistically significant) performance gains over the strongest unimodal models. We present some output examples in Fig. 3.

5.1 Diversity of Generated Captions

In addition to the automatic quality metrics, we measure how diverse the generated caption are for each model, using the following metrics: vocabulary coverage (the percent of vocabulary that was predicted at test-time by each algorithm at least once); proportion not copied (the percent of generated captions that do not appear in the training set verbatim); and output uniqueness (the percent of generated captions that are unique). These metrics are useful because they can highlight undesirable, degenerate behavior for models.[7] As an upper-bound, we compute these metrics for the ground-truth (GT) test-time targets. Note that even the ground-truth targets do not achieve 100% in these diversity metrics: for vocabulary coverage, not all vocabulary items appear in the ground-truth captions for a given cross-validation split; similarly, for proportion not copied/output uniqueness, because there are repeated captions in the label set.

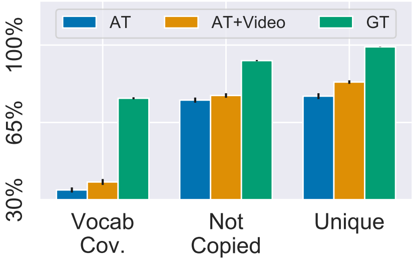


Figure 4: The multimodal model AT+Video produces slightly more diverse captions than its unimodal counterparts.

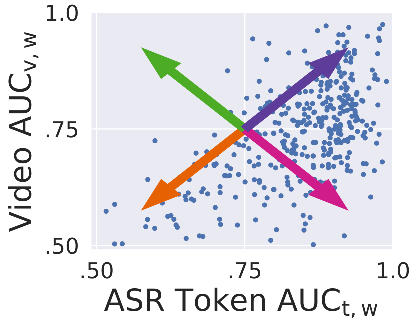
According to all metrics, AT+Video outputs are slightly more diverse compared to the AT outputs (Fig. 4). This observation suggests that the multimodal model is not simply exploiting a degeneracy to achieve its performance improvements.

6 Complementarity of Video and ASR

We now turn to the question of *why* multimodal models produce better captions: what type of signal does video contain that speech does not (and vice versa)? Our initial idea was to quantitatively compare the captions generated by AT versus AT+Video; however, because the dataset is relatively small, we were unable to make observations about the generated captions that were statistically significant.[8]

[7] For instance, the constant prediction baseline we consider would score low in both vocab coverage and uniqueness.

[8] In general, making concrete statements about the causal link between inputs and outputs of sequence-to-sequence models is challenging, even in the text-to-text case, see Alvarez-Melis and Jaakkola (2017).



knead	97.8
nori	97.1
yeast	96.1
mozzarella	95.8
lettuce	95.3
pancake	94.7
wrapper	94.3
patty	93.4
dal	93.0
grill	92.9
pizza	92.9
oven	92.7
bake	92.3

(a) **Easiest** ($\text{AUC}_{\mu,w}$)

4	43.8
bit	51.7
about	52.1
prepare	52.3
mixed	54.5
then	54.5
spoon	54.9
or	55.9
it	56.2
ready	56.7
3	57.1
few	58.3
more	58.5

(b) **Hardest** ($\text{AUC}_{\mu,w}$)

fat	39.7
turn	36.4
sea	35.9
white	31.7
chilies	30.8
dried	30.6
beer	30.3
pancetta	30.0
mustard	29.8
spice	28.4
sliced	28.3
cinnamon	28.0
warm	27.8

(c) **ASR Better** ($\text{AUC}_{\Delta,w}$)

sandwich	-18.0
stove	-15.4
tuna	-14.3
again	-12.6
mince	-11.2
wok	-8.9
burger	-8.8
pizza	-8.4
serve	-7.8
4	-6.6
mussels	-6.6
tray	-6.3
bowl	-5.9

(d) **Video Better** ($\text{AUC}_{\Delta,w}$)

Figure 5: Per-word classification results using ASR and/or Video features. Each point in the scatterplot represents a different word-type; x-coordinate values show how well a word is predicted by ASR-token features; y-coordinate values show how well a word is predicted by video features. Tables (a)-(d) show word types that are easy, universally difficult, better-predicted-by-ASR, and better-predicted-by-video, respectively.

Instead, we examine properties of the ASR-token–based and visual features directly. Following a procedure inspired from (Lu et al., 2008; Berg et al., 2012; Dai et al., 2018; Mahajan et al., 2018), we consider the auxiliary task of predicting presence/absence of unigrams in the ground truth captions from features extracted from corresponding segments. We train two unimodal classifiers, one using ASR-token–based features and one using visual features, and measure their relative capacity to predict different word types; the goal is to measure which word types are most-predictable from the ASR tokens and, conversely, which ones are most-predictable from the visual features.

For each segment, we predict the unigram distribution of its corresponding caption using a unimodal softmax classifier: for simplicity, we use a 2-layer, residual deep averaging network (Iyyer et al., 2015) for both the visual and ASR-based classifier. We measure per-word-type performance using AUC, which is word-frequency independent.

Specifically — for each word type w (e.g., w = beer) we measure how well w is predicted by the classifier based on ASR / spoken tokens $\text{AUC}_{t,w}$ (e.g., $\text{AUC}_{t,beer} = 98$) and, conversely, how well w is predicted by the visual classifier $\text{AUC}_{v,w}$ ($\text{AUC}_{v,beer} = 68$). For a given word type, we measure its overall difficulty by averaging $\text{AUC}_{t,w}$ and $\text{AUC}_{v,w}$; we call this $\text{AUC}_{\mu,w}$ ($\text{AUC}_{\mu,beer} = 83$). Similarly, we measure the difference in difficulty by subtracting $\text{AUC}_{t,w}$ and $\text{AUC}_{v,w}$ to give $\text{AUC}_{\Delta,w}$ ($\text{AUC}_{\Delta,beer} = 30$) with higher values indicating that a word type is predicted better by the spoken-token features compared to the visual features. We plot $\text{AUC}_{t,w}$ versus $\text{AUC}_{v,w}$ for 382 words in Fig. 5 (results are averaged over 10 cross-val splits).

Absolute Performance. Points in the upper-right quadrant of Fig. 5 represent words that are easy for both visual and ASR-token–based features to predict, whereas points in the lower-left represent words that are more difficult. Specific ingredients, e.g., "nori" and "mozzarella," are often easy to detect, as are actions closely associated with particular objects (e.g., "dough" is almost always the object being "knead"-ed). Conversely, pronouns (e.g., "it") and conjunctions (e.g., "or") are universally difficult to predict.

Visual vs. ASR-token–based features. In general, ASR-token–based features carry greater predictive power, as evidenced by the skew towards the bottom right in the scatterplot in Fig. 5. One pattern in the cases where speech features perform better (Fig. 5c) is that words are often modifiers, e.g., *white* (pepper), *sea* (salt), *dried* (chilies), *olive* (oil), etc. Indeed, small, detailed distinctions may be often difficult to make from visual features, e.g., "vegetable oil" and "olive oil" may look identical in most YouTube videos.

Nonetheless, there are types better predicted by video features (Fig. 5d). Often, these are cases that require unstated, background knowledge, i.e., references to objects not explicitly stated by the speaker(s). To quantify this observation, for each word type we compute the likelihood that it is *stated* by the speaker in the video, given that it appears in the ground-truth caption, i.e., $P(w \in \texttt{ASR} \mid w \in \texttt{GT})$. Aside from trivial cases (e.g., words misspelled in the GT never appear in the ASR), words that are often unstated include action words (e.g., "place", "crush") and cookware (e.g., "pan", "wok", "pot"). Words that are often stated include specific ingredients (e.g., "honey", "coconut", "ginger"). In contrast to word frequency (which is uncorrelated with $\text{AUC}_{\Delta,w}$, Spearman

$\rho \approx 0$), stated rate *is* correlated with $\text{AUC}_{\Delta,w}$ ($\rho = 0.44, p < .01$).

7 Oracle Object Detection

The results in Table 2 indicate that, while adding visual information yields statistically significant improvements to the ASR-only model, the improvements are not large in magnitude. This leaves open the question of whether (a) any visual information simply does not provide much additional information on top of ASR, or (b) we need better visual modeling. We take a first step in addressing this question by experimenting with an "oracle" object detector that provides perfect-precision predictions.[9] If even oracle object detection does not help, then the answer is more likely (a) rather than (b) above.

As part of a YouCook2 data release, bounding box annotations for selected objects in the recipe text (Zhou et al., 2018a) were provided. Unfortunately, while these could have served as an oracle, the actual annotations are only available for a small fraction of the data. Instead, we consider the set of 62 object labels made available. We simulate a high-precision, oracle object detector by identifying – per video segment – the overlap between (morphology-normalized) groundtruth caption mentions and the 62 object labels available.[10] For instance, for the groundtruth caption "put the mushrooms in the pan", the oracle object detector yields "mushroom" and "pan". 89% of segments receive at least one oracle object. The oracle object detections are then fed into the Transformer encoder (in random order), either by themselves (Oracle) or along with the ASR token sequence (AT+Oracle). We perform the same cross-validation experiments as described in §5, and report the average ROUGE-L (we observe similar trends with other metrics):

	AT	AT+Video	Oracle	AT+Oracle
ROUGE-L	35.5	36.7	40.8	**45.5**

Because the AT+Oracle model achieves large improvements over AT+Video, we suspect that building higher-quality visual representations is a promising avenue for future work.

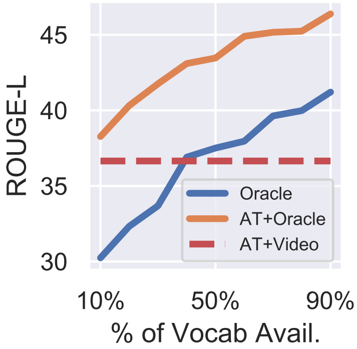


Figure 6: The performance of the oracle methods increases as they are given access to an increasing number of object types.

How weak of an oracle can still produce high performance? Fig. 6 shows performances of models using *subsets* of the 62 objects (most frequent 10% of objects through 90%) over one cross-validation fold. AT+Oracle gives better performance than AT+Video by detecting *just 6 object types,* and the oracle by-itself (which is only given access to object sets) achieves comparable performance to AT+Video with 30 object types. These results suggest that, at least for this task, the Transformer decoder is likely not the main performance bottleneck, as it is able to paste-together unordered object detections into captions effectively.

8 Conclusion

In this work, we demonstrate the impact of incorporating both visual and ASR-token–based features into instructional video captioning models. Additional experiments investigate the complementarity of the visual and speech signals.

Our oracle experiments suggest that performance bottlenecks likely derive from the input encoding, as the decoder is able to paste-together even simple sets of object detections into high-quality captions. Future work would thus be well-suited to investigate better models of input data. Given the small size of the dataset, transfer learning may prove fruitful, e.g., pre-training the encoder with an unsupervised, auxiliary task; work contemporaneous with our submission from the computer vision community suggests that transfer learning indeed is a promising direction (Sun et al., 2019b,a; Miech et al., 2019).

[9] High-precision object detectors are gaining popularity in the computer vision community because the training data is easier to annotate, e.g., Krasin et al. (2017).

[10] This oracle is unlikely to be achievable, as it assumes 100% precision for the 62 objects considered (which also implies modeling *which* objects to talk about, a non-trivial task in itself (Berg et al., 2012)).

Acknowledgements. We would like to thank Maria Antoniak, Nan Ding, Sebastian Goodman, Jean Griffin, Fernando Pereira, Chen Sun, and the anonymous reviewers for their helpful comments.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

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

David Alvarez-Melis and Tommi S Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *EMNLP*.

Relja Arandjelovic and Andrew Zisserman. 2017. Look, listen and learn. In *ICCV*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop on Evaluation Measures for MT and Summarization*.

Alexander C Berg, Tamara L Berg, Hal Daumé III, Jesse Dodge, Amit Goyal, Xufeng Han, Alyssa Mensch, Margaret Mitchell, Aneesh Sood, Karl Stratos, et al. 2012. Understanding and predicting importance in images. In *CVPR*.

Hongge Chen, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, and Cho-Jui Hsieh. 2018. Attacking visual language grounding with adversarial examples: A case study on neural image captioning. In *ACL*.

Shun-Po Chuang, Chia-Hung Wan, Pang-Chi Huang, Chi-Yu Yang, and Hung-Yi Lee. 2017. Seeing and hearing too: Audio representation for video captioning. In *IEEE Automatic Speech Recognition and Understanding Workshop*.

Bo Dai, Sanja Fidler, and Dahua Lin. 2018. A neural compositional paradigm for image captioning. In *NeurIPS*.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *JMLR*.

Sonal Gupta and Raymond J Mooney. 2010. Using closed captions as supervision for video activity recognition. In *AAAI*.

Meera Hahn, Nataniel Ruiz, Jean-Baptiste Alayrac, Ivan Laptev, and James M Rehg. 2018. Learning to localize and align fine-grained actions to sparse instructions. *arXiv preprint arXiv:1809.08381*.

Wangli Hao, Zhaoxiang Zhang, and He Guan. 2018. Integrating both visual and audio cues for enhanced video caption. In *AAAI*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *ICCV*.

Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R Hershey, Tim K Marks, and Kazuhiko Sumi. 2017. Attention-based multimodal fusion for video description. In *ICCV*.

De-An Huang, Shyamal Buch, Lucio Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. 2018. Finding it: Weakly-supervised reference-aware visual grounding in instructional videos. In *CVPR*.

De-An Huang, Joseph J Lim, Li Fei-Fei, and Juan Carlos Niebles. 2017. Unsupervised visual-linguistic reference resolution in instructional videos. In *CVPR*.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *ACL*.

Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J Guo, Robert C Miller, and Krzysztof Z Gajos. 2014. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *CHI*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. 2017. Openimages: A public dataset for large-scale multi-label and multi-class image classification.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *ICCV*.

Hilde Kuehne, Alexander Richard, and Juergen Gall. 2017. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*.

Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. 2018. Jointly localizing and describing events for dense video captioning. In *CVPR*.

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

Yijuan Lu, Lei Zhang, Qi Tian, and Wei-Ying Ma. 2008. What are the high-level concepts with small semantic gaps? In *CVPR*.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. 2018. Exploring the limits of supervised pretraining. In *ECCV*.

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

Lauren E Margulieux, Mark Guzdial, and Richard Catrambone. 2012. Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. In *Conference on International Computing Education Research*.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*.

Tanvi S Motwani and Raymond J Mooney. 2012. Improving video activity recognition using object recognition and text mining. In *ECAI*.

Claude Nadeau and Yoshua Bengio. 2000. Inference for the generalization error. In *NeurIPS*.

Iftekhar Naim, Young C Song, Qiguang Liu, Liang Huang, Henry Kautz, Jiebo Luo, and Daniel Gildea. 2015. Discriminative unsupervised alignment of natural language instructions with corresponding video segments. In *NAACL*.

Iftekhar Naim, Young Chol Song, Qiguang Liu, Henry A Kautz, Jiebo Luo, and Daniel Gildea. 2014. Unsupervised alignment of natural language instructions with video segments. In *AAAI*.

Celie O'Neil-Hart. 2018. Why you should lean into how-to content in 2018. www.thinkwithgoogle.com/advertising-channels/video/self-directed-learning-youtube/. Accessed: 2019-09-03.

Shruti Palaskar, Jindrich Libovickỳ, Spandana Gella, and Florian Metze. 2019. Multimodal abstractive summarization for how2 videos. In *ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Vasili Ramanishka, Abir Das, Dong Huk Park, Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, and Kate Saenko. 2016. Multimodal video description. In *ACM MM*.

Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. 2013. Grounding action descriptions in videos. *TACL*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. 2018. How2: a large-scale dataset for multimodal language understanding. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*. NeurIPS.

Ozan Sener, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. 2015. Unsupervised semantic parsing of video collections. In *ICCV*.

Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li, Yurong Chen, Yu-Gang Jiang, and Xiangyang Xue. 2017. Weakly supervised dense video captioning. In *CVPR*.

Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and Ming Zhou. 2019. Dense procedure captioning in narrated instructional videos. In *ACL*.

Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. 2019a. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019b. Videobert: A joint model for video and language representation learning. In *ICCV*.

Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. 2018. Audio-visual event localization in unconstrained videos. In *ECCV*.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

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

Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. 2018. Bidirectional attentive fusion with context gating for dense video captioning. In *CVPR*.

Sarah Weir, Juho Kim, Krzysztof Z Gajos, and Robert C Miller. 2015. Learnersourcing subgoal labels for how-to videos. In *CSCW*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*.

Jun Xu, Ting Yao, Yongdong Zhang, and Tao Mei. 2017. Learning multimodal attention lstm networks for video captioning. In *ACM MM*.

Jiahong Yuan, Mark Liberman, and Christopher Cieri. 2006. Towards an integrated understanding of speaking rate in conversation. In *International Conference on Spoken Language Processing*.

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

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018b. Towards automatic learning of procedures from web instructional videos. In *AAAI*.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018c. End-to-end dense video captioning with masked transformer. In *CVPR*.