Supplementary Material
HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips

Antoine Miech[1,2*] Dimitri Zhukov[1,2*] Jean-Baptiste Alayrac[2+]
Makarand Tapaswi[2] Ivan Laptev[1,2] Josef Sivic[1,2,3]
[1]École Normale Supérieure [2]Inria [3]CIIRC, CTU
https://www.di.ens.fr/willow/research/howto100m

Overview of the Supplementary Material

We present additional details of our HowTo100M dataset in 1. We also provide practical implementation details of our ranking loss in 2 and analyze the sampling strategy for positive pair selection during training in 3.

1. Additional details of the HowTo100M dataset

Our HowTo100M dataset is based on the hierarchy of WikiHow[1] tasks. The HowTo100M spans a total of 23,611 tasks. Here we visualize the first two levels of the WikiHow hierarchy – the twelve categories and their subcategories, the number of underlying tasks and corresponding videos are illustrated in Figure 2.

HowTo100M comes with transcribed narrations which often describe the content of the videos. Figure 3 shows frequencies of nouns and verbs in transcribed video narrations. We used the MaxEnt Treebank POS Tagger to obtain the nouns and verbs. Please see the figure captions for additional analysis.

2. Ranking loss implementation details

In the main paper, we have defined our mini-batch ranking loss as:

$$\sum_{i\in\mathcal{B}}\sum_{j\in\mathcal{N}(i)}\max(0,\delta+s_{i,j}-s_{i,i})+\max(0,\delta+s_{j,i}-s_{i,i}). \quad (1)$$

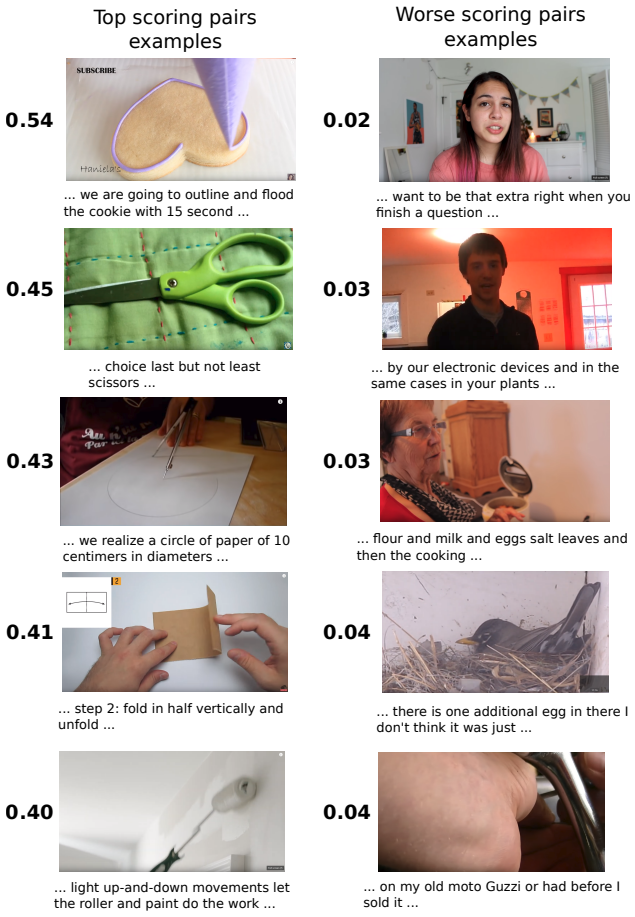

Figure 1: We illustrate examples of high and low scoring clip-caption pairs. Examples from the left column show pairs where the caption visually describes what is seen in the corresponding video clip. On the other hand, low scoring pairs from the right column have captions that do not match visual content.

We explain next how $\mathcal{N}(i)$ is constructed to improve computational efficiency.

*Equal contribution.
+Now at DeepMind.
[1]Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, 75005 Paris, France.
[3]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.
[1]https://www.wikihow.com/

At each training iteration, we first sample v unique YouTube video ids. We then sample with replacement a number k of clip-caption pairs from each of these videos. Therefore, we are left with a mini-batch containing $b = kv$ clip-caption pairs, with $v = 32$ and $k = 64$ in practice. In order to not waste computation efforts, we use every sampled mini-batch pair as a negative anchor, *i.e.* $\mathcal{N}(i) = \mathcal{B} \setminus \{i\}, \forall i$.

Doing so, the proportion of negative examples coming from the same video (*intra-video*) is $\frac{k-1}{kv-1}$ while the proportion of negatives from different videos (*inter-video*) is $\frac{k(v-1)}{kv-1}$. A problem with this is that the ratio between *intra* and *inter* video negative examples depends on the number of unique videos sampled and the amount of clip-caption pairs collected per video (respectively v and k). To address this, we follow [1] by re-weighting the inter-video and intra-video contributions inside the triplet loss. For example, in order to sample intra-video triplets with probability $p \in [0, 1]$ (and inter-video triplets with probability $1-p$), one can equivalently weight the intra-video triplet losses by: $\alpha = \frac{pk(v-1)}{(1-p)(k-1)}$ (thus ensuring a ratio between intra-video and inter-video negative examples of $\frac{p}{1-p}$). This allows us to fix the intra-video to inter-video negative sampling ratio regardless of v and k. Formally, we define the following weighting function:

$$\alpha_{i,j} = \begin{cases} \frac{pk(v-1)}{(1-p)(k-1)} & \text{if } i \text{ and } j \text{ are from same video,} \\ 1, & \text{otherwise.} \end{cases} \quad (2)$$

We then use this weighing function to define the loss:

$$\sum_{i \in \mathcal{B}, j \in \mathcal{N}(i)} \alpha_{i,j} \Big[\max(0, \delta + s_{i,j} - s_{i,i}) + \max(0, \delta + s_{j,i} - s_{i,i}) \Big].$$

3. Sampling strategy for positive pairs

As discussed in the main paper, narrations need not necessarily describe what is seen in the video. As a consequence, some captions from HowTo100M do not correlate with their corresponding video clips (see Figure 1). To deal with this noisy data, we tried a sampling strategy for positive pairs that aims to discard non-relevant video-caption pairs during training. Inspired by multiple instance learning, our idea is to select a subset of top scoring clip-caption training pairs within each video.

In particular, given a video with N video clip-caption pairs $\{(V_i, C_i)\}_{i \in [1,N]}$, we first compute the similarity scores of all the N pairs: $s(V_i, C_i)$ using the current model parameters. We then use a pre-defined max-pool rate $r \in [0, 1]$ of the highest scoring positive training pairs $\{(V_i, C_i)\}_{i \in [1,N]}$ within each video. For example, at $r = 0.5$ we retain the high scoring half of all N pairs for training.

Max pool rate (r)	M (R@10)	L (R@10)	Y (R@10)
0.2	21.9	13.9	19.7
0.5	25.2	12.6	23.5
0.9	27.3	12.6	23.9
1.0 (no max pool)	**29.6**	**14.0**	**24.8**

Table 1: Study of positive pair sampling. When max pool rate r is below 1.0 only the proportion r of top scoring clip-caption pairs are used for learning. We report R@10 retrieval results from M: MSR-VTT, L: LSMDC, Y: YouCook2.

MP rate	RS rate	M (R@10)	L (R@10)	Y (R@10)
1.0	0.5	**28.8**	**14.3**	**24.2**
0.5	1.0	25.2	12.6	23.5

Table 2: Study of Random Sampling (RS) vs. Max Pool (MP) sampling of positive clip-caption pairs. We report R@10 retrieval results from M: MSR-VTT, L: LSMDC, Y: YouCook2.

Table 1 shows results of our positive sampling strategy when varying the max pool rate r with evaluation on video clip retrieval. For example, $r = 1.0$ means that no sampling strategy is applied as we keep all N pairs as potential candidates. Interestingly, in our case, carefully selecting the positive pairs does not improve our model as the best results are obtained with $r = 1.0$. Note that decreasing the max pool rate also decreases the number of triplet losses computed within a mini-batch by the same rate. To show that the number of triplet losses computed for each mini-batch does not impact the overall performance, we have performed a sanity check experiment in Table 2 in which we also replaced the max pool sampling by random sampling of pairs for $r = 0.5$. The results with random sampling at $r = 0.5$ are very similar to the results obtained with no max pool sampling (r=1.0) as shown in Table 1, which confirms our finding that our model is relatively robust to the noisy positive pairs. We think this could be attributed to the fact our model is shallow and is trained on a large amount of data.

References

[1] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with natural language. *ICCV*, 2017. 2

Figure 2: The first two levels of hierarchy of tasks in the HowTo100M dataset. Our dataset includes 12 categories from WikiHow containing 129 subcategories. For each (sub)category we show the total number of collected tasks and clips. This hierarchy of tasks in our dataset follows the WikiHow structure. Please recall that abstract tasks such as Choosing a gift or Meeting new friends, were not considered and were removed from the WikiHow hierarchy semi-automatically by verb analysis, as described in Section 3.1 of the main paper. As a result, the category tree is imbalanced. For example, the *Dining Out* subcategory includes only one physical task (*Fix a Shaky Table at a Restaurant*), while *Recipes* subcategory from the same level of the hierarchy includes a large number of tasks and clips.

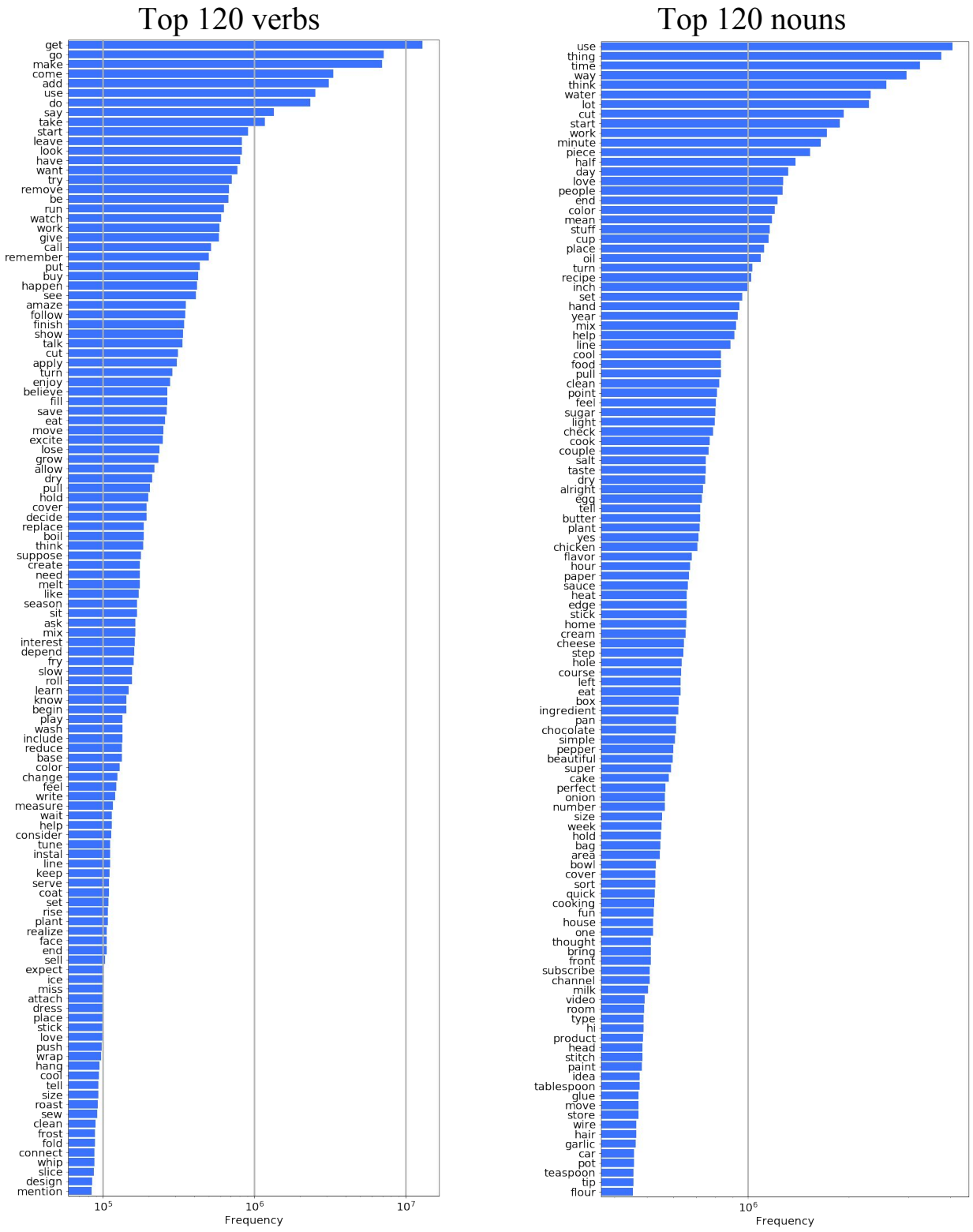

Figure 3: Frequencies of the top 120 most commonly occurring nouns and verbs in our dataset. Note that our dataset is biased towards physical actions, with verbs such as *get*, *go* and *make* being the most frequent, while verbs, such as *be*, *know* and *think* are less frequent than in common English. Top nouns show the dominant topics in our instructional videos. In particular, many cooking-related words, such as *water*, *oil* and *sugar* occur with high frequency.