Learning Action Changes by Measuring Verb-Adverb Textual Relationships

Davide Moltisanti, Frank Keller, Hakan Bilen, Laura Sevilla-Lara
The University of Edinburgh, United Kingdom
{davide.moltisanti, frank.keller, h.bilen, l.sevilla}@ed.ac.uk

Abstract

The goal of this work is to understand the way actions are performed in videos. That is, given a video, we aim to predict an adverb indicating a modification applied to the action (e.g. cut "finely"). We cast this problem as a regression task. We measure textual relationships between verbs and adverbs to generate a regression target representing the action change we aim to learn. We test our approach on a range of datasets and achieve state-of-the-art results on both adverb prediction and antonym classification. Furthermore, we outperform previous work when we lift two commonly assumed conditions: the availability of action labels during testing and the pairing of adverbs as antonyms.

Existing datasets for adverb recognition are either noisy, which makes learning difficult, or contain actions whose appearance is not influenced by adverbs, which makes evaluation less reliable. To address this, we collect a new high quality dataset: Adverbs in Recipes (AIR). We focus on instructional recipes videos, curating a set of actions that exhibit meaningful visual changes when performed differently. Videos in AIR are more tightly trimmed and were manually reviewed by multiple annotators to ensure high labelling quality. Results show that models learn better from AIR given its cleaner videos. At the same time, adverb prediction on AIR is challenging, demonstrating that there is considerable room for improvement.

1. Introduction

Learning how an action is performed in a video is an important step towards expanding our video understanding beyond action recognition. This task has been referred to as "adverb recognition", and has potential useful applications in robotics and retrieval. Consider a scenario where a robot handles fragile objects. In such case we would like to tell the robot to grasp objects *gently* to prevent it from breaking things. Similarly, learning how actions are performed enables more sophisticated queries for retrieval. Imagine learning a new recipe where a crucial step is stirring a mixture *vigorously*. It would be useful to find good examples showing how to whip a mixture in such a manner.

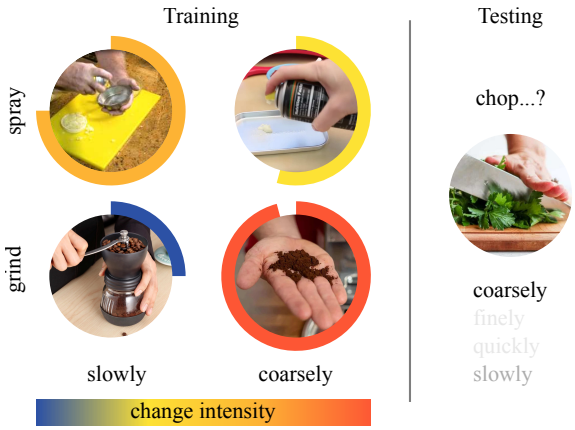

Figure 1. We aim to predict the way an action is performed in a video (right). Actions and their outcomes change in different ways when modified by adverbs, e.g. grinding coffee *slowly* does not have the same effect as grinding it *coarsely* (left). We learn to predict adverbs by recognising action changes in the video. We define action changes measuring verb-adverb textual relationships.

Adverb recognition is a challenging problem. Firstly, a single adverb can modify multiple actions in different ways. For example, to *grind* coffee *coarsely* (see Figure 1) we need to set a certain granularity in a grinder. Conversely, to *spray* something *coarsely* we would just use a spraying can more sparingly. Secondly, compared to actions or objects, adverbs are much harder to identify. Though somewhat subjective [1, 16], the temporal boundaries of an action can be easily determined. The spatial extent of an object is also easy to recognise for a human. Adverbs are instead more abstract: we can see their effect, but we struggle to pinpoint the spatio-temporal location where the adverb is appearing. Consider for example the act of spraying something *slowly* in Figure 1. We could say that *slowly* "is located" on the hand motions. At the same time, spraying something slowly could implicate a more even coating of a surface. That is, the outcome of an action and the end-state of the involved objects can also change according to the adverb. This makes it hard to label adverbs with visual annotations (e.g. bounding boxes), which makes the problem more challenging.

In fact, previous approaches [6, 7] learn adverbs as action changes in a weakly supervised manner. The state-of-the-art approach [6] treats adverbs as learnable parameters that modify actions. Specifically, the action change is learnt during training by contrasting antonyms, i.e. opposite adverbs. We show that learning adverbs in such manner can be difficult and can limit the ability of the model to generalise. We thus propose to define action changes by measuring distances in a text embedding space, and aim to learn such change from the video through regression. We show that our approach achieves new state-of-the-art results through extensive experiments. We also lift two major assumptions made in previous work [6, 7]: the availability of action labels during testing and the pairing of opposite adverbs as antonyms. Our method achieves stronger performance especially when the above assumptions are relaxed.

Besides being challenging, adverb recognition is also an under-explored domain and only few datasets are available. Doughty and Snoek [7] addressed the problem of scarce annotations, proposing a semi-supervised method that assigns pseudo-labels to boost recognition performance. Three datasets sourced from captioning benchmarks were also collected in [7]. These datasets offer a large number of clips, actions and adverbs. However, adverbs in these datasets appear to be descriptive rather than action modifiers, i.e. actions do not display a significant change when modified by the adverb. This is an issue when adverbs are modelled as action changes as in this work. We thus introduce a new dataset focusing on instructional videos where actions change considerably depending on the way they are carried out. We focus on cooking videos since in this domain action changes are prominent. Our **Adverbs in Recipes** (AIR) dataset was manually labelled and consists of over 7K videos, 10 adverbs and 48 actions.

To summarise our contributions: i) we propose a more effective approach to learn adverbs in videos. Our method achieves state-of-the-art results on multiple datasets and with fewer assumptions; ii) we introduce the AIR dataset for adverb recognition. Our focus on a domain where action changes are prominent and our careful manual annotation makes AIR more suitable for training and evaluating models for adverb understanding. We publicly release AIR and our code at github.com/dmoltisanti/air-cvpr23.

2. Related Work

Adverb Recognition Doughty *et al.* [6] introduced the adverb recognition task and proposed Action Modifiers. Adverbs are treated as learnable parameters that guide the optimisation of the video embeddings via a triplet loss. We show that such approach can be difficult to optimise and propose an alternative learning method. Rather than modelling action changes as trainable parameters we define them measuring distances in a text embedding space. We then learn action modifications via regression. We provide a more technical comparison in Section 3. A follow-up work [7] proposes a pseudo-labelling method where videos are assigned new labels based on the model prediction. Pseudo labels are obtained in an adaptive away, i.e. the threshold determining whether a label should be assigned is adapted for each adverb independently. In [7] the underlying model is also Action Modifiers [6]. Pang *et al.* [18, 19] introduced the "Behaviour Adverb" task. Here actions are not goal-oriented (e.g. "kiss, smoke") and adverbs correspond to moods (e.g. "sadly, solemnly"). Models use additional modalities such as human pose [18] or explicitly learn human expressions [8, 19]. While related, this task is different from the scope of this work, where we aim to learn action changes in goal-oriented (e.g. cooking) videos.

Adverb Datasets HowTo100M Adverbs [6] collects data from the instructional dataset HowTo100M [14] finding adverbs in the narration captions. Videos are loosely trimmed around the timestamp associated with the adverb. Due to the video-text misalignment in HowTo100M, and because training videos were not manually reviewed, videos in HowTo100M Adverbs are noisy. Authors estimated that only 44% of the training videos actually show the action. In total 6 adverbs were annotated. Three other datasets were introduced in [7]: ActivityNet/MSR-VTT/VATEX Adverbs. These are subsets of the namesake captioning datasets [11, 23, 26]. Videos were also obtained finding adverbs in captions, but annotations here are clean because captions were provided by annotators. We argue that the captioning nature of the original datasets is a reason of concern for learning action modifications. Indeed, adverbs tend to be a complementary description of the video, i.e. actions do not appear particularly influenced by the adverb. Peering into these datasets (see Figure 4), we found for example "sit inside/outside, talk outdoors/indoors, walk in/out, move down/up", etc. These adverbs modify the appearance of the video, however the action themselves are not modified. Arguably, the act of sitting is the same whether it takes place inside or outside, as is the act of walking in or out. Lastly, [18, 19] annotated adverbs for non-task-oriented actions, e.g. "run, kiss", however adverbs here express moods and manners, e.g. "politely, reluctantly".

The limited availability of datasets specifically designed for adverb understanding motivates us to collect a new dataset better suited for this task. We focus on recipe instructional videos where actions change significantly according to the adverb. We collect Adverbs in Recipes, which contains 7K videos, 10 adverbs and 48 actions. We employ annotators to verify that actions are carried out as indicated by the adverb. We present our dataset in Section 4.

Video-text Retrieval Understanding adverbs in videos is related to video-text retrieval [2, 5, 9, 11–13, 15, 17, 22, 28, 30]. Theoretically, these methods could be used to retrieve

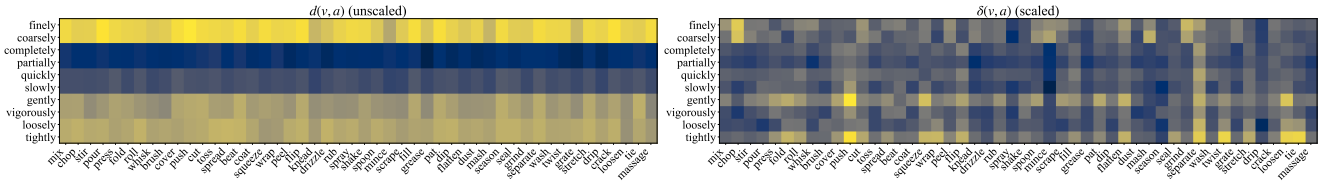

Figure 2. Comparison between the unscaled distance d (left) and the scaled distance δ (right), which is scaled using the verb-adverb cosine similarity. Plotted values are the Euclidean distances between corresponding positive and negative sentences. Blue/yellow indicate small/large distances, i.e. a small/big change when adverbs are flipped in the positive sentence (see text for more information).

videos using adverbs as text queries. However, as noted in [7], most approaches encode text embeddings for whole sentences, i.e. it is hard to retrieve videos querying single words. Some works propose fine-grained video-text retrieval where single parts-of-speech can be queried [4, 24, 27, 29], however these works focus on verbs and nouns and were not evaluated for adverbs. As discussed before, adverbs are more abstract compared to verbs and nouns and are difficult to spatio-temporally localise. This entails that a video-text retrieval approach is unlikely to work well to find adverbs in videos. We test a retrieval baseline with an S3D model [13] where video and text embeddings were jointly learnt. Despite the strong performance on a number of tasks, we show that indeed this model struggles to achieve satisfactory performance.

3. Method

We are given a video x labelled with a verb v, which annotates the action, and an adverb a, which indicates the way the action is carried out. Our goal is to predict a in x given v. We also have a mapping h between adverbs opposite in meaning, e.g. $h(\texttt{quickly}) = \texttt{slowly}$. We use a pre-trained video backbone f to extract video features $f(x)$. In this work we deal with long videos (10–40s), so we employ transformer-style attention [21] to attend to the relevant parts of the video. To this end, we use a pre-trained text model g to extract a text embedding $g(v)$, which we use as query for the attention, obtaining the projected video embedding $f'(x, v)$. To learn more specialised video features we feed $f'(x, v)$ to a shallow MLP and obtain the final output of the model $\rho(f'(x, v))$. We propose two methods to optimise the network casting the problem either as a classification or a regression task. In the former case we utilise the standard Cross Entropy (CE) loss using the ground truth adverb label a. The CE pushes up predictions for the positive class and treats all negative classes equally. In some cases, however, it may be desirable to penalise the model more aggressively for incorrect predictions of antonym classes. Action Modifiers [6] uses the triplet loss for this purpose. However, in [6] negative samples are formed by only pairing antonyms. This is effective for distinguishing opposite adverbs, but does not train the model to distinguish an adverb from other negative non-antonym classes.

We thus propose a second approach based on regression. Casting the task as a regression problem we aim to directly learn the change the adverb introduces to the action and the video. However, we do not have a ground truth quantifying such action modifications. We use a text embedding space to estimate action changes. The high level idea is that we first build a minimal sentence s representing the modified action. We similarly build a negative sentence $\tilde{s}$ representing the action modified in the opposite way. Calculating the difference between s and $\tilde{s}$ we have a proxy measure for the change the adverb applies to the action. We build s concatenating the verb and the adverb (e.g. `coat thinly`), then generate $\tilde{s}$ replacing the adverb with its antonym (e.g. `coat thickly`). We then extract text embeddings $g(s)$ and $g(\tilde{s})$. Note that g receives a variable length sentence and outputs a fixed-dimension embedding. Let $\langle\rangle$ denote concatenation, so $s = \langle v, a\rangle$ and $\tilde{s} = \langle v, h(a)\rangle$. We measure the difference between s and $\tilde{s}$ to capture how a changes v:

$$d(v, a) = \left\| g\big(\langle v, a\rangle\big) - g\big(\langle v, h(a)\rangle\big) \right\|_2 \tag{1}$$

Ideally, d could be used as is. However, we observed that for a given adverb a we obtain similar $d(v, a)$ across most verbs v. This defeats the purpose of measuring action changes via d since d itself is not discriminative enough. This happens because d ignores the correlation between the verb and the adverb. Intuitively, if a verb and an adverb are not semantically correlated (e.g. "run *thinly*") then we cannot expect to capture a meaningful change when flipping the adverb in the negative sentence (recall that g receives full sentences). To address this issue we scale d with the cosine similarity between the verb v and the adverb a:

$$\delta(v, a) = d(v, a) \cdot \frac{g(v) \cdot g(a)}{\|g(v)\|_2 \cdot \|g(a)\|_2} \tag{2}$$

Figure 2 compares d (left) to δ (right) for a few (verb, adverb) combinations. Note how the the unscaled distance d is not discriminative given that adverbs display a similar d across verbs. On the other hand, the scaled δ shows a meaningful distance. For example, we observe a large δ for adverbs "finely" and "coarsely" when paired with verbs "chop, mince, mash, grate".

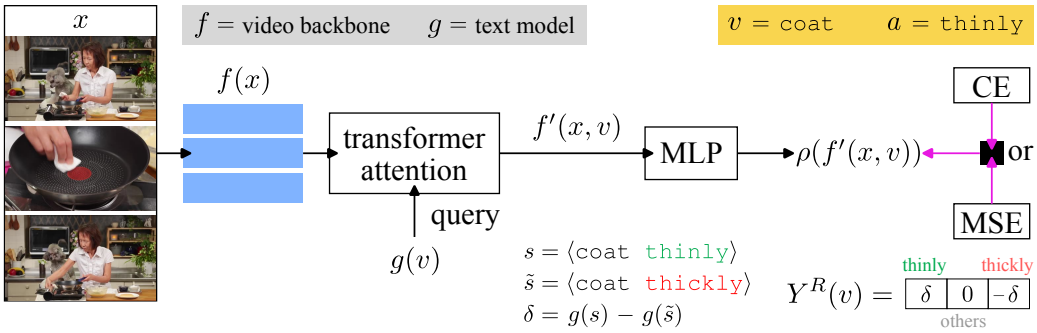

Figure 3. Pipeline of our method. Given a long video x labelled with a verb v and an adverb a we learn video embeddings $f'(x,v)$ through attention. We optimise the model with two alternative methods: standard classification (CE: Cross Entropy) or regression (MSE: Mean Squared Error). We build a regression target measuring distances in a text embedding space, which estimates the action change we aim to learn in the video. The video backbone and text model are initialised from pre-trained models and are not fine-tuned during learning.

Now that we have an estimate to measure action changes we can build a regression target:

$$Y^R(v) = (t_i, \forall i \in \{1,2,\dots,A\}) = \begin{cases} \delta(v,a) & \text{if } y_i = a \\ -\delta(v,a) & \text{if } y_i = h(a) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where A is the number of adverbs in a dataset and y_i denotes the i-th adverb. Let $\hat{y}_i^v$ indicate the output of the model for the i-th adverb obtained for verb v, i.e. querying the attention model with $g(v)$. The full output of the model is then $\hat{Y}^v = \rho(f'(x,v)) = (\hat{y}_i^v, \forall i \in \{1,2,\dots,A\})$.

We use the Mean Squared Error to train the network: $\mathcal{L}^R = \|Y^R(v) - \hat{Y}^v\|_2$. The loss pushes opposite adverbs furthest apart (due to the negative sign of δ for the antonym). The loss also tells the model that the video is not modified by other unrelated adverbs (target is 0 for classes that are neither ground truth nor antonyms). This discourages the model to predict negative non-antonym adverbs. Figure 3 illustrates our method. Note that the video backbone and text model are not part of the training, i.e. they are initialised from pre-trained models and are not fine-tuned.

Learning without Antonyms We also consider a setting where adverbs are not paired in antonyms. Instead of replacing the ground truth adverb with its antonym we simply remove the adverb and generate $\tilde{s} = v$. In this case δ measures the difference between the action modified by the adverb and the unmodified version of the action. Y^R is then δ for the ground truth adverb and 0 for all other adverbs.

Inference As in [6] we are given the action label v during testing. We query the attention model with $g(v)$ and use $\hat{Y}^v$ to predict each adverb. We also test the model without action labels. In such case we obtain $f'(x,v)$ querying all actions and take the maximum prediction for each adverb:

$$p(x) = (\max_v(\hat{y}_i^v, \forall v \in \mathcal{V}), \forall i \in \{1,2,\dots,A\}) \quad (4)$$

where $\mathcal{V}$ denotes the set of action labels.

Comparison to Action Modifiers Our framework shares the same design as Action Modifiers [6] until $f'(x,v)$. In [6] $f'(x,v)$ is optimised with two triplet losses: an action loss to help the attention model and an adverb loss to learn adverbs. We do not use an additional action loss. In Action Modifiers adverbs are also modelled as action changes utilising text embeddings. However, in [6] such change is treated as an additional parameter to be learnt. Specifically, an adverb is represented as a learnable matrix W_a. The linear combination $o_v^a = W_a g(v)$ represents the change the adverb a applies to the verb v, where $g(v)$ is a text embedding as in our model. Importantly, o_v^a is used in the triplet losses, thus the optimisation of the video embedding $f'(x,v)$ depends on the optimisation of parameters W_a. We argue and show that learning adverbs with this approach can be difficult as the action change the model looks for in the video has also to be learnt. In our formulation instead the action change is pre-computed and is not learnt. We also provide explicit penalty for non-antonym classes rather than contrasting only opposite adverbs. Comparing capacity, our model has a smaller number of parameters which scales better with the number of adverbs. Indeed, each adverb in Action Modifiers requires a learnable matrix, whereas in our model only the last layer of the MLP varies with the number of adverbs (see supplementary material).

4. Adverbs in Recipes: the AIR Dataset

In Section 2 we motivated the need for a new dataset for adverb recognition. Here we introduce the Adverbs in Recipes (AIR) dataset. We wish to find a set of videos where actions change significantly according to the adverb. Compared to captioning benchmarks, instructional datasets contain videos where we can expect to find more of such actions. For this reason we source AIR from HowTo100M [14]. We restrict our interest to recipe videos since changes in cooking actions play an important role. For example, a carrot sliced *à la julienne* (thinly) looks very different from a carrot chopped in thick dices (coarsely).

	Dataset	Accuracy	Duration	Adv	Act	Pairs	Videos
CAPT	ActivityNet Adverbs [7]	89.0%	37.5s	20	114	635	3,099
	MSR-VTT Adverbs [7]	91.0%	15.7s	18	106	450	1,824
	VATEX Adverbs [7]	93.5%	10.0s (f)	34	135	1,524	14,617
INST	HowTo100M Adverbs [6]	44.0%	20.0s (f)	6	72	257	5,824
	Adverbs in Recipes (ours)	95.3%	8.4s	10	48	186	7,003

Table 1. Comparing video datasets for adverb recognition. CAPT: captioning datasets, where adverbs are descriptive. INST: instructional datasets, where adverbs are action-focused. "Accuracy" indicates whether the action is visible as indicated by the adverb. "Duration" reports the average length, where (f) denotes a fixed duration. "Pairs" counts (verb, adverb) appearing in the dataset rather then the Cartesian product of all verbs and adverbs.

In other words, adverbs are "easily visible" in recipe videos and involve interesting visual cues such as speed, temporal/spatial completeness and objects end-states.

Data Collection We start selecting all recipe videos in HowTo100M. We then parse captions to keep only those containing a verb and an adverb. We filter videos removing verbs that indicate non visual actions (e.g. "watch, wait") or very long-term actions (e.g. "rise, grow"), since we do not expect these changes to be fully visible in the videos. We manually merge similar verbs, obtaining in total 48 verbs. We also remove infrequent and too generic adverbs (e.g. "carefully"). We cluster similar adverbs gathering a total of 10 adverbs, which we pair in antonyms. To trim videos in a tight manner we use the timestamps of the first/last word in a caption as the start/end of the action segment. At this point we have a set of videos of varying duration, each accompanied with a verb and an adverb. We ask annotators on Amazon Mechanical Turk (AMT) to check whether the action in a video is visible and is carried out as indicated by the adverb. For robustness we ask 3 different annotators to check the same video and employ 5 people for edge cases where annotators disagreed. We keep videos where the majority of the annotators confirmed that the action is visible as indicated by the adverb, collecting 7,003 videos. Figure 2 shows the verbs (columns) and adverbs (rows) in AIR. We provide more details in the supplementary material.

Comparison to Other Datasets Table 1 compares AIR to other adverb datasets[1] reviewed in Section 2. "Accuracy" estimates the percentage of videos where the action and the adverb effect are visible. On the existing datasets this was estimated watching 200 videos in [7]. For AIR multiple annotators reviewed all videos, however, annotations on AMT can be noisy. We review 302 videos (5% of the dataset) and report this check as "Accuracy". Compared

[1]We report the number of videos from [7]. Some videos are missing as they were removed from YouTube. We were able to download: ActivityNet A.: 2,972 - MSR-VTT A.: 1,747 - VATEX A.: 13,947, HowTo100M A.: 5,462. Duration and pairs are reported from the downloaded videos.

Figure 4. Samples from existing datasets for adverb recognition and our new AIR dataset. Many adverbs in the captioning datasets appear to be descriptive rather than a key modification of the action (e.g. the act of talking/sitting is the same whether it takes place indoors/outdoors). In contrast, adverbs in the instructional datasets apply a more prominent change in the action and its outcome (e.g. a vegetable peeled coarsely will still have some skin attached).

to the instructional-based HowTo100M Adverbs, AIR contains over 1,100 additional clips and 4 more adverbs. Importantly, our videos are manually reviewed (AIR accuracy is over 95% compared to 44%) and better trimmed (averaged duration is 8s compared to fixed 20s segments). AIR features 48 actions which is the smallest number among the five datasets (and consequently the smallest number of pairs). This is due to our more restrictive verb filtering based on visual and task-oriented actions. For example, in the captioning datasets there are actions such as "tell, talk, sing, look". Actions in HowTo100M Adverbs are task oriented, however there are also verbs entailing long-term actions (e.g. "brew"), which are unlikely to be fully visible.

Figure 4 shows a few samples from AIR as well as the other datasets. Note how adverbs in the instructional datasets apply a key modification in the action, whereas for the captioning datasets adverbs are descriptive. Captions in the the captioning datasets are reliable because they were provided by human annotators. However, annotators did not necessarily use adverbs to indicate a modification to the action. Thus, due to the automatic processing of these datasets, there are several cases where adverbs are descriptive and do not influence the action. Naturally, Figure 4 is not meant to be an exhaustive representation of the captioning datasets from [7]. These datasets also contain action-focused adverbs, however without extensively reviewing the videos and the associated adverbs it is hard to gauge how many actions are effectively modified by the adverb.

To summarise the contributions of AIR: i) we employ annotators to verify that actions are performed in the way indicated by the adverb; ii) this allowed us to trim videos more tightly and obtain cleaner clips with less unrelated content; iii) we carefully choose verbs and adverbs to pick videos where action changes are significant. While we restrict our domain to recipes, cooking actions are visually diverse, entail complex object interactions and are particularly affected by the nuanced ways they are carried out.

5. Experiments

Implementation Details We use an S3D model [13, 25] jointly pre-trained on video and text on HowTo100M [14] as our video backbone[2]. Specifically, we use the video output `mixed_5c` to obtain $f(x)$. Following [6] we split videos in 1-second segments and sample 16 RGB frames from each segment to obtain the video features. Stacking features from the video segments we have $f(x)^{T\times D}$, where T varies depending on the dataset and D is 1024. We use 4 heads for the attention model. The dimension E of the query, key and value is 512, thus the projected video embedding $f'(x, v)$ has dimension $1 \times E$. The MLP in our model has 3 layers (hidden units of dimension 512) with ReLU activation function for the hidden layers and dropout [20] set to 0.1. We train all models for 1000 epochs with the ADAM optimiser [10], setting learning rate to $1e^{-4}$, weight decay to $5e^{-5}$ and batch size to 512. We use the text model jointly trained with S3D to extract text embeddings $g(.)$. This network, like the visual backbone, is not fine-tuned. We refer to [13] for more details about S3D and the text model. $f(x)$ and $g(.)$ are the same for all experiments.

Baselines *Priors:* here we do not train any model and simply use the prior distributions of the training set to make a prediction. When using the action label during testing, the prediction for an adverb is the number of training samples labelled with the adverb and the given verb, i.e. $p(x, v) = (\pi(v, a)\ ,\ \forall a \in \mathcal{A})$, where π denotes the frequency of (v, a) in the training set and $\mathcal{A}$ is the set of adverbs in a dataset. While this baseline is a simple look-up table, it can achieve strong performance since it exploits the co-occurrence of verbs and adverbs. When the action label is not used during testing the priors baseline predicts the frequency of each adverb in the training set, i.e. $p(x) = (\pi(a)\ ,\ \forall a \in \mathcal{A})$.

S3D pre-trained: we test the S3D model we use as the video-text backbone for all experiments. Video and text embeddings were jointly learnt in this model, so we use the dot product between $f(x)$ and $g(.)$ for predictions. When the action label v is given during inference the prediction is $p(x, v) = (f(x) \cdot g(\langle v, a\rangle)\ ,\ \forall a \in \mathcal{A})$, where $g(\langle v, a\rangle)$ is the text embedding obtained concatenating the given v and each adverb a. When the action label is not given, we have $p(x) = (\max_v(f(x) \cdot g(\langle v, a\rangle)\ ,\ \forall v \in \mathcal{V})\ ,\ \forall a \in \mathcal{A})$, where $\mathcal{V}$ is the set of verbs. This S3D instance has shown very strong performance on several tasks [13], thus this baseline helps gauging the difficulty of detecting adverbs following a basic video-text retrieval approach.

[2]There is no official train/test split in HowTo100M, thus S3D in [13] was trained on videos that appear in the test sets of HowTo100M Adverbs and AIR. However, this is not a concern: Tables 2, 3 show that the S3D pre-trained baseline achieves poor performance, as the original S3D training objective (video-text alignment) is substantially different from ours. Importantly, all methods receive the same features for fair comparison.

SOTA *Act Mod:* we compare against the state-of-the-art for adverb recognition, Action Modifiers [6], both using the original model and a deeper version with an additional MLP (*MLP + Act Mod*). Here we append an identical MLP as in our model to the output of the transformer attention. This variant allows a better comparison with our method. We use the official code implementation to run our experiments. Note that Action Modifiers results reported here are not directly comparable to those reported in [6,7]. This is because we could not download several videos since they were removed from YouTube. Also, we use S3D features (identical for all experiments) instead of I3D [3] features.

Our Variants *CLS:* our model trained with classification. *REG - fixed δ*: our model trained with regression discarding text embeddings altogether, setting $\delta = 1$ in Equation 3. This is to validate the premise of using textual context to build a regression target. *REG*: our model trained with regression using the full formulation.

Evaluation Metrics We report mean Average Precision using two types of averaging: i) weighted (mAP W), where class scores are weighted according to the support size (smaller classes have a smaller weight); ii) macro (mAP M), where all classes have equal weight, which corresponds to "adverb-to-video (all)" in [6]. All the evaluated datasets exhibit a significant class imbalance, thus mAP M is a stricter metric compared to mAP W. We also report binary antonym accuracy (Acc-A), which corresponds to "video-to-adverb (antonym)" in [6]. Here we only look at the prediction of an adverb a versus its antonym $h(a)$. A prediction p is correct if $p(a) > p(h(a))$. All metrics are computed for adverb classes. To show the full potential of all methods we report each best metric independently, i.e. results may come from different epochs. This is to provide a robust comparison, however we note that all models reach stable convergence.

Datasets We present experiments on the two instructional datasets we saw earlier: HowTo100M Adverbs [6] and our Adverbs In Recipes. As discussed before, we argue that instructional videos are more suitable for this task as adverbs are action-focused, i.e. actions change significantly according to the adverb. For completeness, we also evaluate the three adverb datasets sourced from captioning videos in [7]: ActivityNet/MSR-VTT/VATEX Adverbs[3]. Adverbs here are descriptive, thus these datasets are less reliable for evaluation. Datasets are summarised in Table 1.

5.1. Results

Overview We begin our discussion looking at results in Table 2 on the instructional datasets (left). Here we achieve new state-of-the-art results on all metrics. The gap between our best variant and *Act Mod* is more noticeable for the more challenging mAP W/M metrics. In [6] negative samples are

[3]For these datasets there are several partitions for different tasks (e.g. unseen domain). For convenience we create our own train/test split.

	Instructional datasets. Action-focused adverbs						Captioning datasets. Descriptive adverbs								
	HowTo100M Adverbs [6]			Adverbs in Recipes			ActivityNet Adverbs [7]			MSR-VTT Adverbs [7]			VATEX Adverbs [7]		
	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A
Priors	0.446	0.354	0.786	0.491	0.263	0.854	**0.217**	**0.159**	0.745	**0.308**	**0.152**	0.723	0.216	0.086	0.752
S3D pre-trained	0.339	0.238	0.560	0.389	0.173	0.735	0.118	0.070	0.560	0.194	0.075	0.603	0.122	0.038	0.586
Act Mod [6]	0.406	0.372	0.796	0.421	0.170	0.858	0.211	0.152	0.748	0.246	0.148	0.734	0.139	0.059	0.751
MLP + Act Mod [6]	0.279	0.193	0.793	0.458	0.188	0.857	0.154	0.108	**0.753**	0.215	0.111	0.731	0.209	0.096	0.752
CLS	**0.562**	0.420	0.786	**0.675**	**0.325**	0.847	0.114	0.072	0.714	0.189	0.081	0.723	**0.283**	**0.108**	0.754
REG - fixed δ	0.320	0.215	0.706	0.458	0.145	0.835	0.113	0.070	0.706	0.221	0.096	0.706	0.175	0.051	0.701
REG	0.555	**0.423**	**0.799**	0.669	0.297	**0.862**	0.183	0.103	0.709	0.292	0.118	**0.774**	0.261	0.086	**0.755**

Table 2. Results obtained using the action label during inference. mAP W/M: mean Average Precision with weighted (W) and macro (M) averaging. Acc-A: adverb-vs-antonym accuracy. Coloured rows indicate variants of our method. Bold denotes best result per column. In instructional datasets (left) adverbs are action-focused, so these are more reliable benchmarks to learn action changes. In captioning datasets (right) adverbs are more descriptive and do not influence the action significantly. As such, these datasets are less reliable.

formed only pairing opposite adverbs, thus the model does not receive enough penalty for other negative adverbs. In contrast, we provide explicit penalty for non-antonym negative adverbs. The poor performance of *S3D pre-trained* indicates that learning adverbs in videos is not trivial. Adverbs are not well defined visual entities like objects or actions, thus video-text retrieval approaches looking for correlations between text and video embeddings are likely sub-optimal, even when such embeddings are jointly learnt on the same dataset with a strong model, as is the case for S3D.

Still looking at the instructional datasets, on AIR all methods achieve higher mAP W/Acc-A but lower mAP M. Considering that there are more adverbs in AIR than in HowTo100M Adverbs, the higher mAP W/Acc-A is indicative of the quality of AIR: given that videos are tighter and cleaner it is possible to learn adverbs more effectively. The lower mAP M is due to the long-tail class distribution and shows the opportunities AIR offers to improve methods.

On the captioning datasets we observe an interesting result: the *Priors* baseline outperforms or achieves very high performance. This supports our argument that captioning datasets are not particularly suitable for our task. If a simple look-up table surpasses all learning-based methods, then we conclude it is very difficult to learn adverbs as action changes, because actions are not particularly influenced by adverbs here. While these datasets exhibit a skewed verb-adverb distribution, so do the instructional ones. In fact, on VATEX Adverbs (the biggest dataset with 1,524 pairs) our method attains marginally better results. This is to say that the problem does not lie in the imbalanced class distribution, rather in the nature of the videos.

Importance of Context We now focus on the importance of using textual context for regression comparing *REG - fixed* δ to *REG* in Table 2. Visual features vary differently when the same action is modified by different adverbs, e.g. chopping something coarsely entails a more prominent change than chopping it slowly. With REG we learn such visual differences with targets that change (sometimes considerably) across adverbs for a given verb. A fixed target for all positive adverbs discards such differences: the model is penalised equally for small/big changes. *REG - fixed* δ does this, and its consistent worse performance validates the idea of using text context as a proxy to learn action changes.

The Key is in the Optimisation We now compare *MLP + Act Mod* to our method. For this *Act Mod* variant we replace the video embedding $f'(x, v)$ with $\theta(f'(x, v))$ in the two losses in [6], where θ is the additional MLP. With this configuration, the only difference between our method and Action Modifiers lies in the optimisation. In Section 3 we hypothesised that the optimisation of Action Modifiers might be difficult. Results in Table 2 support our hypothesis. When adding the MLP to *Act Mod,* Acc-A remains virtually identical across all datasets. However mAP W/M drop considerably on HowTo100M Adverbs. This dataset is the noisiest among the five datasets, thus in this case increasing the depth of the model makes optimisation more difficult. Overall, comparing *MLP + Act Mod* to our method we achieve better results even when the additional MLP improves the performance of the original *Act Mod*. We conclude that learning adverbs as pre-defined actions changes rather than as trainable parameters is a better strategy.

Testing without Action Labels In Table 3 we report results obtained without using the action label at test time. We test Action Modifiers in the same way as described for our method in Section 3. Experiments are directly comparable between Tables 2 and 3 since the tested models are the same. We note a remarkable drop in performance for all methods. This is expected since the task is now harder. In fact we need to predict the right adverb from multiple video embeddings obtained querying all verbs. Nevertheless, our method still achieves the best global performance in this more challenging setting. This shows that our method is more robust and is able to better generalise.

Learning without Antonyms We now study the case where adverbs are not paired in antonyms. This is a desirable setting as it allows more flexibility in data collection. To train Action Modifiers without antonyms we just sample a random negative adverb as opposed to the antonym for the triplet loss. We illustrated in Section 3 how we train our *REG* variant. We do not evaluate *REG - fixed* δ here since

	Instructional datasets. Action-focused adverbs						Captioning datasets. Descriptive adverbs								
	HowTo100M Adverbs [6]			Adverbs in Recipes			ActivityNet Adverbs [7]			MSR-VTT Adverbs [7]			VATEX Adverbs [7]		
	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A
Priors	0.247	0.167	0.560	0.335	0.100	0.835	0.094	0.050	0.692	0.137	0.056	0.723	0.089	0.029	0.651
S3D pre-trained	0.329	0.222	0.594	0.425	0.177	0.702	0.113	0.065	0.598	0.199	0.088	0.603	0.117	0.037	0.604
Act Mod [6]	0.269	0.183	0.601	0.346	0.107	0.837	0.110	0.063	**0.718**	0.164	0.072	0.729	0.100	0.035	0.686
MLP + Act Mod [6]	0.268	0.184	0.706	0.368	0.113	0.835	0.109	0.070	0.714	0.162	0.097	0.723	0.092	0.036	0.686
CLS	**0.404**	**0.307**	**0.724**	0.469	**0.198**	0.835	0.114	0.072	0.714	0.189	0.078	0.723	0.161	0.053	0.712
REG - fixed δ	0.320	0.215	0.706	0.461	0.148	0.835	0.113	0.070	0.706	0.213	0.095	0.714	0.158	0.047	0.703
REG	0.377	0.277	**0.724**	**0.481**	0.157	**0.857**	**0.120**	**0.073**	0.706	**0.240**	**0.102**	**0.749**	**0.169**	**0.057**	**0.737**

Table 3. Results obtained *without* action labels during inference. mAP W/M: mean Average Precision with weighted (W) and macro (M) averaging. Acc-A: adverb-vs-antonym accuracy. Coloured rows indicate variants of our method. Bold denotes best result per column.

		Instructional datasets. Action-focused adverbs				Captioning datasets. Descriptive adverbs					
		HowTo100M Adverbs [6]		Adverbs in Recipes		ActivityNet Adverbs [7]		MSR-VTT Adverbs [7]		VATEX Adverbs [7]	
	Antonyms	mAP W	mAP M	mAP W	mAP M	mAP W	mAP M	mAP W	mAP M	mAP W	mAP M
Priors	✗	0.446	0.354	0.491	0.263	**0.217**	0.159	0.308	0.152	0.216	0.086
S3D pre-trained	✗	0.339	0.238	0.389	0.173	0.118	0.070	0.194	0.075	0.122	0.038
Act Mod [6]	✓	0.406	0.372	0.421	0.170	0.211	0.152	0.246	0.148	0.139	0.059
	✗	0.408	0.352	0.428	0.180	**0.217**	**0.160**	0.254	**0.154**	0.144	0.060
MLP + Act Mod [6]	✓	0.279	0.193	0.458	0.188	0.154	0.108	0.215	0.111	0.209	0.096
	✗	0.281	0.198	0.465	0.197	0.164	0.115	0.229	0.132	0.213	0.098
CLS	✗	0.562	0.420	0.675	0.325	0.114	0.072	0.189	0.081	**0.283**	**0.108**
REG	✓	0.555	0.423	0.669	0.297	0.183	0.103	0.292	0.118	0.261	0.086
	✗	**0.573**	**0.481**	**0.716**	**0.393**	0.206	0.121	**0.319**	0.114	0.282	0.100

Table 4. Results obtained with action labels during inference and *without* antonyms during training. mAP W/M: mean Average Precision with weighted (W) and macro (M) averaging. Coloured rows indicate variants of our method. Bold denotes best result per column.

without antonyms it would be similar to binary classification. *CLS* by design does not use antonyms. Table 4 compares results obtained with and without antonyms. Since we assume we do not have antonym labels here we only evaluate mAP W/M. Interestingly, Action Modifiers slightly improves when removing antonyms, which shows again that using only opposite adverbs for contrastive learning limits the model. Discarding antonyms for our regression variant has also a strong positive effective. Since we remove the target $-\delta$ for antonyms and treat opposite adverbs as any other negative class, we are not forcing the model to push antonyms further compared to other negative adverbs. This is beneficial to learn adverbs more generally.

Regression vs Classification We now compare *CLS* and *REG* in Tables 2, 3 and 4. When testing without action labels (Table 3) and learning without antonyms (Table 4) we notice a more consistent improvement of *REG* over *CLS*. Rather than maximising the margin between the positive adverb and all other negative adverbs as in classification, with regression we try to learn the extent of the change applied by an adverb. This is evidently a more effective way to learn adverbs since the visual change introduced by an adverb varies considerably according to the action.

6. Conclusion

We proposed to address adverb recognition as a regression task. Our method attains new state-of-the-art performance on multiple datasets. Importantly, we achieve compelling results when removing two major assumptions: the availability of action labels during testing and the pairing of opposite adverbs as antonyms. To address the scarcity of datasets for adverb recognition we introduced AIR. The dataset collects instructional recipe videos where actions are particularly influenced by the modification indicated by adverbs. We believe AIR is a valuable resource that can readily foster advance in adverb understanding.

Limitations Since we model action changes measuring distances in a text embedding space we rely on a reasonably good text model. If the model does not capture meaningful relationships between verbs and adverbs then we do not have a reliable proxy to learn adverbs. Because we look for action changes directly in the video, actions need to exhibit a sufficient visual change when modified by the adverb.

Future Work Directions for future work include exploring ways to capture a broader action context via text. For example, the full caption associated with a video can provide additional information (e.g. object nouns) that could be exploited to better learn action modifications. This would require addressing the problem of video-text alignment, as well as the intrinsic noisy nature of long text embeddings.

Potential Societal Impact We source AIR from HowTo100M dataset, which gathers videos from YouTube. As such, any societal bias introduced in HowTo100M is potentially included in our data and our trained models.

Acknowledgements Research funded by UKRI through the Edinburgh Laboratory for Integrated Artificial Intelligence (ELIAI) and the Turing Advanced Autonomy project.

References

[1] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *ECCV*, 2018. 1

[2] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 2

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 6

[4] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *CVPR 2020*, 2020. 3

[5] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *CVPR*, 2019. 2

[6] Hazel Doughty, Ivan Laptev, Walterio Mayol-Cuevas, and Dima Damen. Action modifiers: Learning from adverbs in instructional videos. In *CVPR*, 2020. 2, 3, 4, 5, 6, 7, 8

[7] Hazel Doughty and Cees GM Snoek. How do you do it? fine-grained action understanding with pseudo-adverbs. In *CVPR*, 2022. 2, 3, 5, 6, 7, 8

[8] Yin Fan, Xiangju Lu, Dian Li, and Yuanliu Liu. Video-based emotion recognition using cnn-rnn and c3d hybrid networks. In *ACM - Multimodal interaction*, 2016. 2

[9] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, 2017. 2

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

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

[12] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *BMVC*. 2

[13] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*, 2020. 2, 3, 6

[14] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019. 2, 4, 6

[15] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *ACM - Multimedia Retrieval*, 2018. 2

[16] Davide Moltisanti, Michael Wray, Walterio Mayol-Cuevas, and Dima Damen. Trespassing the boundaries: Labeling temporal bounds for object interactions in egocentric video. In *ICCV*, 2017. 1

[17] Jonghwan Mun, Minsu Cho, and Bohyung Han. Local-global video-text interactions for temporal grounding. In *CVPR*, 2020. 2

[18] Bo Pang, Kaiwen Zha, and Cewu Lu. Human action adverb recognition: Adha dataset and a three-stream hybrid model. In *CVPR (workshop)*, 2018. 2

[19] Bo Pang, Kaiwen Zha, Yifan Zhang, and Cewu Lu. Further understanding videos through adverbs: A new video task. In *AAAI*, 2020. 2

[20] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. In *JMLR*, 2014. 6

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

[22] Weining Wang, Yan Huang, and Liang Wang. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *CVPR*, 2019. 2

[23] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*, 2019. 2

[24] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*, 2019. 3

[25] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 6

[26] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 2

[27] Ran Xu, Caiming Xiong, Wei Chen, and Jason Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015. 3

[28] Jianwei Yang, Yonatan Bisk, and Jianfeng Gao. Taco: Token-aware cascade contrastive learning for video-text alignment. In *ICCV*, 2021. 2

[29] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, 2017. 3

[30] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. Dense regression network for video grounding. In *CVPR*, 2020. 2