Universiteit Leiden
The Netherlands

# Data Science & Artificial Intelligence

Beyond Coarse-Grained Matching in Video-Text Retrieval

Maksym Lytovka

Supervisors:
Hazel Doughty and Luc Sträter

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

30/06/2025

**Abstract**

Video-text retrieval is a rapidly evolving task in computer vision, yet current models often struggle to distinguish between captions with subtle semantic differences. In this work, we take a step toward addressing this limitation by investigating the impact of augmenting training datasets with hard negatives and hard positives. We compare two approaches for generating these challenging examples: a rule-based method leveraging part-of-speech tagging with synonym/antonym substitution, and a data-driven method using large language models (LLMs). Our experiments explore various configurations, including the number of hard negatives, the inclusion and ratio of hard positives, and different generation and sampling strategies. We evaluate performance using standard metrics—Recall@k and Mean Rank—as well as two emerging, fine-grained evaluation metrics from recent research: PosRank, which measures a model's ability to prioritize correct captions among semantically similar ones, and Brittleness, which captures the model's sensitivity to minor textual variations. Our findings offer three key insights: (1) when using only one type of hard example for augmentation, hard positives lead to more robust and sustainable performance improvements than hard negatives; (2) LLM-generated examples consistently outperform rule-based substitutions; and (3) for text-to-video retrieval tasks, incorporating both hard positives and negatives yields the best results, even if it slightly compromises performance on vision-to-text retrieval.

**Key words:** Video-Language · Video-Text Retrieval · Fine-grained · Hard Positives

**Code:** https://github.com/Reymer249/Thesis_X-CLIP

# Contents

# 1 Introduction

Computer vision plays an important role in many technologies we use every day. Examples include FaceID for unlocking phones, automatic tagging of photos on social media, real-time translation using the phone camera, and driver-assistance systems in cars. Among the active areas of research are **image-text** and **video-text retrieval**. These tasks aim to align visual content with corresponding textual descriptions—images in the former and videos in the latter. In this thesis, we focus specifically on the task of video-text retrieval.

A good example of this task is YouTube. In its early years, YouTube could only search for videos using metadata such as the title or description. A more advanced system would allow users to search using the actual content of a video. For instance, if a user remembers a scene with a "pink dog," they should be able to find that video by typing this phrase, even if the title or description does not mention it. This type of search is called video-text retrieval. While platforms like YouTube have started using this technology, they still rely mostly on metadata.

To understand video-text retrieval, it helps to first look at image-text retrieval, which follows the same idea but with images. Imagine a large collection of images, each paired with a short text describing what is in it. Given a text query, the goal is to find and rank the images from most to least relevant (see Figure 1).

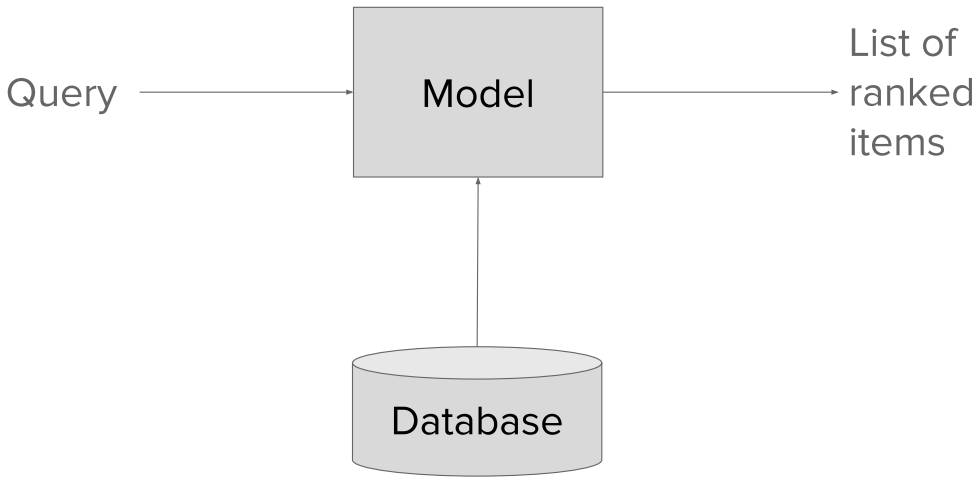


Figure 1: Overall schema of the image-text retrieval task

As an example, consider a database of memes. If a user is looking for a specific meme (Figure 2), the system should return relevant results based on a textual query. The process also works in reverse: given an image, the model should retrieve and rank matching text descriptions from the database. Moreover, the system can handle queries or images that were not part of the original dataset, which is a key advantage of such models.

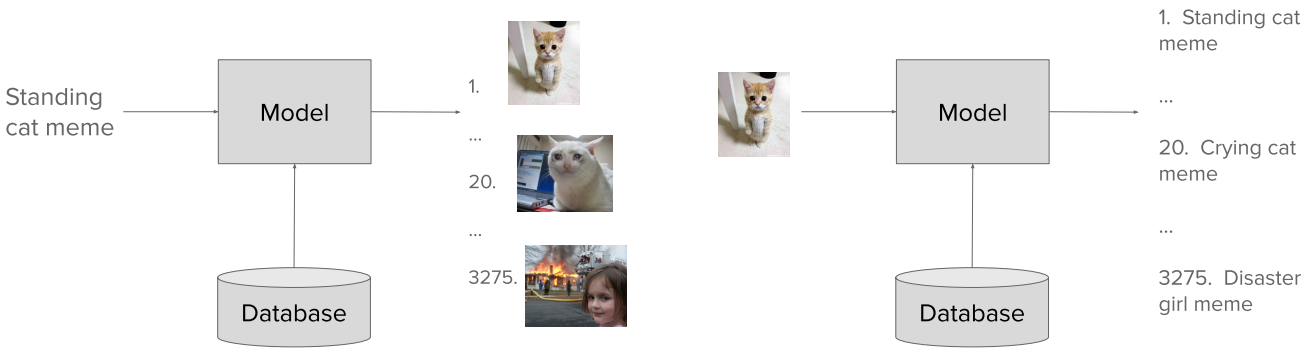


Figure 2: Illustrations of the image-text retrieval task on memes

## 1.1 Problem Definition

Although recent approaches to video-text retrieval have achieved strong performance on standard metrics such as Recall@k and Mean Rank (see Section 4.3 for definitions), these scores do not necessarily indicate a deep understanding of the connection between videos and their textual descriptions. Studies have revealed that current systems often fail to distinguish between captions that differ by subtle but critical details [1]. For example, consider Figure 3a.

We define **fine-grained negatives**—mainly referred to as **hard negatives**—as sentences that are very similar to the original caption but differ in a minor detail. These differences might involve changing a single word, such as the object, a property of the object, the action being performed, or some part of the surrounding context. Because they are close in meaning but not semantically identical, these examples are particularly challenging for models to distinguish. In contrast, **coarse-grained negatives** refer to sentences that are clearly different from the original and do not share much semantic similarity.

As Chen et al. [1] point out, current state-of-the-art models often fail to detect these fine-grained or hard negative differences, especially when they involve subtle changes such as prepositions or adverbs.

This difficulty stems from two main factors: the training procedure and the nature of the data. First, the commonly used loss functions (see Section 3.1) are not designed to encourage the model to learn such fine distinctions. They primarily optimize for recall, which does not directly reward sensitivity to small semantic changes. Therefore, a more targeted training approach is needed—one that helps the model recognize small differences without reducing its overall retrieval performance.

Second, the datasets themselves are often coarse in nature. Most examples are far apart in meaning, making it difficult for the model to learn to differentiate between closely related concepts. For instance, in Figure 3b, the nearest captions to a given sentence are still quite different in meaning. We refer to such training as *coarse-grained*, and this lack of fine-grained or hard negative examples in the dataset is a key issue addressed in this work.

## 1.2 Research Questions and Contributions

This paper investigates the effect of incorporating hard positives into retrieval model training on video-text retrieval efficiency and the model's comprehension of objects and actions within video


**a black car is driving down the road**

➢ **Coarse-grained Negatives**

a person is connecting something to system
this is a video of a live tv show
people are singing on the beach
a little girl does gymnastics
a boy is singing

➢ **Fine-grained Negatives**

a **white** car is driving down the road
a black **motorcycle** is driving down the road
a black car is **parked** down the road
a black car is driving **across** the road
a black car is driving down the **mountain**

(a) Coarse-grained Negatives and Positives illustration

**a man in sunglasses and a blue shirt beat boxes**

➢ ***Top 5 closest captions***

a man with glasses and a goatee talking about his former job
a man is singing
a man with brown hair is singing a song
a boy is singing
a man is yelling on the phone

(b) Example of a data point from MSR-VTT[2] dataset. Noticeably, the 5 closest captions in the dataset are quite distant in meaning.

Figure 3: Examples to illustrate the problem with coarse-grained training. Both images were taken from Chen et al. [1]

content. Our work extends existing research that has primarily focused on hard negatives in video retrieval systems (Chen et al. [1]) by systematically exploring the complementary role of hard positives.

We examine several key aspects of hard positive integration: the optimal number of hard positives and their proportion relative to hard negatives, alternative methods for generating hard sentences, and variations in sampling strategies during training. Our experimental findings reveal that hard positives provide more sustainable performance improvements compared to hard negatives when employed independently. Additionally, we demonstrate that models trained with hard sentences generated through large language models outperform those using sentences created via parts-of-speech tagging. While the combination of hard negatives and positives yields the best text-to-video retrieval performance, we observe a trade-off where vision-to-text retrieval performance diminishes.

This research makes three primary contributions to the field: (i) we replicate and verify the findings of Chen et al. [1] while extending their work to encompass hard positives, (ii) we provide a comprehensive analysis of how hard negatives affect both standard and fine-grained retrieval metrics, examining the impact of their quantity and quality, and (iii) we conduct a comparative evaluation of two distinct generation methods for creating hard negatives and positives. To facilitate future research, we have developed and released a comprehensive framework for integrating hard positives and negatives into X-CLIP model training, which is publicly available on our research GitHub repository.

# 2 Related Work

## 2.1 Training and Usage

Video-text retrieval is an important topic in the vision-language domain and has received increasing attention in recent years [3, 4, 5, 6, 7, 8]. The most commonly used training method for retrieval models is **contrastive loss**, which has gained popularity due to its ability to leverage large-scale, uncaptioned image datasets available online. This approach was formalized by Chen et al. [9] in the SimCLR framework, which introduced contrastive learning for image data. The core idea is to apply augmentations to different parts of the same image and bring their embeddings closer together, while pushing apart the embeddings of different images (Figure 4). This allows the model to learn meaningful visual representations without requiring labels.

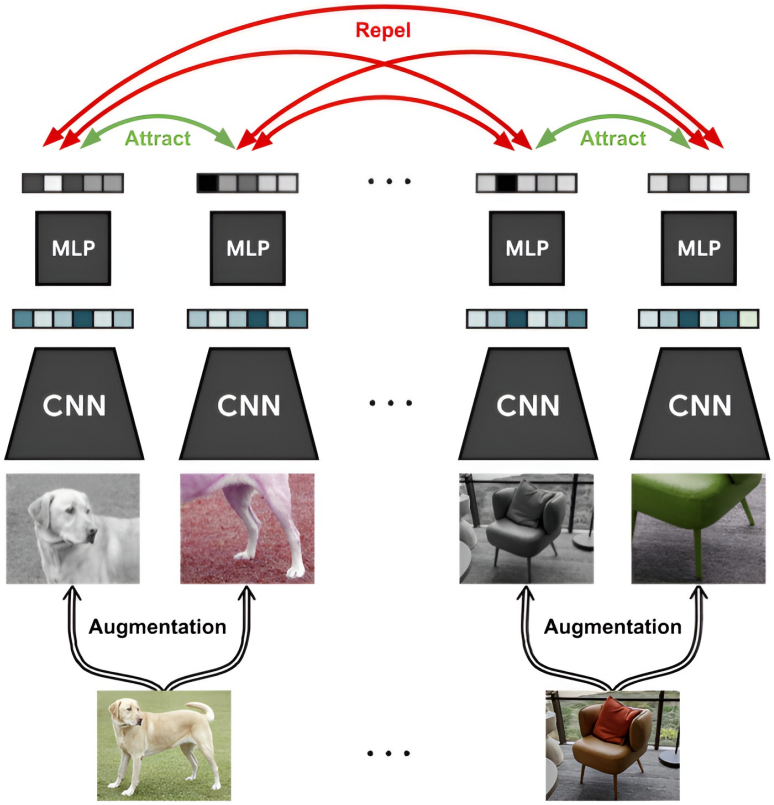


Figure 4: SimCLR framework. Image from Chen et al. [9]

One of the most well-known applications of contrastive learning is the CLIP model [10], introduced by Alec Radford et al. (OpenAI) in 2021. Unlike SimCLR, CLIP uses contrastive learning in a self-supervised setting with image-caption pairs. Its development was motivated by three key observations.

First, traditional computer vision methods relied heavily on manually labeled datasets, which were inflexible. Adding new classes required retraining. In contrast, captions provide a more flexible and descriptive alternative.

Second, training with captions has been shown to be more data-efficient. For example, Desai and Johnson [11] demonstrated that a model trained on captions could be fine-tuned for classification

tasks and still achieve strong performance with less data. Similarly, Zhang et al. [12] showed that learning from medical images and their textual descriptions using contrastive loss produced strong visual representations, even with a limited number of examples.

Third, large-scale NLP models such as GPT-3 [13] and T5 [14] demonstrated that training on large datasets leads to strong generalization. CLIP applied this idea to the vision-language domain.

CLIP was trained on a large dataset of image-caption pairs (Figure 5). The images and captions are encoded independently using separate neural networks: a vision encoder and a text encoder. Contrastive loss is then used to bring matching image-caption pairs closer in the embedding space while pushing non-matching pairs apart.

A key feature of CLIP is its end-to-end training: the image and text encoders are updated jointly from raw input data. This enables the encoders to co-adapt and align their learned features in a shared embedding space. A more detailed explanation of CLIP's contrastive loss function is provided in Section 3.1.

CLIP is especially useful during inference, as it supports both retrieval and zero-shot classification (Figure 5). To classify an image, candidate class names are converted into prompts such as "A picture of [class]". The model then ranks these prompts by similarity to the image embedding and selects the highest-scoring one. Despite the simplicity of this method, it performs surprisingly well and achieved state-of-the-art results at the time of release, outperforming models trained on labeled data.

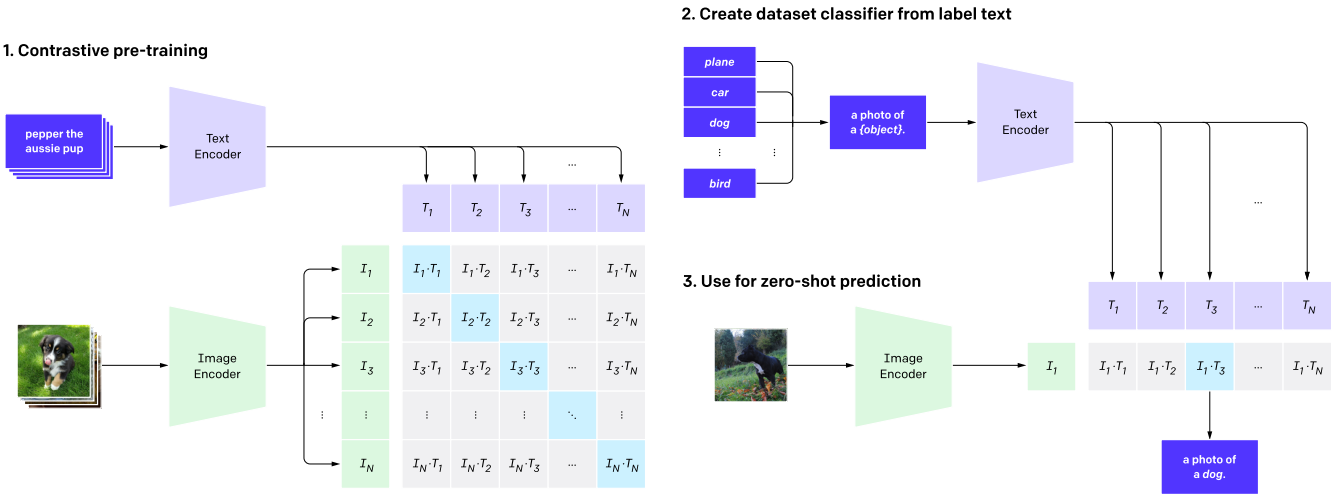


Figure 5: CLIP training (left) and usage (right). Images from Radford et al. [10]

## 2.2 Going to Video

Similarly to image-text retrieval, early video-text retrieval methods relied on pre-extracted video and text features [15, 16, 17]. More recent approaches move toward end-to-end models that jointly train video and text encoders, such as CLIP-BERT [18] and Frozen [19].

Following the success of CLIP [10], several models adapted its ideas to the video domain. One of the first was CLIP4Clip [20], which initialized its encoders using pretrained CLIP weights,

allowing the model to benefit from CLIP's strong image-text alignment. Other examples include VideoCLIP [21], CenterCLIP [22], and CLIP-ViP [23]. Some models introduced fine-grained techniques to better capture video-text relationships. For instance, TS2-Net [24] shifts frame tokens to better model temporal continuity and selects the top-K most informative tokens. X-CLIP [25] computes similarities across multiple levels, including video-caption, video-word, frame-caption, and frame-word. UCoFiA [26] uses a similar coarse-to-fine alignment strategy, though it omits video-word alignment.

Despite these advancements, most models still face significant challenges in capturing fine-grained semantic differences between video and text [1]. This has led to a growing interest in developing improved training strategies, evaluation metrics, and model architectures. In the following subsection, we review recent efforts aimed at addressing these limitations.

## 2.3 Recent Work

Several recent studies have proposed methods to overcome the coarse-grained nature of existing models. The approaches discussed below represent some of the most prominent strategies in the literature.

**Hard Negatives** One direction for addressing these fine-grained limitations is the use of hard negatives during both training and evaluation. Chen et al. [1] propose such an approach, which is described in more detail in Section 3.2. To better assess fine-grained understanding, they also introduce a new metric called PosRank (see Section 4.3).

Similarly, Wufei et al. [27] explore alternative ways to improve fine-grained reasoning. While Chen et al. generate hard negatives by replacing one word with its antonym, Wufei et al. use two main techniques to construct their training data.

The first is *mask filling*, where a word in a sentence is removed and predicted using a language model like BERT [28]. For example, "I [MASK] in the lake" may become "I drown in the lake." The second technique utilizes large language models (LLMs), such as ChatGPT [13], to rephrase the caption. This allows for multiple word substitutions. For instance, "I swim in the lake" could become "I drown in the deep lake." While this increases diversity and realism, it also requires more computational resources.

In both techniques, the introduction of hard negatives leads to a significant improvement in performance on fine-grained evaluation metrics proposed in the respective studies. However, it also causes a slight drop in traditional metrics such as Mean Rank and Recall@k.

**Related Problem in NLP** Similar efforts to improve model understanding were previously made in the field of natural language processing, particularly with the introduction of the Winograd Schema Challenge [29]. This benchmark was designed to evaluate a model's ability to understand cause-and-effect relationships and resolve pronoun references in contexts that require commonsense reasoning, rather than relying on superficial statistical patterns.

To illustrate the challenge, consider the following two sentences from [29]:

- The city councilmen refused the demonstrators a permit because they feared violence.
- The city councilmen refused the demonstrators a permit because they advocated violence.

In both cases, we can ask: "To whom does the pronoun 'they' refer—the city councilmen or the demonstrators?" Solving this requires a deeper understanding of the sentence semantics and context. A model that depends only on learned statistical correlations would struggle to answer this correctly, highlighting the need for genuine reasoning capabilities. The original challenge was later extended by Sakaguchi et al. [30], who introduced a larger and more diverse dataset. They also aimed to reduce biases in the questions, such as those introduced by frequent word co-occurrences in human-written texts and biases embedded in the model's training data.

Building on this idea, Thrush et al. [31] adapted the task to the vision-language domain, resulting in the Winoground benchmark. Their goal was to address a major limitation in vision-language models: the tendency to treat image-text matching as a bag-of-words problem [32]. In such cases, models often match images and captions based on individual word presence, rather than understanding how the words interact to form meaning. This limitation is also one of the core challenges we aim to address in our work. An example from the Winoground dataset is shown in Figure 6.

(a) Some plants surrounding a lightbulb

(b) A lightbulb surrounding some plants

Figure 6: Example from the Winoground dataset. State-of-the-art models struggle to distinguish between the two. Images from Thrush et al. [31].

**More Approaches** In the image-text domain, the inclusion of hard positives—semantically similar but distinct pairs—has also been explored [33]. In this work, we extend that idea to the video-text setting (see Section 5.4). Other recent ideas include hierarchy-aware embeddings [34], though their effectiveness in the video domain remains unclear. Some works have also explored integrating attention mechanisms for hard negatives within the InfoNCE loss [35], or using adaptive margins [36], but these yield only minor improvements. Consequently, the question of how to effectively enhance video-text retrieval remains open.

# 3 Methodology

This section outlines the methodological foundations of our work. We begin by presenting the necessary background knowledge and definitions required to understand the proposed approach (Section 3.1). Subsequently, we detail the loss functions employed in our experiments, including

their formulation and relevance to the tasks at hand (Sections 3.2 and 3.3), and the sampling strategy for hard sentences (Section 3.4).

## 3.1 Definitions

**Similarity** To measure the similarity between two vectors—representing the image and text encodings in the same Euclidean space—we use cosine similarity. We denote this function as $s(\cdot, \cdot)$, and define it as follows:

$$s(v_{\text{img}}, v_{\text{txt}}) = \frac{v_{\text{img}} \cdot v_{\text{txt}}}{|v_{\text{img}}| \cdot |v_{\text{txt}}|} \tag{1}$$

Here, $v_{\text{img}}$ represents the encoded image features, and $v_{\text{txt}}$ represents the encoded text features.

**Contrastive (coarse-grained) loss** The most commonly used objective for training video-text retrieval models is the **contrastive loss** [37], also known as InfoNCE. In this work, we refer to this formulation as the *coarse-grained loss*.

To train a retrieval model, we typically consider a set of videos $V$ and a corresponding set of captions $T$, where $|T| \geq |V|$ since a video may have multiple associated captions. During training, we sample matching video-text pairs $(v_i, t_i)$. The coarse-grained loss is then computed as:

$$\begin{aligned}
\mathcal{L}_{v2t} &= -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp\left(s\left(v_i, t_i\right)\right)}{\sum_{j=1}^{B} \exp\left(s\left(v_i, t_j\right)\right)}, \\
\mathcal{L}_{t2v} &= -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp\left(s\left(v_i, t_i\right)\right)}{\sum_{j=1}^{B} \exp\left(s\left(v_j, t_i\right)\right)}, \\
\mathcal{L}_{\text{coarse}} &= \mathcal{L}_{v2t} + \mathcal{L}_{t2v}
\end{aligned} \tag{2}$$

Here, $B$ denotes the batch size, $v_i$ is the encoded representation of the $i$-th video, and $t_i$ is the encoded representation of the corresponding caption. Each term compares the similarity score $s(v, t)$ of a positive pair (i.e., a matching video and caption) against the similarities of negative pairs within the batch. The loss encourages the model to align matching pairs while distinguishing mismatched ones. Importantly, the batch size directly affects the strength of the contrastive signal: larger batches provide more negative samples, which can lead to more robust and effective learning.

## 3.2 Fine-grained Training and Evaluation (Hard Negatives)

Chen et al. [1] propose a fine-grained training methodology aimed at improving model sensitivity to subtle semantic differences between samples. Their approach introduces two key contributions: a novel evaluation metric, *PosRank* (outlined in Section 4.3), and a training strategy that incorporates *hard negative* samples to enhance contrastive learning.

The central component of their method is the use of hard negatives—samples that are semantically close to the anchor but belong to a different class—to enforce more discriminative representations. This encourages the model to focus on nuanced distinctions rather than relying on coarse differences. The strategy for sampling these hard negatives is described in detail in Section 3.4.

Abstracting away from the sampling specifics, the fine-grained training objective can be formulated as follows:

$$\mathcal{L}_{\text{fine}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp\left(s\left(v_i, t_i\right)\right)}{\sum_{t_{\text{neg}}\in N}\exp\left(s\left(v_i, t_{\text{neg}}\right)\right)}, \tag{3}$$

where $B$ is the batch size, $v_i$ is the encoded representation of the $i$-th video, $t_i$ is the corresponding ground-truth caption, and $N$ denotes the set of hard negative captions for $t_i$. This formulation emphasizes the model's ability to distinguish the correct caption from semantically similar yet incorrect alternatives.

The final training objective combines this fine-grained loss with the standard contrastive loss as follows:

$$\mathcal{L} = \mathcal{L}_{\text{coarse}} + \lambda_{fine}\mathcal{L}_{\text{fine}} \tag{4}$$

where $\lambda_{fine}$ is a weighting factor that balances the contribution of the fine-grained loss, and $\mathcal{L}_{coarse}$ is defined in equation 2.

### 3.3 Hard Positives

In addition to leveraging hard negatives, we further refine the training procedure by incorporating *hard positive* samples. This approach, introduced by Kamath et al. [33], addresses a key limitation of using only hard negatives: the model may learn that any perturbation to a sentence necessarily alters its meaning. By including hard positives—modified captions that preserve the original meaning—we encourage the model to be more discerning, learning to distinguish between superficial changes and those that truly affect semantics. While this idea was originally proposed in the image domain, we adapt and extend it to the video domain. The strategy for generating these hard positive captions follows the same methodology described in Section 3.4.

The key methodological contribution in this section lies in the adaptation of the loss function to accommodate these positive sentences. Our formulation draws inspiration from the work of Doveh et al. [38], who proposed a principled approach to balancing informative positives and negatives within contrastive learning frameworks.

The evaluation protocol remains consistent with that described in Section 3.2, allowing us to isolate the impact of the enhanced loss formulation. We define the extended objective as follows:

$$\mathcal{L}_{\text{hard\_pos}} = -\frac{1}{B}\sum_{i=1}^{B}\frac{1}{|P_i|}\sum_{j=1}^{|P_i|}\log\frac{s(t_j, v_i)}{\sum_{k=1}^{B} s(t_j, v_k)} \tag{5}$$

where $B$ denotes the batch size, and $|P_i|$ represents the number of hard positive samples associated with the text $t_i$. In essence, for each sentence in the batch, we iterate over its corresponding hard positive sentences. For each such pair, we compute the ratio between the similarity of the hard positive to the original sentence and the total similarity of that hard positive to all other sentences in the batch. These ratios are then summed, and the final result is normalized by both the batch size and the number of hard positives. We combine this with other loss values as follows:

$$\mathcal{L} = \mathcal{L}_{\text{coarse}} + \lambda_{fine}\mathcal{L}_{\text{fine}} + \lambda_{hard\_pos}\mathcal{L}_{\text{hard\_pos}} \tag{6}$$

### 3.4 Hard positives/negatives sampling

To generate hard positive and negative samples, we employ two primary methods: *part-of-speech (POS) replacement*, a more controlled and deterministic approach, and *large language model (LLM)-based generation*, which relies on prompting LLMs to either paraphrase a given sentence (producing hard positives) or generate syntactically plausible but semantically incorrect alternatives (producing hard negatives). In both cases, augmented sentences are pre-generated prior to training and subsequently sampled online during training to enhance learning robustness.

**Parts of Speech (POS) Method** This method follows the approach described in [1] and all the notation is duplicated in Appendix A for a quick reference. Let us consider a dataset consisting of a set of videos $V$ and a corresponding set of captions $T$, where $|V| \leq |T|$ (i.e., multiple captions may correspond to a single video). Given a video $v_i$ and its associated set of captions $T_i$, we aim to construct a set of hard negative (or positive) captions $N_{ij}$ (or $P_{ij}$) of size $n$ for each caption $t_j \in T_i$.

To achieve this, we utilize a predefined set $P$ of five parts of speech: *nouns, verbs, adjectives, adverbs, and prepositions*. The overall idea is to make subtle yet meaningful changes to the sentence structure while preserving grammatical structure. These changes are designed to create hard negatives and hard positives that challenge the model's understanding of nuanced semantic differences. The algorithm to generate **hard negatives** is presented in Algorithm 1.

In summary, the replacement word $w'$ is chosen based on the following priority:

1. An antonym of $w$, if available (via WordNet [39]).
2. An antonym of a hypernym or hyponym of $w$ (via WordNet [39]).
3. If neither is available, a randomly selected word from the dataset vocabulary sharing the same POS tag $p$.

This strategy ensures that substitution introduces semantic contrast while preserving the proper grammar. Using antonyms and their semantic relatives increases the likelihood that the altered caption becomes a true negative. Moreover, by modifying only a single word in each caption, the method creates challenging examples that test the model's sensitivity to fine-grained semantic differences, such as "slow" vs. "fast", "forward" vs. "backward", or differences in color, age, and gender.

In cases where no antonyms or related semantic opposites can be identified (i.e., Options 1 and 2 are not applicable), the replacement word $w'$ is randomly selected from a predefined dataset-specific vocabulary. This vocabulary is constructed from the training captions and organized as a dictionary indexed by the five part-of-speech categories (nouns, verbs, adjectives, adverbs, and prepositions). Each entry in the dictionary contains a list of words observed in the training data that correspond to the respective POS tag. This design ensures that all replacement words are familiar to the model, thereby mitigating the risk of introducing out-of-distribution or unfamiliar tokens during training. By constraining substitutions to words that the model has already encountered, we maintain the integrity of the training distribution and avoid artificially increasing the difficulty of the task through exposure to unseen vocabulary.

For constructing **hard positives**, the process is similar and presented in Algorithm 2.

**Algorithm 1** Generate Hard Negatives

**Require:** Sentence $t_j$, set of parts of speech $P$, number of required sentences $n$, dataset vocabulary $V$

**Ensure:** Set of hard negatives $N_{ij}$

Initialize $N_{ij} \leftarrow \emptyset$
Tokenize sentence $t_j$ into a list of words $W$
Get a list of available parts of speech $P_{avail}$ in a sentence $t_j$ from $W$
Initialize $D \leftarrow \emptyset$ {Set of substitutions with direct antonyms}
Initialize $H \leftarrow \emptyset$ {Set of substitutions with antonyms of hypernyms or hyponyms}
**for** $p_{avail}$ in $P_{avail}$ **do**
    Select a random word $w$ with POS-tag $p_{avail}$ from a list $W$
    Get set $D_w$ of substitutions with direct antonyms of $w$
    $D \leftarrow D \cup D_w$
    Get a set $H_w$ of substitutions with antonyms of hypernyms and hyponyms of $w$
    $H \leftarrow H \cup H_w$
**end for**
$N_{ij} \leftarrow N_{ij} \cup D$
**if** $|N_{ij}| < n$ **then**
    $N_{ij} \leftarrow N_{ij} \cup H$
**end if**
**if** $|N_{ij}| < n$ **then**
    **for** $i = |N_{ij}|$ **to** $n$ **do**
        Select random word $w$ from a list $W$
        Get a POS-tag $p_w$ of $w$
        Find a random word $w_{dict}$ from $V$ with the same POS-tag $p_w$
        Make substitution of $w$ for $w_{dict}$, and save it as $s$
        Add $s$ to $N_{ij}$
    **end for**
**end if**
Reduce the $|N_{ij}|$ to $n$, giving the first preference to the substitutions from $D$, second preference to the substitutions from $H$, and the third one for all remaining random substitutions
**return** $N_{ij}$

**Algorithm 2** Generate Hard Positives

**Require:** Sentence $t_j$, set of parts of speech $P$, number of required sentences $n$
**Ensure:** Set of hard positives $P_{ij}$

Initialize $P_{ij} \leftarrow \emptyset$
Tokenize sentence $t_j$ into a list of words $W$
Get a list of available parts of speech $P_{avail}$ in a sentence $t_j$ from $W$
Initialize $D \leftarrow \emptyset$ {Set of substitutions with direct synonyms}
Initialize $H \leftarrow \emptyset$ {Set of substitutions with synonyms of hypernyms or hyponyms}
**for** $p_{avail}$ in $P_{avail}$ **do**
    Select a random word $w$ with POS-tag $p_{avail}$ from a list $W$
    Get set $D_w$ of substitutions with direct synonyms of $w$
    $D \leftarrow D \cup D_w$
    Get a set $H_w$ of substitutions with synonyms of hypernyms and hyponyms of $w$
    $H \leftarrow H \cup H_w$
**end for**
$P_{ij} \leftarrow P_{ij} \cup D$
**if** $|P_{ij}| < n$ **then**
    $P_{ij} \leftarrow P_{ij} \cup H$
**end if**
Reduce the $|P_{ij}|$ to $n$, giving the first preference to the substitutions from $D$, and the second preference to the substitutions from $H$
**return** $P_{ij}$

Therefore, the preference for the substitution word is the following:

1. A synonym of $w$, if available (via WordNet [39]).
2. A synonym of a hypernym or hyponym of $w$ (via WordNet [39]).

The goal here is to create variants that retain the original meaning while introducing surface-level diversity, helping the model learn that these variants still refer to the same video content. This can aid in improving robustness to lexical variation.

**LLM-Based Generation Method** Our approach is inspired by [27], leveraging the paraphrasing capabilities of LLMs to generate both hard positive and hard negative examples. Specifically, we employ Qwen2.5-1.5B-Instruct [40], selected for its favorable balance between performance and model size. Although it is not the most advanced variant of the Qwen family, it provides sufficient capability for producing challenging sentence-level paraphrases.

The generation parameters are configured as follows: a temperature of 0.6, left-side tokenizer padding, and a maximum output length of 30 tokens per paraphrase. This cap was chosen based on the dataset's average sentence length of 14.25 words. By allowing for up to 20 words and assuming an average of 1.5 tokens per word, we ensure coverage of most sentence lengths while avoiding excessive computational overhead. Detailed statistics on sentence length distribution and the specific prompts used are provided in Appendices B and C, respectively.

LLM responses are not always perfectly aligned with the intended paraphrasing task; occasionally, the generated outputs may be unsuitable (fewer sentences than were asked) or duplicated. To address this, we developed an iterative filling-and-filtering pipeline that ensures a complete and diverse set of hard examples. The key idea is to repeatedly invoke the model to generate missing paraphrases for underfilled caption sets, while simultaneously filtering out repeated outputs associated with the same original sentence.

The augmentation process is formalized in Algorithm 3. The function FillUp identifies which original captions lack a sufficient number of paraphrases and re-invokes the LLM to generate the required number of additional examples, using the same prompt (Appendix C) with a different number of sentences needed. The FilterRepetitions function ensures that each generated paraphrase set contains unique entries for a given caption ID by removing duplicates. This iterative process continues until a pass through the loop results in no further changes to the dataset size, indicating convergence.

**Sampling** As mentioned earlier, we generate all sentences prior to training and store them in a separate file. The number of generated sentences varies depending on the method and sentence type, as shown in Table 1.

| | **Positives** | **Negatives** |
|---|---|---|
| **POS** | 100, 20, 5 | 40, 20, 5 |
| **LLM** | 20 | 20 |

Table 1: Number of generated sentences by method and sentence type. For POS-generated sentences, we create three sets with varying numbers of items to support the experiment described in Section 5.6.

**Algorithm 3** Iterative Hard Sentence Augmentation Using LLMs

**Require:** Original caption set $C$, initial generated sentence set $G$
**Ensure:** Augmented set of hard sentences $G$

$G \leftarrow \textsc{FilterRepetitions}(G)$
*finished* $\leftarrow$ **false**
**while** not *finished* **do**
  $G \leftarrow \textsc{FillUp}(G, C)$
  $s_1 \leftarrow \textsc{Size}(G)$
  $G \leftarrow \textsc{FilterRepetitions}(G)$
  $s_2 \leftarrow \textsc{Size}(G)$
  **if** $s_1 - s_2 = 0$ **then**
    *finished* $\leftarrow$ **true**
  **end if**
**end while**
**return** $G$

The sampling strategy reflects the capabilities and constraints of each generation method. For the POS-based method, it is relatively easy to generate a large number of hard negative examples by leveraging a predefined dictionary. However, the number of positive examples that can be generated depends on the specific word in question and varies significantly. Therefore, the numbers shown in the table for the POS method represent the maximum possible number of sentences that can be generated. A detailed distribution of these counts can be found in the Appendix D.

In contrast, for the LLM-based method, we generated exactly 20 sentences for both positive and hard negative examples, due to the peculiarities of the generation algorithm (Algorithm 3).

**Quality** To empirically assess the quality of the generated sentences, we present two illustrative examples in a Table 2.

Table 2: Examples of generated hard positive and negative sentences using different methods. Full table may be found in Appendix E

| Method | Generated Sentence |
|---|---|
| **Original:** | *A couple of people roping a calf in a ring and riding horses.* |
| **POS Hard Negative** | A couple of people roping a calf in a shanty and riding horses. |
| **POS Hard Positive** | A couple of citizenry roping a calf in a ring and riding horses. |
| **LLM Hard Negative** | A party of individuals are attempting to herd an animal using only their legs. |
| **LLM Hard Positive** | Two folks encircling a bull with ropes as they gallop their mounts. |

Continued on next page

Table 2 – continued from previous page

| Method | Generated Sentence |
|---|---|
| **Original:** *a man is showing how a square knot looks like with a white string.* | |
| **POS Hard Negative** | a man miss showing how a square knot looks like with a white string. |
| **POS Hard Positive** | a man is picture how a square knot looks like with a white string. |
| **LLM Hard Negative** | A man tries to teach others how to perform a square knot using blue rope. |
| **LLM Hard Positive** | A person clarifies the elements of a square shackle employing a soft beige thread. |

The POS-based methodology tends to produce sentences that closely resemble the original ones, but it is prone to grammatical or logical errors. In contrast, the LLM-based approach, while generally more diverse, is not without its limitations. For instance, in the case of hard negatives, it may produce sentences that deviate too much from the original, making them insufficiently challenging as negative samples. For hard positives, the LLM can sometimes introduce content not present in the original sentence—so-called hallucinations—which may significantly alter the intended meaning. This is exemplified in the second example in Table 2. However, in that specific instance, the original caption is arguably ambiguous, allowing for multiple valid interpretations, including the generated one.

In a manual evaluation of 25 additional sentences, we identified only two instances where the LLM-generated hard positive captions introduced hallucinated content. One such case is the modification: "A young girl gives a wink and then waves her hand goodbye" → "She lifts her hand to wave goodbye, making sure everyone knows that the party is coming to an end," where the LLM adds a new narrative element not present in the original sentence. A more significant example is: "Two amazing catches by high school football quarterbacks and receivers" → "Two incredible throws thrown by elite high school quarterbacks and wideouts," in which the term "elite" is introduced without justification, and the semantic shift from "catches" to "throws" alters the meaning to the extent that the generated sentence is arguably more appropriate as a hard negative than a positive.

An additional characteristic of the LLM-based generation is its ability to significantly expand the dataset's vocabulary. Nevertheless, as most new terms are synonymous with those in the original captions and correspond to visual elements in the videos, they do not pose a problem for the training process.

**Distribution** Following Algorithms 1 and 2, we obtain sets of hard negatives and positives, whose part-of-speech (POS) distributions are shown in Figure 7.

The distributions of parts of speech used to generate hard negatives and positives closely resemble the appearance of these parts of speech in the original dataset. This alignment holds especially for larger set sizes. However, for smaller sets, more noticeable deviations appear (see Appendix F). This

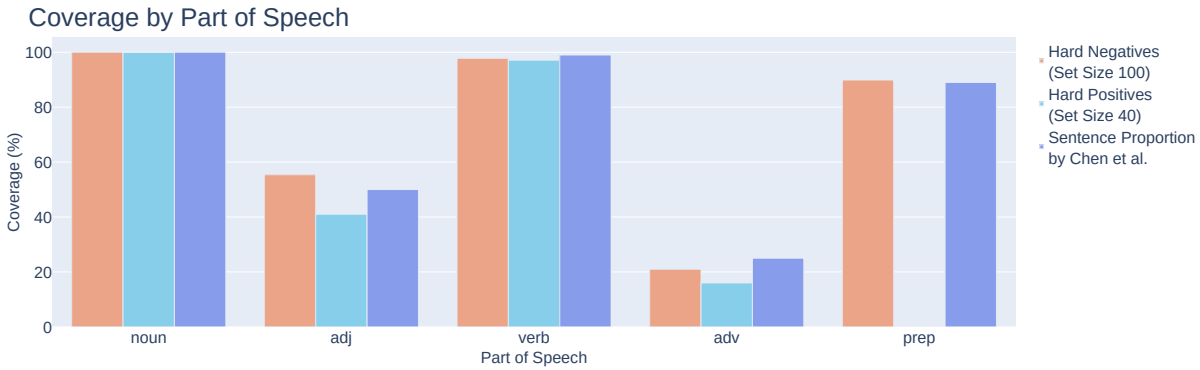


Figure 7: Distribution of parts of speech across generated sentences. For the first two legend groups (hard negatives and positives), the coverage value indicates the percentage of sentences in which a specific part of speech was substituted. For the final legend group ("Sentence Proportion by Chen et al."), the value represents the proportion of sentences in the VATEX [41] dataset containing the corresponding part of speech, as reported by Chen et al. [1]. The plots for smaller set sizes may be found in the Appendix F

is primarily because we prioritize direct antonyms or synonyms and, in cases where fewer sentences are needed, rely less on the dataset vocabulary. As a result, while nouns and verbs are frequently substituted, parts of speech like prepositions are often underrepresented due to the difficulty of finding suitable synonyms or antonyms.

In particular, we observe that no hard positive substitutions involving prepositions were generated. This is due to the lack of available synonyms for prepositions and the fact that our hard positive generation process (Algorithm 2) does not incorporate vocabulary from the dataset. While the absence of synonyms does not necessarily imply the absence of antonyms, most hard negatives involving prepositions were likely generated using the dataset vocabulary.

# 4 Data, Model, and Evaluation

## 4.1 The Dataset

In this work, we utilize the **VATEX** dataset [41], which contains over 41,250 videos and 825,000 captions in both English and Chinese. VATEX was selected due to several key advantages over other commonly used datasets, such as YouCook2 [42], DiDeMo [43], ActivityNet [44], MSR-VTT [2], MSVD [45], and TGIF [46]. Notably, VATEX contains the highest proportion of captions with adjectives, adverbs, and prepositions [1], making it particularly challenging and informative for evaluating a model's understanding of parts of speech—an aspect central to the PosRank (see Section 4.3) metric.

In addition to its linguistic richness, VATEX is also the largest dataset among those mentioned in terms of the number of captions, which is particularly important given our focus on assessing the effectiveness of hard negative and positive caption augmentations. Furthermore, the dataset employs the Recall@k and Mean Rank metrics for evaluation, which are consistent with the methodology used in our experiments.

VATEX builds upon the Kinetics-600 dataset [47], offering more diverse and descriptive annota-

tions. Compared to the widely used MSR-VTT [2], it provides longer, multilingual captions and a larger volume of data. For the purposes of this study, we restrict our use to the English captions.

It is worth noting that the English captions were collected via Amazon Mechanical Turk, with contributions limited to workers from English-speaking countries, specifically Australia, Canada, Ireland, New Zealand, the United Kingdom, and the United States.

Since the videos are hosted on YouTube, the dataset authors are not permitted to redistribute the video files. As a result, we had to download the videos ourselves. However, not all videos were still available—some had been deleted or made private—so we were only able to partially reconstruct the dataset (see Table 3).

| Split | Collected | Original | Percentage Collected |
|---|---|---|---|
| Train | 23,778 | 25,991 | 91.5% |
| Val | 2,375 | 3,000 | 79.2% |
| Test | 3,391 | 6,000 | 56.5% |
| Total | 29,544 | 34,991 | 84.4% |

Table 3: VATEX download statistics

Although we report statistics for the test split, we do not use it in our experiments to remain consistent with the approach of Chen et al. [1], who used the validation set for evaluation due to the unavailability of the test set at the time.

## 4.2 The Model

In all our experiments, we employ the **X-CLIP** model proposed by Yiwei et al. [25]. The choice of X-CLIP is motivated by several key factors. Firstly, to maintain consistency with prior work by Chen et al. [1], who evaluated multiple models including UCoFiA [26], Frozen [19], X-CLIP [25], and TS2-Net [24], we selected X-CLIP for its demonstrated susceptibility to subtle semantic variations. This sensitivity is crucial when working with hard positive and negative samples, as X-CLIP is explicitly trained to capture localized semantic cues rather than relying solely on global video-text matching. Consequently, it is more capable of reflecting nuanced changes in fine-grained retrieval metrics. Moreover, X-CLIP consistently outperforms many baseline methods on standard benchmarks such as MSR-VTT [2], ActivityNet [44], and DiDeMo [43], providing a robust and reliable foundation for our analysis. This strong performance ensures that observed effects in our experiments can be attributed to the augmentation strategies rather than baseline model weaknesses.

X-CLIP builds on CLIP4Clip [20], which itself is based on the CLIP model [10] (more about it in Section 2.2). While CLIP4Clip aligns full video clips with full text descriptions (video-sentence alignment), X-CLIP improves this by introducing cross-grained contrastive learning (see Figure 8). Instead of only comparing video-sentence pairs, it also learns to align finer-grained elements, such as individual video frames with words. This is achieved through a component called Attention Over Similarity Matrix, which allows the model to focus on relevant frame-word interactions.

Overall, X-CLIP is trained to optimize alignment at four different levels: (1) video–sentence, (2) video–word, (3) frame–sentence, and (4) frame–word. This multi-level alignment helps the model learn a richer and more detailed understanding of the video-text relationship. A simplified diagram of the model architecture is provided in Figure 9.

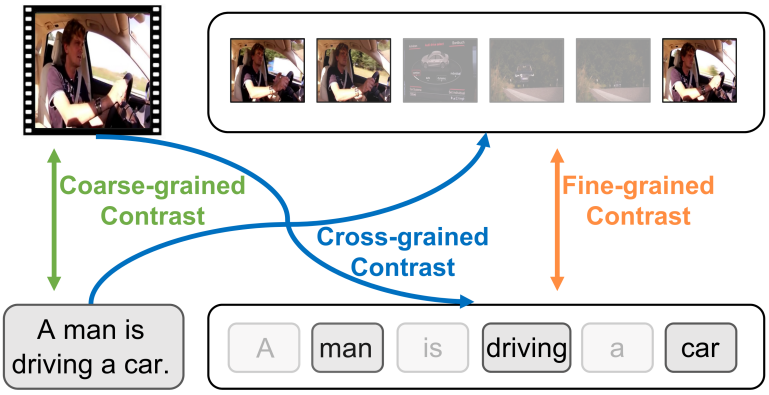


Figure 8: Illustration of the concept of cross-grained contrast used in X-CLIP model. Image is taken from Yiwei et al. [25].

## 4.3 Evaluation Metrics

To evaluate the performance of our retrieval system, we employ four key measures: Recall@k, Mean Rank, PosRank [1], and Brittleness [33]. Recall@k and Mean Rank are widely used in the video-text retrieval literature and serve as standard benchmarks. However, our primary focus is on PosRank and Brittleness, which are better suited for assessing fine-grained retrieval performance. These allow us to evaluate the model's ability to distinguish subtle semantic differences between videos and captions. While Recall@k and Mean Rank are still reported for completeness, they are treated as secondary indicators. Our objective is to enhance fine-grained retrieval quality while maintaining—or potentially improving—performance on conventional benchmarks.

**Recall@k** Recall@k measures the proportion of relevant items that appear in the top-$k$ results. It is defined as:

$$\text{Recall@}k = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\text{rank}(y_i) \leq k\right) \tag{7}$$

where $N$ is the number of queries, $y_i$ is the ground-truth item for the $i$-th query, and $\text{rank}(y_i)$ denotes the rank of the correct item in the list of retrieved results. The indicator function $\mathbb{1}(\cdot)$ returns 1 if the condition is true and 0 otherwise.

**Mean Rank** Mean Rank computes the average rank position of the ground-truth item over all queries:

$$\text{Mean Rank} = \frac{1}{N}\sum_{i=1}^{N} \text{rank}(y_i) \tag{8}$$

Lower values of Mean Rank indicate better retrieval performance, as they imply that relevant items are retrieved closer to the top of the list.

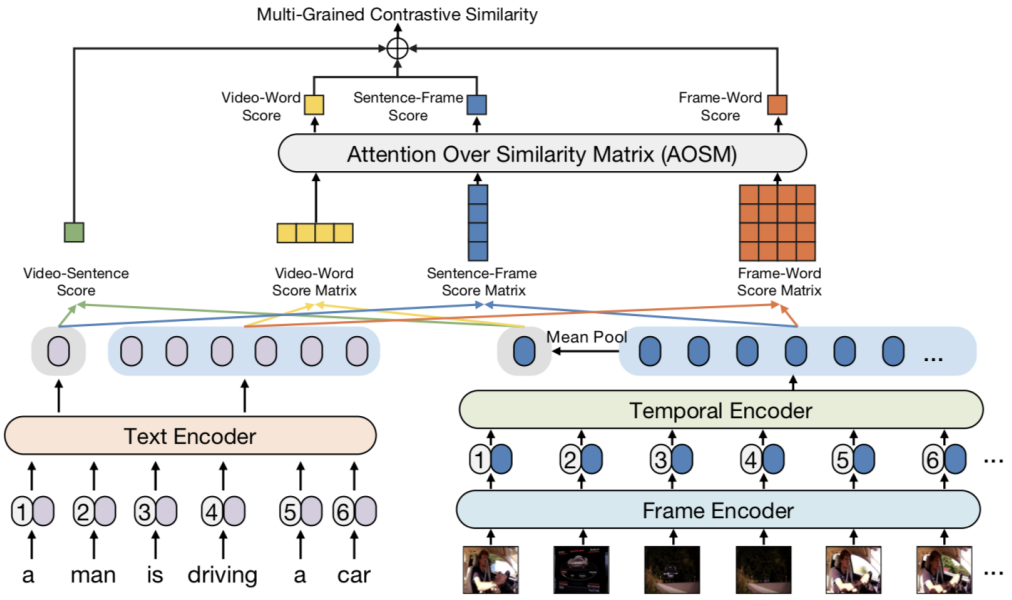


Figure 9: Illustration of the X-CLIP model, taken from Yiwei et al. [25]. The input text is tokenized with positional encodings and passed through a text encoder to produce both coarse-grained (single-token) and fine-grained (multi-token) features. The input video is first divided into frames, which are processed through a frame encoder. Each frame becomes a token, also augmented with positional information. These frame tokens are passed into a temporal encoder to generate fine-grained video features. The coarse-grained video representation is then computed by averaging these fine-grained outputs. Finally, the model computes the four types of similarities between video and text representations.

**PosRank** PosRank quantifies a model's ability to distinguish the ground-truth caption from synthetically generated hard negatives, conditioned on a specific part of speech (POS). This metric is closely aligned with the hard negative generation methodology introduced in Section 3.4, and we adopt the same notation here (see Appendix A for reference).

Given the test set, which, in this work, corresponds to the validation set $D_{\text{val}}$ (used for this purpose due to the reasons outlined in Section 4), we generate a set of $n$ hard negatives $N_{ij}$ for each reference caption $t_j$ corresponding to a video $v_i$. Each hard negative in $N_{ij}$ is created by altering a single word in $t_j$, targeting one of five specified parts of speech in a set $P$={nouns, verbs, adverbs, adjectives, prepositions}. Because the part of speech to be modified is selected uniformly at random, the distribution of POS types among the generated perturbations in $N_{ij}$ is expected to be approximately uniform across the set.

We define $N_{ij}^p \subseteq N_{ij}$ as the subset of hard negatives in which the altered word belongs to POS $p \in P$. We then construct the evaluation set

$$Q = N_{ij}^p \cup \{t_j\},$$

which includes the original caption and its associated hard negatives modified on POS $p$.

By computing the similarity between each sentence in $Q$ and the corresponding video $v_i$, we obtain a ranking of all candidate captions in $Q$. Let $r_{ij}^p$ denote the rank of the original caption $t_j$ within this list (lower rank indicates higher similarity).

The PosRank for part of speech $p$ is then defined as:

$$\text{PosRank}^p = \frac{1}{|D_{val}|} \sum_{i=1}^{|D_{val}|} \sum_{j=1}^{|T_i|} \frac{1}{r_{ij}^p} \tag{9}$$

Recall that, ideally, $D_{\text{val}}$ would be replaced by a dedicated test set $D_{\text{test}}$ under standard evaluation protocols.

**Brittleness** Brittleness measures a model's susceptibility to confusion between semantically similar (hard positive) and misleading (hard negative) captions, and was introduced by Kamath et al. [33]. Unlike PosRank, which evaluates performance on hard negatives alone, Brittleness accounts for both hard positives and hard negatives, thus providing a more holistic view of the model's discriminative capacity.

Given the test set (although we use $D_{\text{val}}$ as a stand-in for a test set), we generate for each reference caption $t_j \in T_i$ (associated with video $v_i$) a set of $n$ hard negatives $N_{ij}$ and a matching set of $m = n$ hard positives $P_{ij}$. Brittleness is then computed as the proportion of evaluation instances where the model ranks the hard negative between the original and hard positive captions. Formally:

$$\text{Brittleness} = \frac{1}{|D_{val}||T_i||N_{ij}|} \sum_{i=1}^{|D_{val}|} \sum_{j=1}^{|T_i|} \sum_{n=1}^{|N_{ij}|} \mathbb{1} \left[ \begin{array}{l} s(t_j, v_i) > s(t_n^{neg}, v_i) > s(t_n^{pos}, v_i) \\ s(t_n^{pos}, v_i) > s(t_n^{neg}, v_i) > s(t_j, v_i) \end{array} \right. \tag{10}$$

Here, the symbol [ denotes the logical "or", and $t_n^{\text{neg}} \in N_{ij}$ and $t_n^{\text{pos}} \in P_{ij}$ represent a hard negative and a hard positive sample, respectively. The indicator function $\mathbb{1}(\cdot)]$ returns 1 when either condition is satisfied, signifying that the hard negative is incorrectly ranked between the reference and the hard positive. See Appendix A for all other notation. The final *Brittleness* score quantifies the proportion of such ambiguous rankings across the dataset. A lower value (↓) indicates greater robustness, suggesting that the model more consistently assigns higher similarity scores to positive captions over negative ones.

In all experiments, we set $n = m = 1$, generating one hard negative and one hard positive per reference caption. Both are constructed using the POS-based methodology described in Section 3.4. By limiting the number of generated sentences, we ensure high-quality and controlled perturbations.

# 5 Results

In this section, we evaluate various training augmentation strategies for video-text retrieval models, focusing on the following research questions: (i) How does the number of hard negatives included during training impact the model's final performance and accuracy? (ii) What is the effect of including hard positives on performance and accuracy? (iii) How does the method used to generate hard sentences influence the model's outcomes? (iv) How does the proportion of hard positives to negatives and the size of the hard sentence pool affect performance and accuracy? These questions are addressed in Sections 5.3 through 5.6. Sections 5.1 and 5.2 provide the justification and details of the experimental setup.

## 5.1 Replication of Original Study

We begin by replicating the original study conducted by Chen et al. [1]. Unlike their work, we use a single dataset—VATEX (see Section 4.1)—and a single model—X-CLIP (see Section 4.2). The training configuration is aligned with that described in the original publication: the model is optimized using Adam, trained for 5 epochs with a batch size of 64. The initial learning rate for the visual and text encoders is set to $1 \times 10^{-7}$, while the remaining modules use an initial learning rate of $1 \times 10^{-4}$. The results from the original study and our replication are presented in Figure 10.

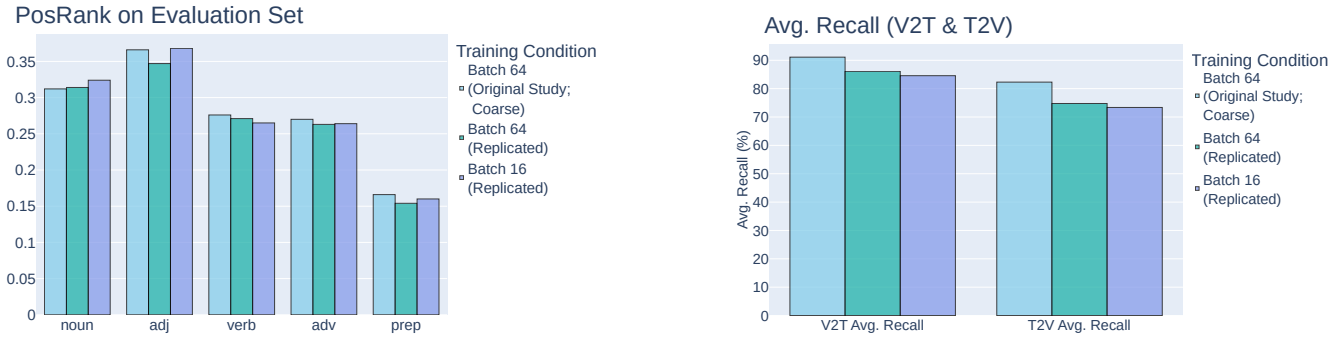


Figure 10: Results for PosRank (left) and Recall (right) evaluations for the original and replicated models. Avg. Recall refers to the average of Recall@1, Recall@5, and Recall@10. For comparison, a model trained with a batch size of 16 is also included to justify its use in subsequent experiments. V2T and T2V denote vision-to-text and text-to-vision retrieval tasks, respectively.

As shown in the plots, the highest PosRank scores are observed for nouns and adjectives, while the lowest are recorded for prepositions. This can likely be attributed to two factors: (i) prepositions require a deeper understanding of spatial and temporal relationships across objects and frames, in contrast to nouns and adjectives, which can often be inferred from a single object in a single frame; and (ii) the dataset contains fewer sentences involving prepositions than those containing adjectives or verbs (see Section 3.4).

As for the recall values, there is a slight decrease when comparing the original and replicated models. This is likely because the replicated model was trained on only 91.5% of the training dataset, as we did not manage to collect the full body of data (see Section 4.1).

Regarding the comparison between models, the performance differences are negligible. Additionally, we include a third model trained with a reduced batch size of 16. This configuration is less demanding on GPU memory, making it suitable for less powerful hardware. Given the minimal impact on performance, we adopt a batch size of 16 for all subsequent experiments.

## 5.2 Partial Data Usage

Following the previous experiment, we now examine the performance trends observed when transitioning from standard training to training that includes hard negatives, in both the original and replicated settings. For the replicated models, we use the same configuration as described in Section 5.1, including a reduced batch size of 16. The results are presented in Figure 11.

As shown in the plots, we observe a consistent trend between the original and replicated experiments, even though our setup differs in three key ways: (i) the batch size is reduced from

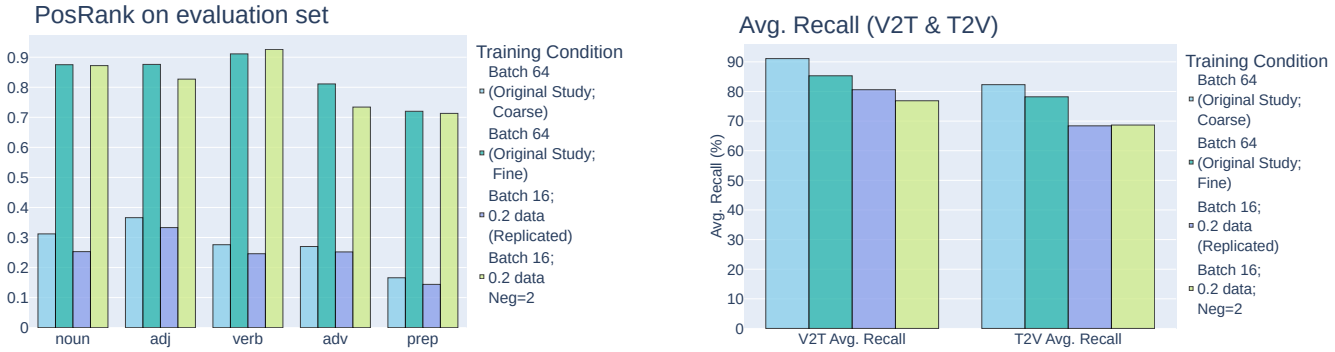


Figure 11: Results for PosRank (left) and Recall (right) evaluations for the original and replicated models. Avg. Recall represents the average of Recall@1, Recall@5, and Recall@10. The replicated models are trained with a reduced batch size of 16 and use only 20% of the dataset, compared to the original models, which use a batch size of 64. Despite these differences, the observed trend—namely, performance improvement with the inclusion of hard negatives—remains consistent across both settings. V2T and T2V refer to vision-to-text and text-to-vision retrieval tasks, respectively.

64 to 16, (ii) the number of hard negatives per caption is decreased from 16 to 2, and (iii) only 20% of the full dataset is used for training. These modifications significantly reduce computational requirements without notably impacting the observed trends. Given the resource efficiency and consistency of results, we adopt this reduced configuration for all subsequent experiments.

## 5.3 Hard Negatives

Having established a reliable experimental setup, we now examine the impact of incorporating hard negatives on video-text retrieval performance. Using the same configuration as described in Section 5.1, we assess the effectiveness of training with varying numbers of hard negatives. The loss formulation is provided in Section 3.2. The results are summarized in Figure 12.

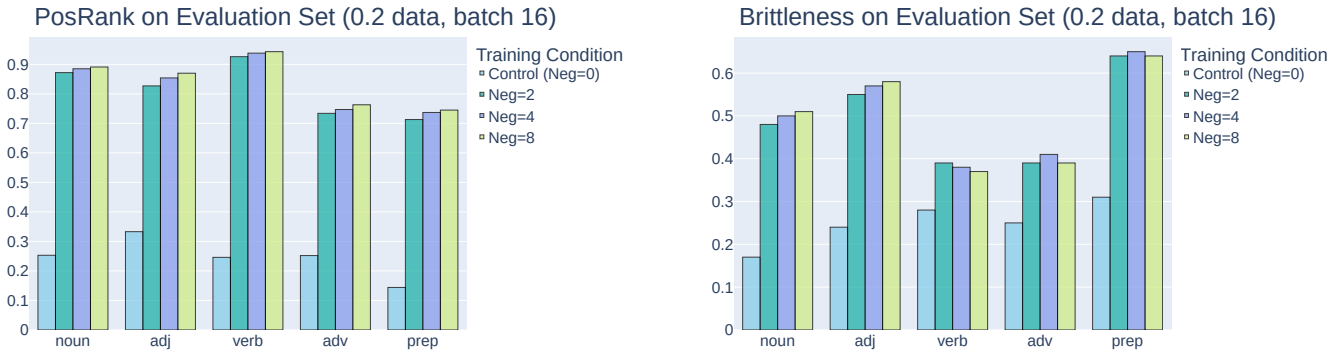


Figure 12: Evaluation results for PosRank (left) and Brittleness (right) across different quantities of hard negatives used during training. Avg. Recall denotes the average of Recall@1, Recall@5, and Recall@10.

From the PosRank plot, we observe a consistent improvement in performance with the inclusion of additional hard negatives. However, the magnitude of this improvement is modest, with a

maximum gain of approximately 0.04.

In contrast, the Brittleness metrics do not exhibit a uniform trend. For example, Brittleness increases with more hard negatives in the case of nouns and adjectives, while it decreases for verbs. Adverbs and prepositions display non-monotonic behavior, with a peak in Brittleness occurring at *Neg=4*.

These fluctuations in Brittleness for prepositions and adverbs may be attributed to the nature of the hard negatives generated for these categories. Specifically, adverb substitutions yield the fewest hard negatives, while preposition-based augmentations rely heavily on existing dataset vocabulary, resulting in less informative negatives (see Section 3.4). Furthermore, the augmentation of verbs appears to be the most effective: it leads to the largest improvement in PosRank and the smallest increase in Brittleness, which even decreases as the size of the negative set grows.

Interestingly, prepositions—and to a lesser extent, adjectives—show the highest Brittleness scores without any clear gains in PosRank. While we do not currently have a definitive explanation for this pattern, it appears consistently across all later experiments (see Sections 5.4, 5.5, and 5.6). This makes it a useful direction for future work.

The Recall@k and Mean Rank metrics exhibit minor fluctuations but do not show substantial changes across configurations. These plots are provided in Appendix G.

In summary, the addition of hard negatives appears to enhance sentence understanding, as reflected in the PosRank and recall-based metrics. However, increasing the number of hard negatives beyond 2 yields diminishing returns and may exacerbate Brittleness for specific parts of speech, potentially indicating overfitting.

It is also worth noting that prior work by Chen et al. [1] identified 16 hard negatives as optimal, though on a different dataset—VLN-UVO [48]. Since our experiments utilize the VATEX dataset [41], which differs in structure, vocabulary, sentence length, and complexity, the optimal number of hard negatives may be dataset-dependent.

## 5.4 Hard Positives and LLM-Generated Sentences

As we have seen, using a small number of hard negatives leads to the best balanced performance. In this section, we explore a different approach by adding hard positives to the training process. The modified loss function used for this setup is described in Section 3.3. In addition, we experiment with generating both hard negatives and hard positives using LLMs (see Section 3.4), instead of relying on POS-generated sentences (see Section 3.4). Based on the results from previous experiments, we limit the number of hard negatives or positives to fewer than 2. The results of these experiments are shown in Figures 13 and 14.

Looking at the PosRank results, POS-generated sentences clearly perform better. This may be because the PosRank evaluation itself is based on POS-generated sentences, which could lead to a form of overfitting.

When we include hard positives, we do not observe significant changes—values remain close to each other. However, for LLM-generated sentences, adding hard positives often leads to slight improvements in performance compared to training with only negatives.

Additionally, for the LLM method, PosRank scores for both negative-only and combined augmentations show a decreasing trend from noun to preposition. The same trend is observed for the POS method, but starting with the verb. Recall that the PosRank metric evaluates how well the model can distinguish between an original sentence and its hard negative variations (see

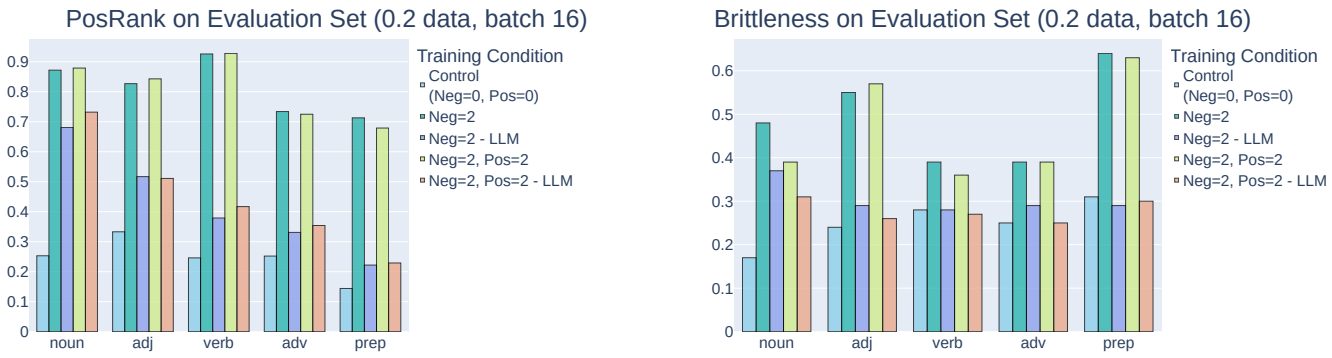


Figure 13: Results for PosRank (left) and Brittleness (right) evaluations for the inclusion or exclusion of hard positives and LLM-generated sentences.

Section 4.3). This suggests that models trained on POS-generated sentences are best at detecting changes to nouns, while those trained on LLM-generated sentences are better at detecting changes to verbs. In both methods, adjectives are the second most distinguishable part of speech.

We propose the following explanation. For POS-generated sentences, the PosRank values closely mirror the distribution of parts of speech within the hard positives and negatives, with the notable exception of prepositions. The relatively low PosRank for prepositions may be attributed to the reduced quality of those sentence variations (see Section 3.4). In contrast, for LLM-generated sentences, we lack direct information about the distribution of parts of speech. However, it is reasonable to assume that this distribution approximates that of the underlying dataset, and this assumption may account for the observed patterns.

For the Brittleness metric, POS-generated training leads to higher scores, just like in the PosRank results. However, since we want to minimize Brittleness, higher values are undesirable. In fact, POS-generated sentences lead to the highest Brittleness, particularly for prepositions and adjectives in POS methodology—the same picture as we saw in Section 5.3.

In contrast, LLM-generated sentences result in more consistent performance across parts of speech. This may be because LLMs produce a more even distribution of grammatically correct sentences across parts of speech.

When we look at the effect of adding hard positives on Brittleness, we see a consistent but small reduction in Brittleness for training with LLM-generated sentences, except for prepositions. The values for prepositions are very close, and the small differences in the plot are likely due to rounding. For the POS method, Brittleness also decreases in most cases, except for adjectives. This may be because adjective changes often introduce grammar errors, as mentioned earlier. However, more data is needed to draw strong conclusions.

Figure 14 shows that the configuration *Neg=2, Pos=2* leads to the most notable deviation from the baseline. While this setup achieves the highest performance on the T2V task—an expected outcome given the use of auxiliary hard positives and negatives—it simultaneously shows reduced performance on the V2T task. A similar pattern is observed in Recall@1, where the changes follow the same direction but with a smaller magnitude.

Among all configurations, the *Neg=2, Pos=2 - LLM* setup yields the most balanced results. It maintains a high T2V Mean Rank while only slightly compromising V2T performance. This pattern is consistent with the trends observed for the PosRank and Brittleness metrics, where

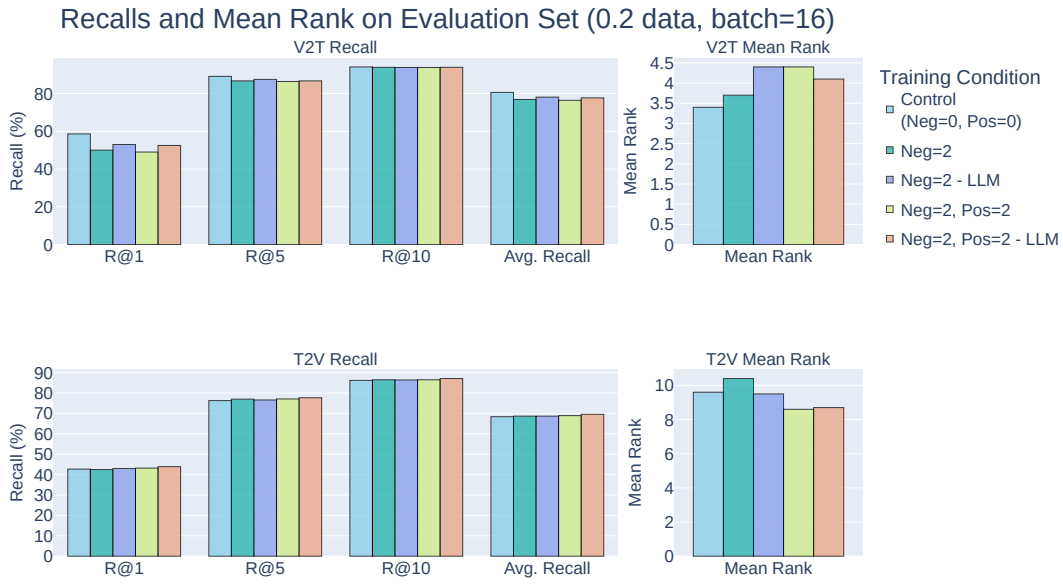


Figure 14: Recall evaluation results for the exclusion or inclusion of hard positives and LLM-generated sentences. Avg. Recall is the average of Recall@1, Recall@5, and Recall@10. V2T and T2V refer to vision-to-text and text-to-vision retrieval tasks, respectively.

LLM-generated augmentations result in less pronounced but more reliable improvements.

In conclusion, training with POS-generated sentences yields higher PosRank scores but also increases Brittleness, particularly for parts of speech with limited lexical alternatives or frequent grammatical inconsistencies. In contrast, LLM-generated sentences offer more consistent performance across all metrics, likely due to their grammatical fluency and broader linguistic coverage. Furthermore, increasing the number of hard positives and negatives generally enhances T2V retrieval but tends to degrade V2T retrieval.

## 5.5 Effect of the Proportion of Hard Positives to Negatives

As shown in the previous experiment, including hard positives during training slightly improved model performance. In this section, we explore how different proportions of hard positives to hard negatives affect this improvement. We also study how performance changes when using only hard negatives or only hard positives. This acts as an ablation study on the effect of each type of hard example. The results are presented in Figures 15, 16, and 17.

Looking at Figure 15 and focusing on PosRank, we observe a small improvement when adding hard positives and a larger improvement when adding hard negatives. When both are added, the performance does not increase further and may even decrease slightly. This could be because PosRank evaluation relies on hard negative-like examples, so training with only them leads to overfitting for this specific task.

Interestingly, using both hard negatives and positives results in lower PosRank compared to using hard negatives alone, but it leads to better Brittleness scores, especially for nouns and verbs. In Figure 16, the configuration *Neg=2, Pos=2* achieves the best Mean Rank for text-to-video (T2V) retrieval but performs the worst on video-to-text (V2T) retrieval. Training with hard positives only

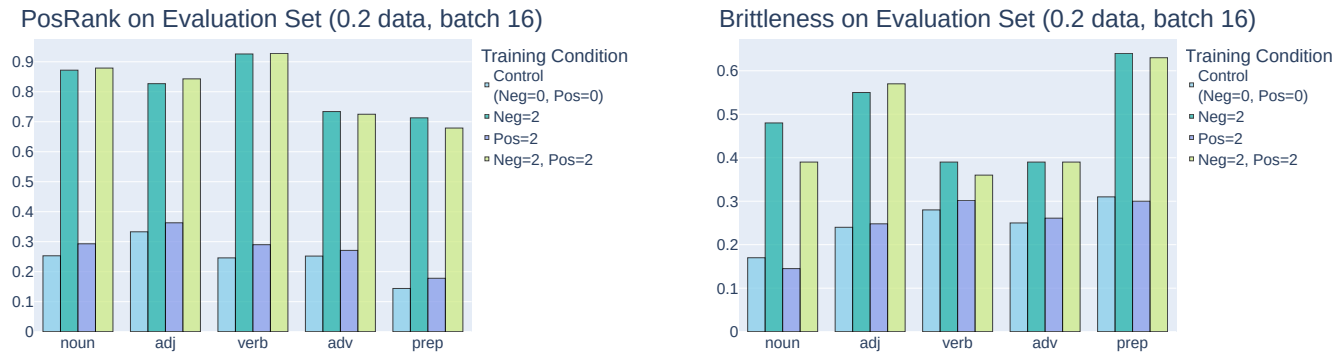


Figure 15: Results for PosRank (left) and Brittleness (right) in the ablation study of including hard negatives and hard positives in training.

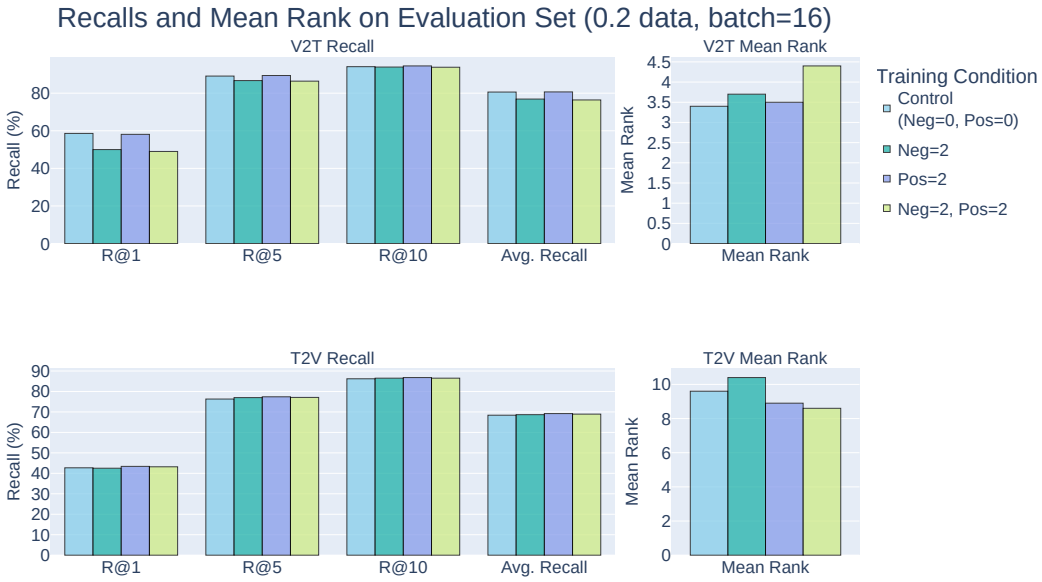


Figure 16: Recall results for the ablation study of including hard negatives and hard positives in training. Avg. Recall is the average of Recall@1, Recall@5, and Recall@10.

achieves the second-best results for both T2V and V2T retrievals.

Therefore, if the goal is a balanced and consistent improvement, using hard positives only is preferable. If the aim is to boost T2V performance, even at the cost of V2T performance, then training with both hard negatives and positives is more effective. This aligns with the findings discussed in Section 5.4.

Next, we turn to the effect of different proportions, as shown in Figure 17. Training with 2 hard negatives results in a significant PosRank improvement, but also increases Brittleness. When training with 1 hard negative and 2 hard positives, PosRank improves while Brittleness remains relatively unchanged. Full Recall@k and Mean Rank results are reported in Appendix G.

These results support the earlier conclusion from Section 5.4 that hard positives bring more modest improvements compared to hard negatives but also help reduce Brittleness. Notably, the biggest performance jump happens when 2 hard negatives are used. Adding more after that brings

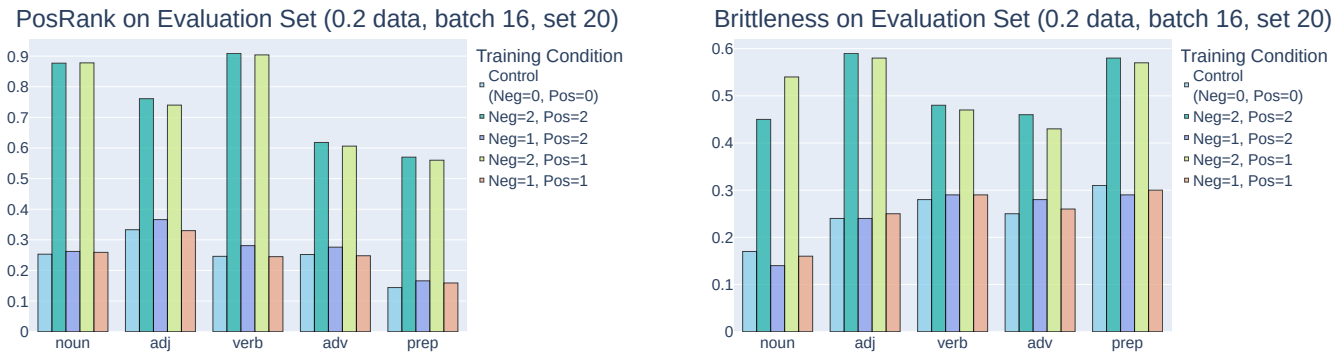


Figure 17: Results for PosRank (left) and Brittleness (right) when varying the proportion of hard positives and hard negatives during training.

only small gains (see Section 5.3). It is also possible that 2 hard positives help "balance" 1 hard negative in the *Neg=1, Pos=2* setup, but this effect is unlikely.

## 5.6 Effect of the Set Size

So far, we have seen that training with LLM-generated captions leads to small but consistent improvements. However, this might be due to the higher quality of the generated captions, rather than the generation method itself. To better understand this, we conducted an experiment to test how the size of the set of generated hard positives and negatives using POS methodology affects the model's performance. Specifically, we wanted to know if using smaller sets, containing only the highest-quality captions (see Section 3.4), could lead to better results.

In previous experiments, the set size for POS-generated samples was 100 for negatives and 40 for positives. In contrast, LLM-generated sets had a fixed size of only 20. This difference in set size may have influenced the results, so we tested different sizes to see how they affect performance. The results of this experiment are shown in Figure 18.

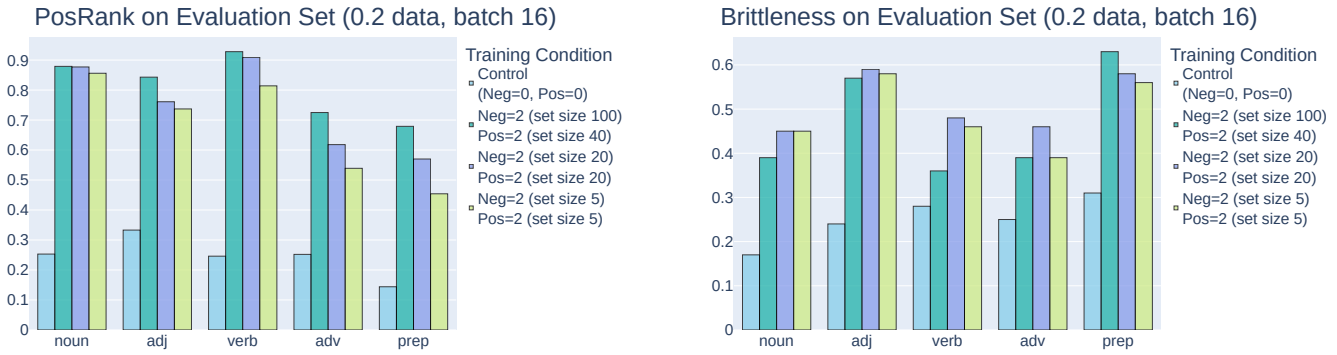


Figure 18: Results for PosRank (left) and Brittleness (right) evaluations showing the effect of set size, created using POS methodology, and used to sample hard positives and negatives, on model performance.

As we can see, reducing the set size generally leads to worse performance. PosRank consistently decreases as the set size becomes smaller. For Brittleness, we observe more variation. The best

(lowest) Brittleness scores are observed when the set size is 20 for most parts of speech, except for prepositions.

The exception observed for prepositions is likely attributable to the limited availability of high-quality substitution candidates. As discussed in Section 3.4, hard negatives for prepositions are primarily generated from the dataset vocabulary—the least optimal strategy—while no hard positives are created, due to the absence of synonymous prepositions in the dataset. Consequently, the brittleness score for prepositions is largely influenced by substitutions involving other parts of speech. Reducing the set size diminishes this influence, which may explain the observed deviation in performance trends.

For other parts of speech, however, the worst Brittleness scores occur with the middle-sized set, i.e., 20. One possible explanation for this trend may lie in the methodology used for generating captions (see Section 3.4). In our current approach, we prioritize substitutions as follows: first, we select direct synonyms or antonyms; second, we consider synonyms or antonyms of hypernyms or hyponyms; and finally, we use a dataset-specific dictionary, in case of hard negatives.

This ordering might lead to suboptimal results, particularly when the set size is 20, which may include only examples derived from the first and second preference levels. Including the second-tier substitutions—those derived from hypernyms or hyponyms—might introduce confusion or noise into the training data, thereby affecting model performance.

In contrast, smaller set sizes, such as 5, which rely mostly on direct synonyms or antonyms, or larger set sizes that incorporate a wider variety of substitutions (especially for hard negatives), might provide more robust training signals. However, this remains a hypothesis, and additional data and analysis are required to validate this assumption.

The corresponding Recall@k and Mean Rank values can be found in Appendix G.

# 6 Conclusion and Future Work

In this work, we studied how adding hard positive examples affects video-text retrieval performance. We also explored different ways to generate hard positive and negative sentences. Our main findings are the following: first, adding hard positives leads to more consistent improvements than adding hard negatives when used separately. Second, generation methods based on large language models perform better than rule-based methods for both hard positives and negatives. Third, in text-to-video retrieval tasks, using both hard positives and negatives gives the best performance, although it slightly reduces the performance in video-to-text retrieval.

We also found that adding more than two hard negatives does not improve the results. Moreover, for the POS-based generation method, reducing the pool size of generated sentences available for sampling lowers the performance.

Another contribution of our work is that we replicated the results of the study by Chen et al. [1] and expanded it by introducing the Brittleness metric to enable more thorough analysis.

This study has several limitations. First, we evaluated only a single model on a single dataset. Given the variability in dataset characteristics—such as sentence complexity and length—as well as differences in model architectures, further experiments are necessary to assess the generalizability of our findings.

Second, we did not explore all possible combinations of hard positive/negative counts and set sizes. Future work could address this limitation through techniques like Bayesian optimization to

systematically explore the hyperparameter space. Additionally, the method used to generate hard positives and negatives deserves further investigation. The quality of these generated sentences may significantly affect the PosRank and Brittleness metrics. Exploring alternative generation techniques—such as using larger language models or varying the POS-based generation strategy—could yield improvements. Defining a metric to evaluate the quality of generated sentences would further help contextualize and compare their impact on performance.

Another open question is why augmentation using POS-generated sentences consistently led to the highest Brittleness scores, particularly for prepositions and adjectives. Investigating this phenomenon could offer valuable insights into the interaction between augmentation strategies and linguistic categories. Furthermore, training on sentences augmented with only one part of speech may help isolate effects caused by statistical mediation and clarify the influence of specific linguistic features.

Combining different generation methods for hard positives and negatives—for example, generating hard negatives using POS-based techniques and hard positives via LLM paraphrasing—also presents a promising direction. Improvements can be made to the training methodology as well. Building on the work by Chen et al. [1], we could incorporate fine-grained supervision by introducing distinct tokens for hard negatives *and* positives, whether textual or visual. Modifications to the loss function are another potential enhancement. While this work uses Formula 5, which considers only the distance between text and video embeddings, alternative formulations—such as the one proposed by Doveh et al. [38]—could incorporate the distance between generated hard positives and all other sentences in the batch.

Lastly, exploring hierarchical training structures, as proposed by Alper and Averbuch-Elor [34], may yield further performance gains and should be investigated for their impact on PosRank and Brittleness scores.

**Acknowledgments:** This work was performed using the ALICE compute resources provided by Leiden University.

# References


[1] A. Chen, H. Doughty, X. Li, and C. G. M. Snoek, "Beyond coarse-grained matching in video-text retrieval," 2024. [Online]. Available: https://arxiv.org/abs/2410.12407

[2] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[3] S. Chen, Y. Zhao, Q. Jin, and Q. Wu, "Fine-grained video-text retrieval with hierarchical graph reasoning," *CoRR*, vol. abs/2003.00392, 2020. [Online]. Available: https://arxiv.org/abs/2003.00392

[4] Y. Cheng, F. Wei, J. Bao, D. Chen, and W. Zhang, "Cico: Domain-aware sign language retrieval via cross-lingual contrastive learning," 2023. [Online]. Available: https://arxiv.org/abs/2303.12793

[5] H. Lu, Y. Huo, G. Yang, Z. Lu, W. Zhan, M. Tomizuka, and M. Ding, "Uniadapter: Unified parameter-efficient transfer learning for cross-modal modeling," 2023. [Online]. Available: https://arxiv.org/abs/2302.06605

[6] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan, "Flamingo: a visual language model for few-shot learning," 2022. [Online]. Available: https://arxiv.org/abs/2204.14198

[7] R. Zellers, J. Lu, X. Lu, Y. Yu, Y. Zhao, M. Salehi, A. Kusupati, J. Hessel, A. Farhadi, and Y. Choi, "Merlot reserve: Neural script knowledge through vision and language and sound," 2022. [Online]. Available: https://arxiv.org/abs/2201.02639

[8] F. Shu, B. Chen, Y. Liao, S. Xiao, W. Sun, X. Li, Y. Zhu, J. Wang, and S. Liu, "Masked contrastive pre-training for efficient video-text retrieval," 2022. [Online]. Available: https://arxiv.org/abs/2212.00986

[9] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," *CoRR*, vol. abs/2002.05709, 2020. [Online]. Available: https://arxiv.org/abs/2002.05709

[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[11] K. Desai and J. Johnson, "Virtex: Learning visual representations from textual annotations," *CoRR*, vol. abs/2006.06666, 2020. [Online]. Available: https://arxiv.org/abs/2006.06666

[12] Y. Zhang, H. Jiang, Y. Miura, C. D. Manning, and C. P. Langlotz, "Contrastive learning of medical visual representations from paired images and text," *CoRR*, vol. abs/2010.00747, 2020. [Online]. Available: https://arxiv.org/abs/2010.00747

[13] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[14] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *CoRR*, vol. abs/1910.10683, 2019. [Online]. Available: http://arxiv.org/abs/1910.10683

[15] A. Torabi, N. Tandon, and L. Sigal, "Learning language-visual embedding for movie understanding with natural-language," 2016. [Online]. Available: https://arxiv.org/abs/1609.08124

[16] Y. Yu, J. Kim, and G. Kim, "A joint sequence fusion model for video question answering and retrieval," 2018. [Online]. Available: https://arxiv.org/abs/1808.02559

[17] Y. Yu, H. Ko, J. Choi, and G. Kim, "Video captioning and retrieval models with semantic attention," 10 2016.

[18] X. Li, C. Xu, G. Yang, Z. Chen, and J. Dong, "W2vv++: Fully deep learning for ad-hoc video search," in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1786–1794. [Online]. Available: https://doi.org/10.1145/3343031.3350906

[19] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, "Frozen in time: A joint video and image encoder for end-to-end retrieval," 2022. [Online]. Available: https://arxiv.org/abs/2104.00650

[20] H. Luo, L. Ji, M. Zhong, Y. Chen, W. Lei, N. Duan, and T. Li, "Clip4clip: An empirical study of CLIP for end to end video clip retrieval," *CoRR*, vol. abs/2104.08860, 2021. [Online]. Available: https://arxiv.org/abs/2104.08860

[21] H. Xu, G. Ghosh, P.-Y. Huang, D. Okhonko, A. Aghajanyan, F. Metze, L. Zettlemoyer, and C. Feichtenhofer, "Videoclip: Contrastive pre-training for zero-shot video-text understanding," 2021. [Online]. Available: https://arxiv.org/abs/2109.14084

[22] S. Zhao, L. Zhu, X. Wang, and Y. Yang, "Centerclip: Token clustering for efficient text-video retrieval," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '22. ACM, Jul. 2022, p. 970–981. [Online]. Available: http://dx.doi.org/10.1145/3477495.3531950

[23] H. Xue, Y. Sun, B. Liu, J. Fu, R. Song, H. Li, and J. Luo, "Clip-vip: Adapting pre-trained image-text model to video-language representation alignment," 2023. [Online]. Available: https://arxiv.org/abs/2209.06430

[24] Y. Liu, P. Xiong, L. Xu, S. Cao, and Q. Jin, "Ts2-net: Token shift and selection transformer for text-video retrieval," 2022. [Online]. Available: https://arxiv.org/abs/2207.07852

[25] Y. Ma, G. Xu, X. Sun, M. Yan, J. Zhang, and R. Ji, "X-clip: End-to-end multi-grained contrastive learning for video-text retrieval," 2022. [Online]. Available: https://arxiv.org/abs/2207.07285

[26] Z. Wang, Y.-L. Sung, F. Cheng, G. Bertasius, and M. Bansal, "Unified coarse-to-fine alignment for video-text retrieval," 2023. [Online]. Available: https://arxiv.org/abs/2309.10091

[27] W. Ma, K. Li, Z. Jiang, M. Meshry, Q. Liu, H. Wang, C. Häne, and A. Yuille, "Rethinking video-text understanding: Retrieval from counterfactually augmented data," 2024. [Online]. Available: https://arxiv.org/abs/2407.13094

[28] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[29] H. J. Levesque, E. Davis, and L. Morgenstern, "The winograd schema challenge," in *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, ser. KR'12. AAAI Press, 2012, p. 552–561.

[30] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi, "Winogrande: An adversarial winograd schema challenge at scale," 2019. [Online]. Available: https://arxiv.org/abs/1907.10641

[31] T. Thrush, R. Jiang, M. Bartolo, A. Singh, A. Williams, D. Kiela, and C. Ross, "Winoground: Probing vision and language models for visio-linguistic compositionality," 2022. [Online]. Available: https://arxiv.org/abs/2204.03162

[32] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, and J. Zou, "When and why vision-language models behave like bags-of-words, and what to do about it?" 2023. [Online]. Available: https://arxiv.org/abs/2210.01936

[33] A. Kamath, C.-Y. Hsieh, K.-W. Chang, and R. Krishna, "The hard positive truth about vision-language compositionality," 2024. [Online]. Available: https://arxiv.org/abs/2409.17958

[34] M. Alper and H. Averbuch-Elor, "Emergent visual-semantic hierarchies in image-text representations," 2024. [Online]. Available: https://arxiv.org/abs/2407.08521

[35] C. Jiang, H. Liu, X. Yu, Q. Wang, Y. Cheng, J. Xu, Z. Liu, Q. Guo, W. Chu, M. Yang, and Y. Qi, "Dual-modal attention-enhanced text-video retrieval with triplet partial margin contrastive learning," in *Proceedings of the 31st ACM International Conference on Multimedia*, ser. MM '23. ACM, Oct. 2023, p. 4626–4636. [Online]. Available: http://dx.doi.org/10.1145/3581783.3612006

[36] F. He, Q. Wang, Z. Feng, W. Jiang, Y. Lv, Y. zhu, and X. Tan, "Improving video retrieval by adaptive margin," 2023. [Online]. Available: https://arxiv.org/abs/2303.05093

[37] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2019. [Online]. Available: https://arxiv.org/abs/1807.03748

[38] S. Doveh, A. Arbelle, S. Harary, R. Panda, R. Herzig, E. Schwartz, D. Kim, R. Giryes, R. Feris, S. Ullman, and L. Karlinsky, "Teaching structured visionlanguage concepts to visionlanguage models," 2023. [Online]. Available: https://arxiv.org/abs/2211.11733

[39] G. A. Miller, "Wordnet: a lexical database for english," *Commun. ACM*, vol. 38, no. 11, p. 39–41, Nov. 1995. [Online]. Available: https://doi.org/10.1145/219717.219748

[40] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Yang, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, X. Liu, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, Z. Guo, and Z. Fan, "Qwen2 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2407.10671

[41] X. Wang, J. Wu, J. Chen, L. Li, Y.-F. Wang, and W. Y. Wang, "Vatex: A large-scale, high-quality multilingual dataset for video-and-language research," 2020. [Online]. Available: https://arxiv.org/abs/1904.03493

[42] L. Zhou, C. Xu, and J. J. Corso, "Procnets: Learning to segment procedures in untrimmed and unconstrained videos," *CoRR*, vol. abs/1703.09788, 2017. [Online]. Available: http://arxiv.org/abs/1703.09788

[43] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. C. Russell, "Localizing moments in video with natural language," *CoRR*, vol. abs/1708.01641, 2017. [Online]. Available: http://arxiv.org/abs/1708.01641

[44] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 961–970.

[45] D. Chen and W. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, D. Lin, Y. Matsumoto, and R. Mihalcea, Eds. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 190–200. [Online]. Available: https://aclanthology.org/P11-1020/

[46] Y. Li, Y. Song, L. Cao, J. R. Tetreault, L. Goldberg, A. Jaimes, and J. Luo, "TGIF: A new dataset and benchmark on animated GIF description," *CoRR*, vol. abs/1604.02748, 2016. [Online]. Available: http://arxiv.org/abs/1604.02748

[47] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," 2017. [Online]. Available: https://arxiv.org/abs/1705.06950

[48] P. Voigtlaender, S. Changpinyo, J. Pont-Tuset, R. Soricut, and V. Ferrari, "Connecting vision and language with video localized narratives," 2023. [Online]. Available: https://arxiv.org/abs/2302.11217

# Appendix

## A Definitions

**Notation** $D$ - dataset with videos and captions for each video; $D_{train}, D_{val}, D_{test}$ refer to the train, validation, and test subsets respectively
$V$ - set of videos in $D_{train}$, $D_{val}$, or $D_{test}$
$T$ - set of captions in $D_{train}$, $D_{val}$, or $D_{test}$
$T_i$ - set of captions for video $v_i$; $T_i \in T$
$t_j$ - one of the captions for video $v_i$; $t_j \in T_i$
$N_{ij}$ - set of hard negative captions for $t_j$; $|N_{ij}| = n$
$P_{ij}$ - set of hard positive captions for $t_j$; $|P_{ij}| = m$
$P = \{$noun, verb, adverb, adjective, preposition$\}$ - set of parts of speech we use to make changes to the sentences
$p$ - some part of the speech; $p \in P$

**Abbreviations** POS - part of speech

## B Dataset Sentences Statistics

Here is the detailed information about the sentence properties in the collected VATEX dataset subset. A visual representation is presented in Figure 19.

**Total sentences analyzed:** 295,430
**Mean words per sentence:** 14.25
**Median words per sentence:** 13.00
**Minimum words in a sentence:** 8
**Maximum words in a sentence:** 69
**Quartiles:**

- Q1 (25%): 12.00
- Q2 (50%): 13.00
- Q3 (75%): 16.00

**Example of shortest sentence:** *"two men on stage and performing a show."*
**Example of longest sentence:** *"A girl in a swimsuit jumps off of a cliff into a body of water, another girl stands above on the cliff, a person sits on a jet ski and various people sit and stand in boats on the water, two girls laugh in a boat, and then a man with a microphone talks to a girl while a large group of people stand in boats in the background."*

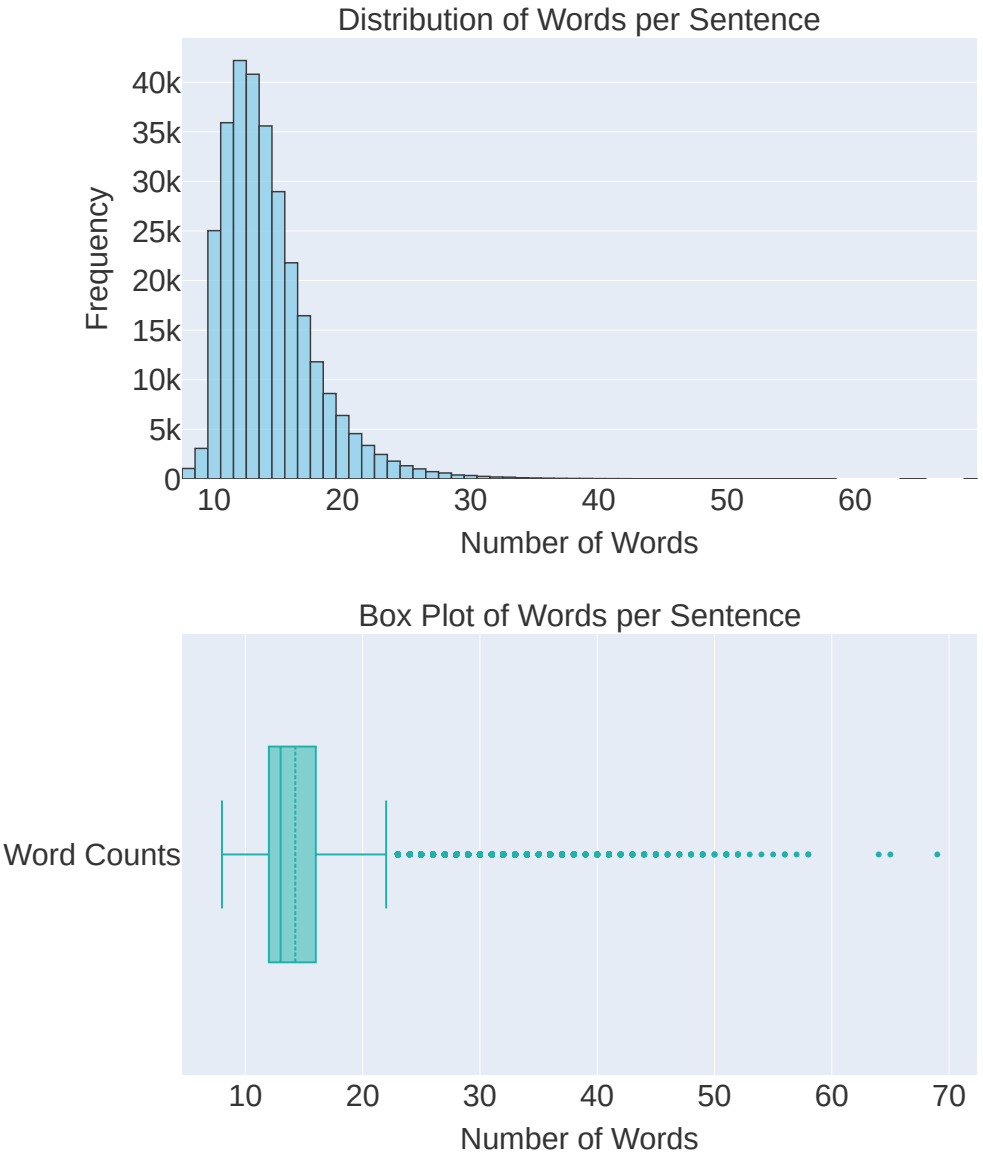


Figure 19: The distribution of sentence length in the collected VATEX subset

# C Prompts Used for Generation

**Hard positives:**

> Generate [number of sentences] different paraphrases of the following sentence that retain the same meaning but use different wording. Only output the paraphrases, one per line, without any additional text.
>
> Sentence: [caption]

**Hard Negatives:**

> I will give you a sentence. Generate [number of sentences] hard negative sentences for it. A hard negative sentence is very similar in wording and structure to the original, but the meaning is different or opposite. Start by changing keywords to antonyms or contrasting terms, or modifying the actions to contradict the original meaning. Keep the sentences fluent and grammatically correct.
>
> Example:
> Input: A man is hiking.
> Output:
> A woman is hiking.
> A female is hiking.
> A man is sitting.
> A man is lying.
> ...
>
> Now generate hard negatives for this sentence:
>
> [caption]

# D Statistics of Sentences Generated With POS-methodology

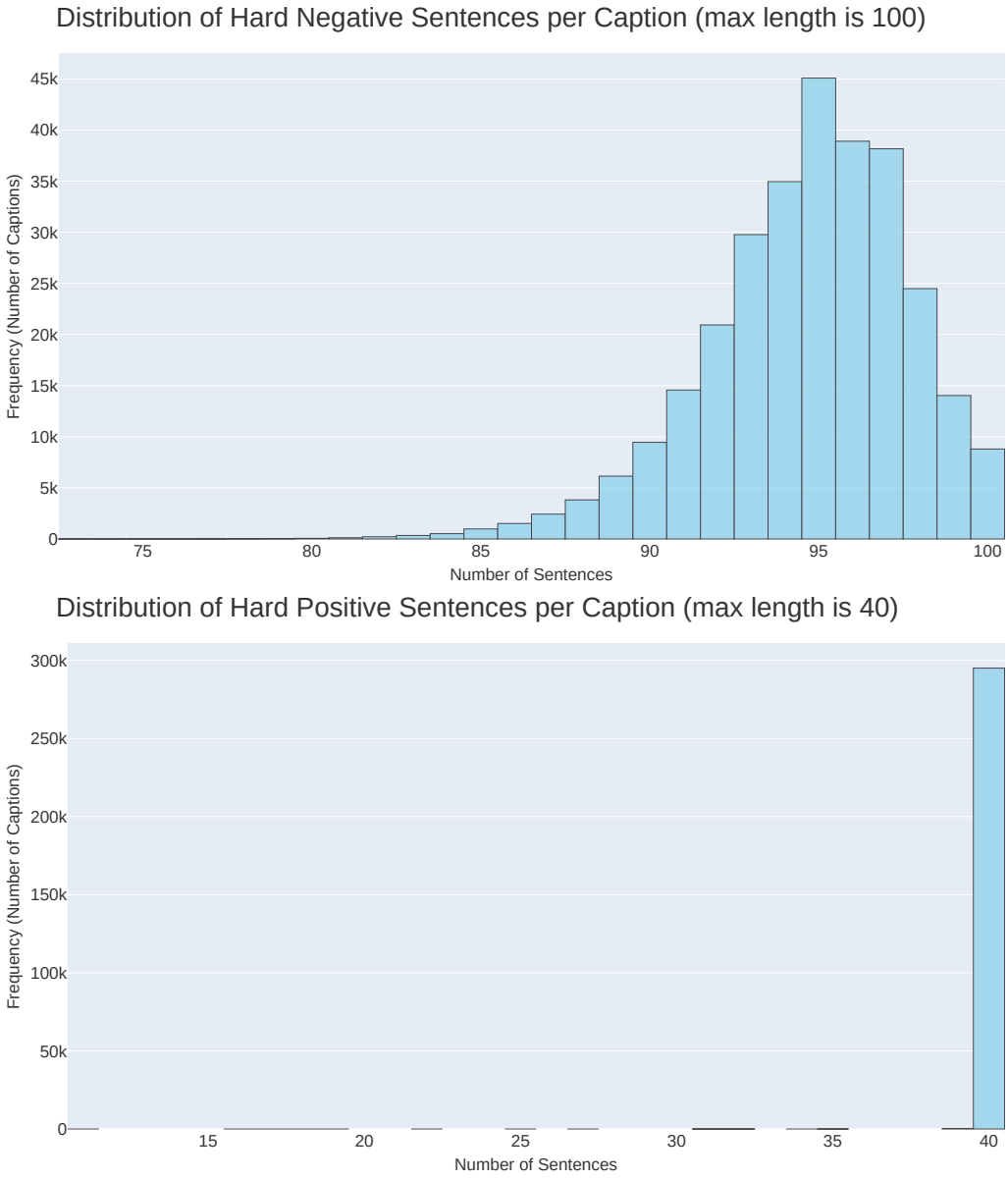


Figure 20: Distributions of generated hard negative (top) and hard positive (bottom) sentences for the max set sizes of 100 and 40, respectively. For hard negatives, the number of generated sentences per caption ranges from 73 to 100. For hard positives, the range is from 11 to 40.

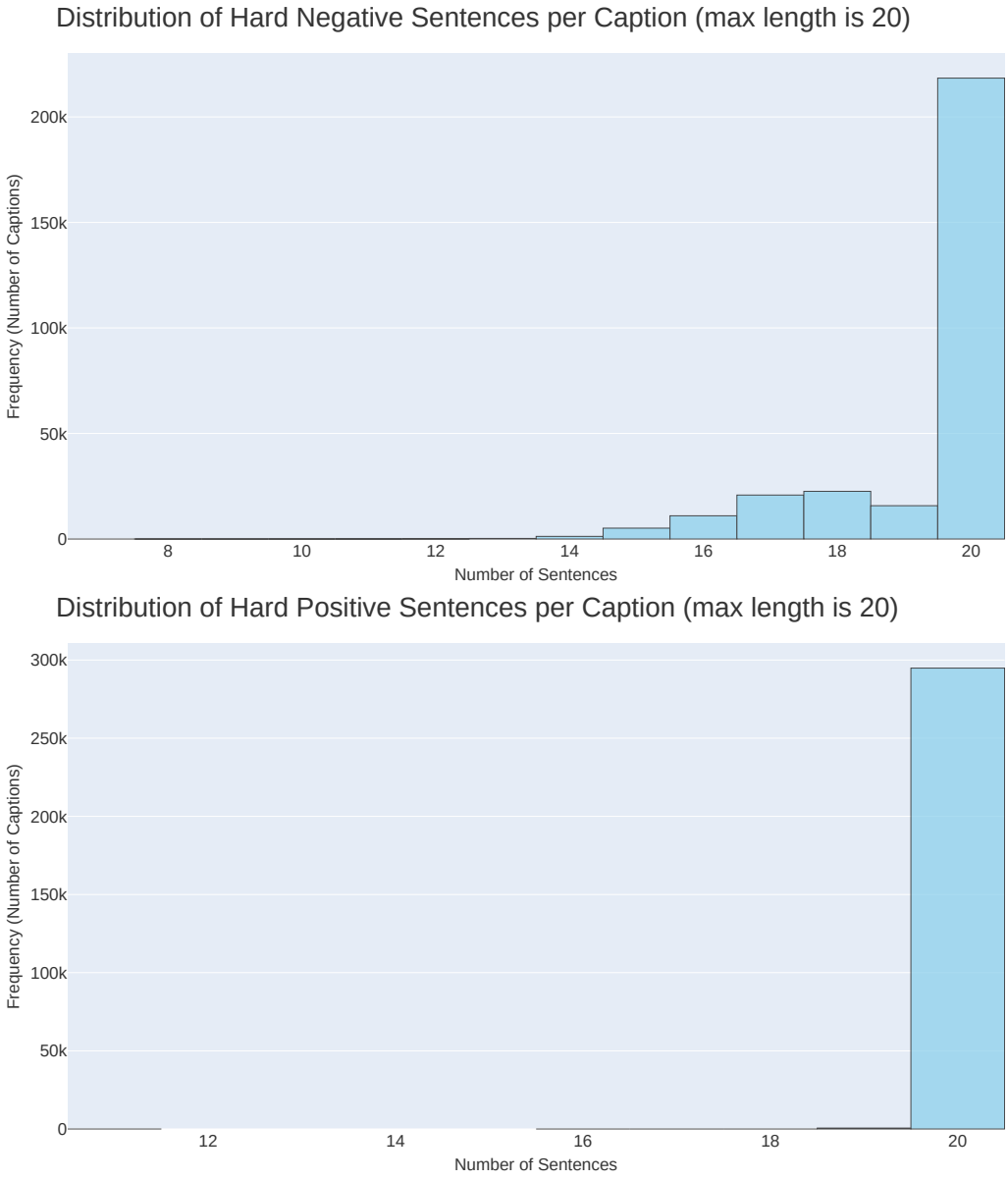


Figure 21: Distributions of generated hard negative (top) and hard positive (bottom) sentences for the max set sizes of 20 and 20, respectively. For hard negatives, the number of generated sentences per caption ranges from 7 to 20. For hard positives, the range is from 11 to 20.

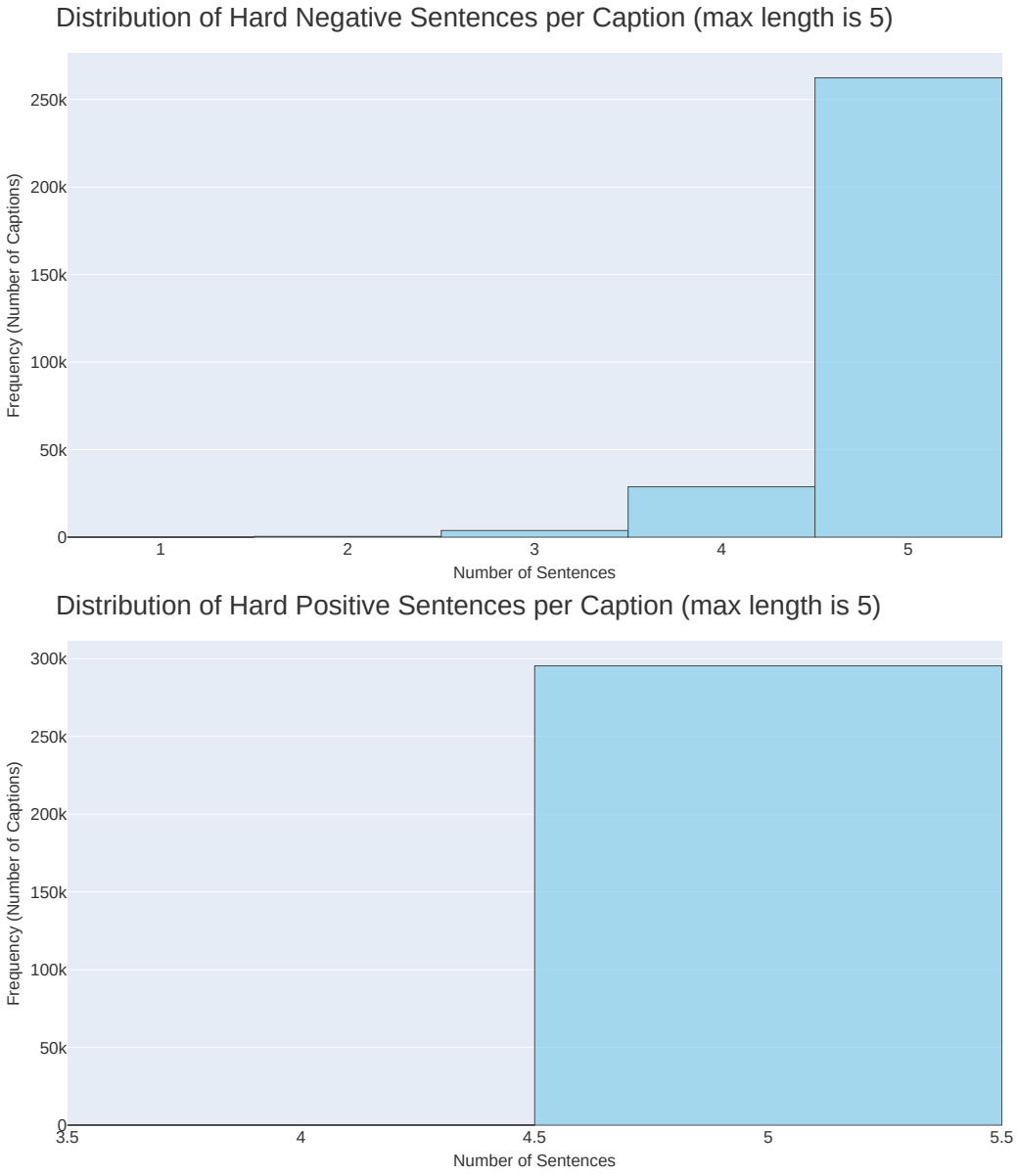


Figure 22: Distributions of generated hard negative (top) and hard positive (bottom) sentences for the max set sizes of 20 and 20, respectively. For hard negatives, the number of generated sentences per caption ranges from 1 to 5. For hard positives, the range is from 4 to 5.

# E Complete Table of Example Generated Sentences

Table 4: Examples of Hard Positive and Negative Sentence Generation

| Method | Generated Sentence | Set Size |
|---|---|---|
| **Original:** *A couple of people roping a calf in a ring and riding horses.* | | |
| **POS Hard Negative** | A couple of people roping a calf in a shanty and riding horses.<br>A couple of dead roping a calf in a ring and riding horses.<br>A couple of people roping a calf in a open chain and riding horses. | 100<br>20<br>5 |
| **POS Hard Positive** | A couple of citizenry roping a calf in a ring and riding horses.<br>A duo of people roping a calf in a ring and riding horses.<br>A twain of people roping a calf in a ring and riding horses. | 40<br>20<br>5 |
| **LLM Hard Negative** | A party of individuals are attempting to herd an animal using only their legs. | – |
| **LLM Hard Positive** | Two folks encircling a bull with ropes as they gallop their mounts. | – |
| **Original:** *A man operates a power tool and then lays it down on the cement.* | | |
| **POS Hard Negative** | A man pruned a power tool and then lays it down on the cement.<br>A female operates a power tool and then lays it down on the cement.<br>A civilian operates a power tool and then lays it down on the cement. | 100<br>20<br>5 |
| **POS Hard Positive** | A man operates a power tool and then pose it down on the cement.<br>A gentleman's gentleman operates a power tool and then lays it down on the cement.<br>A human race operates a power tool and then lays it down on the cement. | 40<br>20<br>5 |
| **LLM Hard Negative** | A expert electrician activates a soldering iron and then lies on the rocky slope. | – |
| **LLM Hard Positive** | An individual manages a power device before putting it down on concrete. | – |
| **Original:** *A young girls is sitting in a high chair, and she begins to shake her body and then her head.* | | |
| **POS Hard Negative** | A young girls differ sitting in a high chair, and she begins to shake her body and then her head.<br>A young male is sitting in a high chair, and she begins to shake her body and then her head.<br>A young son is sitting in a high chair, and she begins to shake her body and then her head. | 100<br>20<br>5 |
| **POS Hard Positive** | A young girls is sitting in a high chair, and she begins to shake her body and then her mind.<br>A young girls is sitting in a high chairwoman, and she begins to shake her body and then her head. | 40<br>20 |

Continued on next page

| Method | Generated Sentence | Set Size |
|---|---|---|
| | A young female child is sitting in a high chair, and she begins to shake her body and then her head. | 5 |
| **LLM Hard Negative** | An elderly lady is sitting in a high chair, and she starts to sway her arms before shaking her torso. | – |
| **LLM Hard Positive** | A diminutive girl is placed on a high seat, and she commences twirling her body and afterward arching her neck. | – |
| **Original:** *A boy sings while slapping the hands of an adult while both are seated on the floor.* | | |
| **POS Hard Negative** | A boy sings while slapping the gig of an adult while both are seated on the floor.<br>A boy sings while slapping the hands of an conservative while both are seated on the floor.<br>A boy sings while slapping the hands of an woman while both are seated on the floor. | 100<br>20<br>5 |
| **POS Hard Positive** | A boy sings while slapping the script of an adult while both are seated on the floor.<br>A boy sings while slapping the custody of an adult while both are seated on the floor.<br>A boy sings while slapping the manpower of an adult while both are seated on the floor. | 40<br>20<br>5 |
| **LLM Hard Negative** | A young girl warbles with her hands lightly stroking another person's scalp while perched on the edge of a bed. | – |
| **LLM Hard Positive** | A young man performs as he beats his palms against an elder's hands, all sitting together on the ground. | – |
| **Original:** *a man is showing how a square knot looks like with a white string.* | | |
| **POS Hard Negative** | a man miss showing how a square knot looks like with a white string.<br>a black is showing how a square knot looks like with a white string.<br>a civilian is showing how a square knot looks like with a white string. | 100<br>20<br>5 |
| **POS Hard Positive** | a man is picture how a square knot looks like with a white string.<br>a humans is showing how a square knot looks like with a white string.<br>a adult male is showing how a square knot looks like with a white string. | 40<br>20<br>5 |
| **LLM Hard Negative** | A man tries to teach others how to perform a square knot using blue rope. | – |
| **LLM Hard Positive** | A person clarifies the elements of a square shackle employing a soft beige thread. | – |

# F Distribution of Parts of Speech in Generated Hard Negatives and Positives

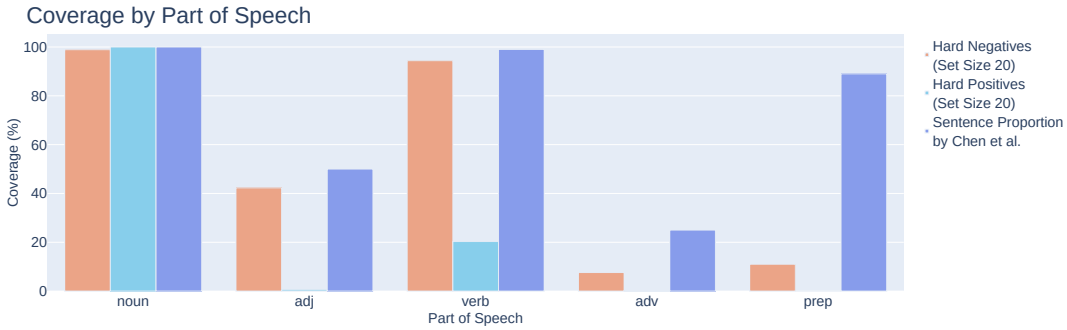


Figure 23: Distribution of parts of speech across generated sentences. For the first two legend groups (hard negatives and positives), the coverage value indicates the percentage of sentences in which a specific part of speech was substituted. For the final legend group ("Sentence Proportion by Chen et al."), the value represents the proportion of sentences in the VATEX [41] dataset containing the corresponding part of speech, as reported by Chen et al. [1].

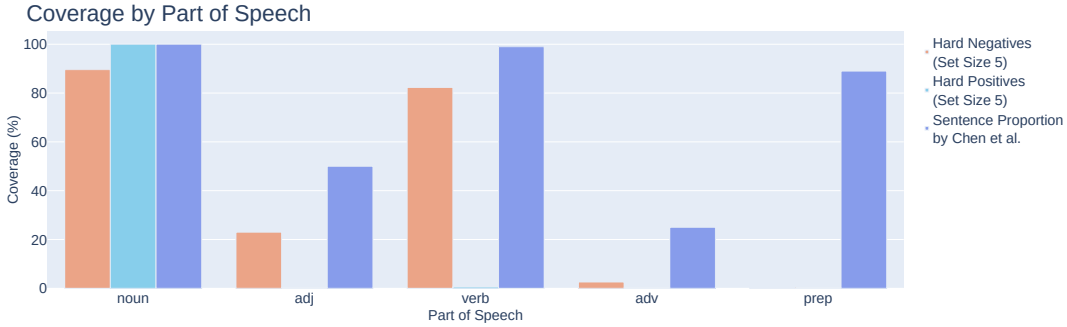


Figure 24: Distribution of parts of speech across generated sentences. For the first two legend groups (hard negatives and positives), the coverage value indicates the percentage of sentences in which a specific part of speech was substituted. For the final legend group ("Sentence Proportion by Chen et al."), the value represents the proportion of sentences in the VATEX [41] dataset containing the corresponding part of speech, as reported by Chen et al. [1].

# G Recall@k and Mean Rank Plots

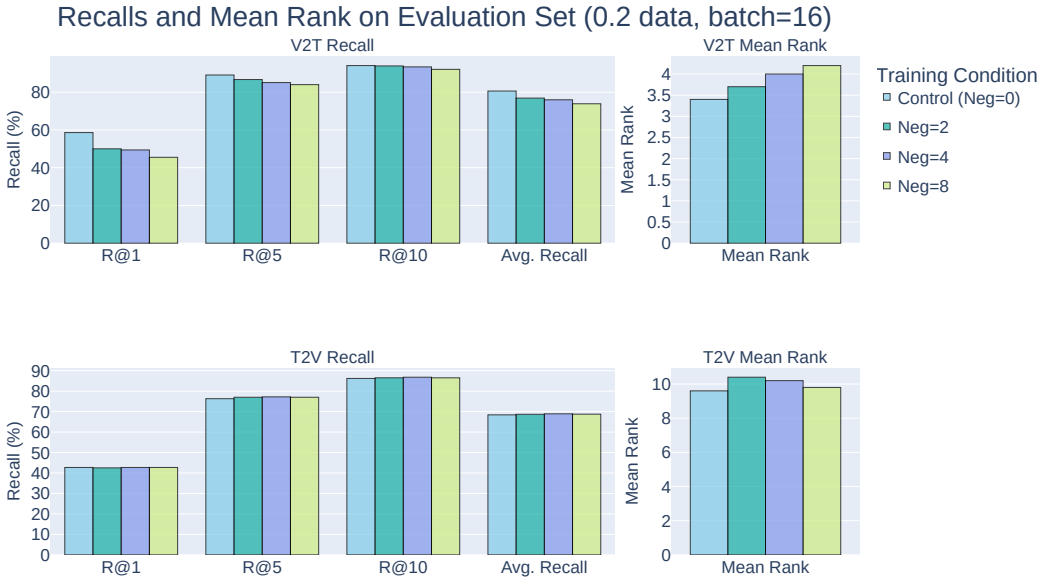


Figure 25: Evaluation results for Recall@k and Mean Rank across different quantities of hard negatives used during training. Avg. Recall represents the average of Recall@1, Recall@5, and Recall@10. V2T and T2V refer to vision-to-text and text-to-vision retrieval tasks, respectively. This figure is an addition to the Figure 12.

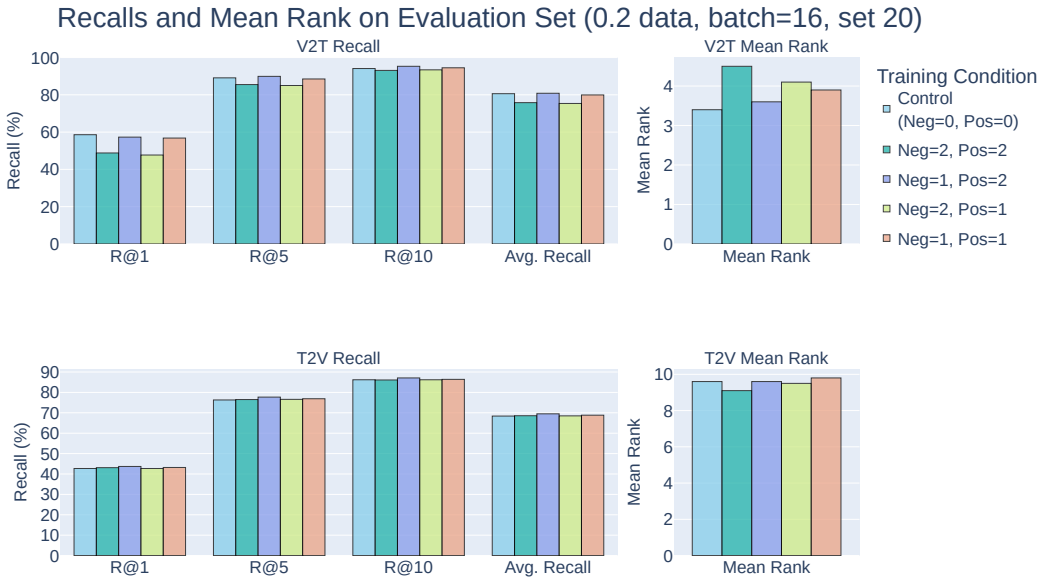


Figure 26: Results for Recall evaluations or the study of the effect of a set size, we sample hard positives or negatives from on the performance of the model. Avg. Recall represents the average of Recall@1, Recall@5, and Recall@10. V2T and T2V refer to vision-to-text and text-to-vision retrieval tasks, respectively. This figure is an addition to the Figure 17.

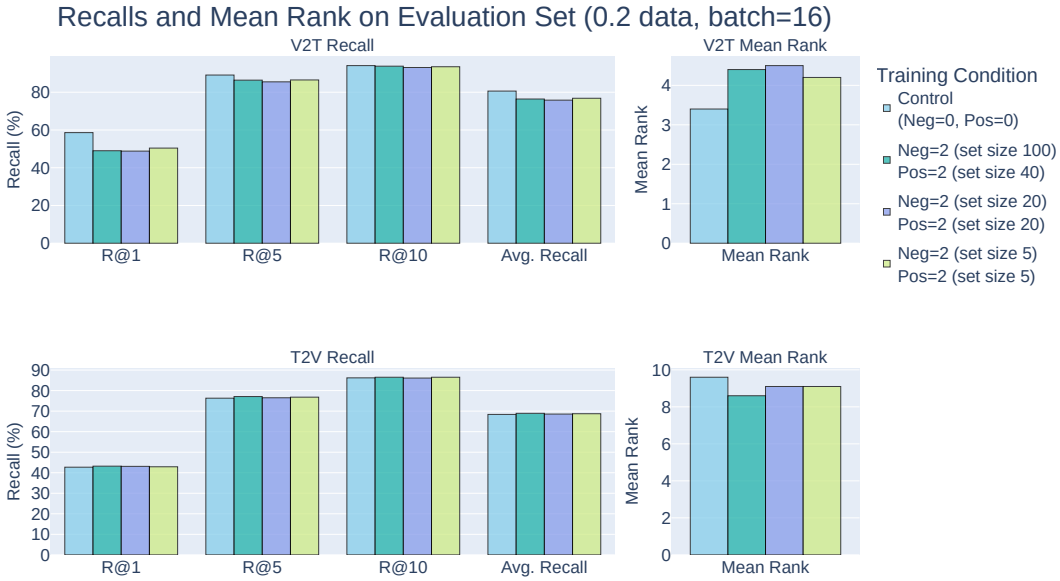


Figure 27: Results for Recall evaluations for the study of proportions of the inclusion of hard negatives or positives into a training process. Avg. Recall represents the average of Recall@1, Recall@5, and Recall@10. V2T and T2V refer to vision-to-text and text-to-vision retrieval tasks, respectively. This figure is an addition to the Figure 18.