Learn to Understand Negation in Video Retrieval

Ziyue Wang*
AIMC Lab, School of Information, Renmin University of China
China

Aozhu Chen*
AIMC Lab, School of Information, Renmin University of China
China

Fan Hu
AIMC Lab, School of Information, Renmin University of China
China

Xirong Li†
MoE Key Lab of DEKE, Renmin University of China
China

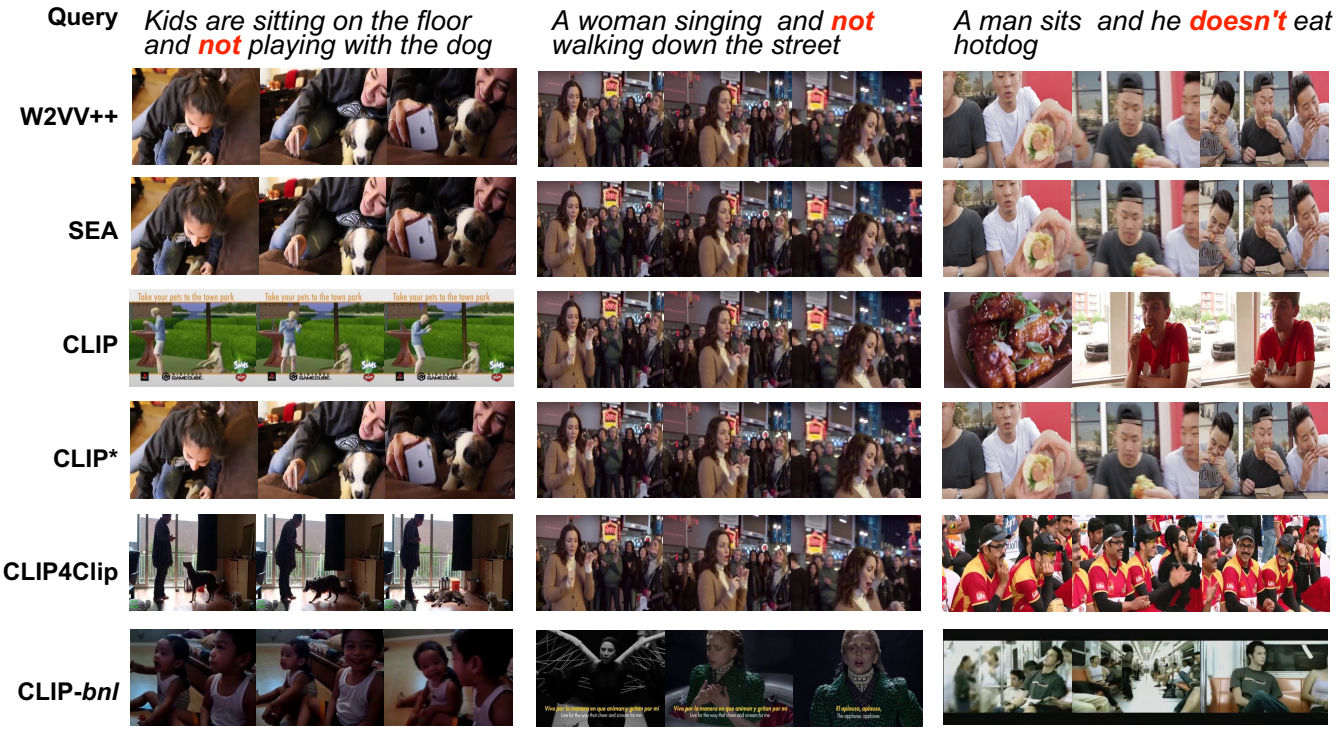


Figure 1: Top-1 video retrieved by different models, *i.e.* W2VV++ [19], SEA [20], CLIP [28], CLIP* (fine-tuned by this work), CLIP4Clip [25] and our CLIP-*bnl*, which is CLIP re-trained with proposed negation learning. This paper presents the first study on a learning based method for handling negation in text-to-video retrieval (nT2VR). Data source: MSR-VTT [32].

ABSTRACT

Negation is a common linguistic skill that allows human to express what we do NOT want. Naturally, one might expect video retrieval to support natural-language queries with negation, e.g., finding shots of kids sitting on the floor and not playing with a dog. However, the state-of-the-art deep learning based video retrieval models lack such ability, as they are typically trained on video description datasets such as MSR-VTT and VATEX that lack negated descriptions. Their retrieved results basically ignore the negator in the sample query, incorrectly returning videos showing kids playing with dog. This paper presents the first study on learning to understand negation in video retrieval and make contributions as follows. By re-purposing two existing datasets (MSR-VTT and VATEX), we propose a new evaluation protocol for video retrieval with negation. We propose a learning based method for training a negation-aware video retrieval model. The key idea is to first construct a soft negative caption for a specific training video by partially negating its original caption, and then compute a bidirectionally constrained loss on the triplet. This auxiliary loss is weightedly added to a standard retrieval loss. Experiments on the re-purposed benchmarks

*Z. Wang and A. Chen contributed equally to this research.
†Corresponding author: Xirong Li (xirong@ruc.edu.cn).

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3547968

show that re-training the CLIP (Contrastive Language-Image Pre-Training) model by the proposed method clearly improves its ability to handle queries with negation. In addition, the model performance on the original benchmarks is also improved.

CCS CONCEPTS

• **Information systems** → **Video search**; **Test collections**.

KEYWORDS

Text-to-video retrieval (T2VR), nT2VR, negation learning

ACM Reference Format:
Ziyue Wang, Aozhu Chen, Fan Hu, and Xirong Li. 2022. Learn to Understand Negation in Video Retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3503161.3547968

1 INTRODUCTION

This paper is targeted at text-to-video retrieval (T2VR), also known as video retrieval by text. T2VR aims to let common users retrieve the increasing amounts of unlabeled videos by textual queries. Due to its high practical value, the topic has attracted much attention recently [13, 19, 24, 26, 30, 31]. These dedicated research efforts have been well paid off, with continuous performance improvement reported on both public datasets [29, 32] and international benchmark evaluations [1, 2]. It seems unquestionable that more powerful T2VR models will be developed to help users find what they want. The question is do the (current) models understand what the users do *not* want?

Negation is an important and common linguistic skill for human beings to express what we do not want. A query with negation can be "finding shots of kids sitting on the floor and *not* playing with the dog". As exemplified in Fig. 1, a number of current models, *e.g.* W2VV++ [19], CLIP [28] and CLIP4Clip [25], are actually not good at answering this specific query. Recall that these models are trained on video description datasets, such as MSVD [3], MSR-VTT [32] and VATEX [29], which were originally developed for the video captioning task. For that task, annotators tend to describe what was present in the video content other than what was absent. By a rule-based negation cue detection, we find that only 1.5% of the MSR-VTT video descriptions contain negation cues such as *no*, *not*, and *without*. . The lack of negated descriptions in the training data has a clear consequence on the models. Their retrieved results basically ignore the negator in the sample query, incorrectly returning videos showing kids playing with the dog, see Fig. 1.

Towards addressing negation in T2VR (nT2VR), an initial attempt has been made by Wu and Ngo [30]. The authors describe a rule-based strategy to process queries with negation. In particular, given a query such as *beach not man*, the strategy treats the query as a logic expression of *beach* AND (NOT *man*). The boolean operation is practically implemented by subtracting a video's relevance score to *man* from its relevance score to *beach*. We consider their study preliminary as they experimented with only five hand-crafted queries. More importantly, the boolean operation is essentially a post-processing trick. The underlying T2VR model remains unaware of the negation.

In this paper, we present a first study on learning to understand negation in T2VR. Our major contributions are as follows:

- Due to the absence of related data and evaluation criteria, we introduce a new evaluation protocol. In particular, we re-purpose MSR-VTT and VATEX by automatically constructing thousands of negated and composed queries from the original descriptions. Such a re-purposing allows the protocol to support large-scale evaluation without the need of extra manual labeling. By preserving the original test queries, the new protocol can also be used to test how a negation-aware model performs on the original benchmarks.
- We propose a learning based method for training a negation-aware T2VR model. Specifically, given a training video, its original description and a partially negated description, we compute on the triplet a bidirectionally constrained loss. Consequently, negation learning (NL) is realized with ease by adding this auxiliary loss to a standard retrieval loss.
- Extensive experiments on the two re-purposed benchmarks show that re-training the CLIP (Contrastive Language-Image Pre-Training) model [28] by the proposed method clearly improves the model's ability to handle queries with negation. In addition, its performance on the original benchmarks is also improved. Data and code are available at GitHub[1].

The remaining part is organized as follows. We discuss related work in Section 2. The new evaluation protocol is described in Section 3, followed by the NL method in Section 4 and experiments in Section 5. Major conclusions are presented in Section 6.

2 RELATED WORK

Progress on T2VR. Depending on whether visual / text encoders used to extract raw features from videos / queries are frozen, we divide current methods for T2VR into the following two groups, *i.e.* feature re-learning methods [8, 19, 26, 31, 33] and end-to-end methods [11, 15, 25].

A feature re-learning method typically has a two-stage working pipeline. In the first stage, one or multiple pre-trained 2D/3D CNNs are used to extract frame-level or segment-level features from the video content, whilst pre-trained language models, *e.g.* word2vec (w2v) or BERT, are adopted for extracting dense features from the text. Consider the W2VV++ series [19, 20, 24] for instance. The method represents queries by concatenating the output of multiple existing text encoders. CE [23], MMT [12] and MDMMT [8] exploit multiple visual features that capture motion, appearance, face and OCR, respectively. We refer to [2] for more details about the choice of the features. In the second stage, the pre-extracted visual / textual features are fed into a cross-modal representation learning network so that the re-learned features can be directly used for cross-modal matching. The choice of the network varies, raging from a simple feedforward network used by W2VV++, multi-level encoding networks used by DualEncoding [7], DualTask [30], HGR [4] and HANet [31], to more complicated graph auto-encoders used by FCA-Net [13]. The capability of these methods to handle negation is subject to their raw textual features. If these features are initially not discriminative to negation, their ability to represent negation is unlikely to be improved by feature re-learning.

[1] https://github.com/ruc-aimc-lab/nT2VR

Thanks to the advent of CLIP (Contrastive Language-Image Pre-Training) [28], end-to-end methods for T2VR have been developed recently. Built on the top of CLIP, CLIP-FT [15], CLIP4Clip [25] and CLIP2Video [11] have shown superior performance over the feature re-learning methods on multiple T2V benchmark datasets including MSR-VTT [32], MSVD [3] and VATEX [29]. However, their ability to answer nT2VR is unknown so far.

Wu and Ngo [30] describe briefly a boolean operation to tackle queries with negation. In that work, a total of five queries were manually created, *i.e. beach not man*, *face not woman*, *drinking not wine or beer*, *flower not red or yellow*, and *two people kissing not bride and groom*. Per query, *e.g. face not woman*, its score to a given video is computed as the video's similarity to the positive subquery (*face*) subtracted by its similarity to the negative subquery (*woman*). Note that the above operation is essentially post-processing, leaving the problem of negative learning untouched.

Earlier efforts have been made on exploiting negative feedback for T2VR, yet all in an *interactive* search mode. For instance, Cooper *et al.* [5] describe an interactive video retrieval system where a user can manually label the currently retrieved video shots either as positive or negative (a.k.a. non-relevant). The system then exploits the negative shot set to implement negative reinforcement / feedback. As such, the user's negative intent has to be specified after the first-round search and indirectly via labeling specific shots as negative. By contrast, this paper is targeted at *automated* search, allowing a user to directly express what she or he does not want in a natural-language query in the first place. Hence, the proposed negation-aware video retrieval is conceptually novel and technically orthogonal to negative feedback.

Understanding Negation in Large-scale Language Modeling. Large-scale pre-trained language models (PLM), as exemplified by BERT [6], have demonstrated impressive performance on varied NLP tasks. However, recent works report that PLM's comprehension over negation is not satisfying [9, 14, 17]. Kassner and Schütze show by their experiments that the probability for PLM to generate "Birds cannot fly" is nearly the same as "Birds can fly" [17]. Hosseini *et al.* [14] report that when filling the blank with negation, *e.g.* "The macOS was not developed by __", BERT answered with "Apple" in spite of the negator. To improve negation understanding of BERT, the authors propose to use an unlikelihood objective on negated sentences. Targeted at NLP tasks, the above technique is not directly applicable for addressing nT2VR.

3 PROPOSED EVALUATION PROTOCOL

As we have noted earlier, benchmark for evaluating nT2VR is nonexistent. We choose to re-purpose two public video-caption datasets, *i.e.* MSR-VTT [32] and VATEX [29], commonly used in the T2VR literature. This is achieved by (partially) negating original queries and composing novel and controlled queries in Section 3.1. Evaluation criteria suited for the re-purposed datasets are presented in Section 3.2. All this provides a new evaluation protocol for comprehensively assessing an (existing) T2VR model's ability to handle original, negated, and composed queries.

3.1 New Query Construction

3.1.1 Negated Query Construction. In order to construct (partially) negated queries from the original video captions, we use a simple rule-based strategy as follows. Given a caption q, we first use the NLTK[2] part-of-speech (POS) tagging API to identity verbs (VERB) and auxiliary verbs (AUX). Then, a negation cue is inserted right before the identified verb, *e.g. while* [not] *dancing with many other people*, or after the AUX, *e.g. there is* [not] *a fight at a basketball game*, yielding a negated variant of the caption, denoted by q^-. Due to the richness of the video content, a caption typically contains multiple verbs. In such cases, one of the verbs or AUX is randomly chosen to be negated, making q^- partially negated *w.r.t.* q. See the appendix for more instances.

There also exists a relatively small amount of captions originally having negation cues, *e.g. a boy running is running without dress*. For these captions, we negate their original meaning by removing the negation cues, *e.g. a boy running is running* [with] *dress*.

When using q^- as a query, the video corresponding to q now becomes negative, see Fig. 2(a). Hence, a T2VR model that can handle negation shall rank the video lower. However, a downside of the negated query is that we are unsure which other videos are truly relevant *w.r.t.* the negated query. Evaluating on the negated queries alone is thus insufficient. Next, we propose to compose queries with negation that have reference videos available.

3.1.2 Composed Query Construction. For constructing composed queries, we first extract two linguistic groups, *i.e.* subjects and verb phrases (VP), from the captions. Instances of subjects are *a man*, *people* and *a car*, while instances of VPs are *take selfie*, *drive down a road* and *play basketball*. Algorithm 1 shows python-style pseudo code for extracting pairs of subjects and VPs from an (unrestricted) video caption.

Algorithm 1: Subject and verb phrase (VP) extraction by NLTK chunk parsing

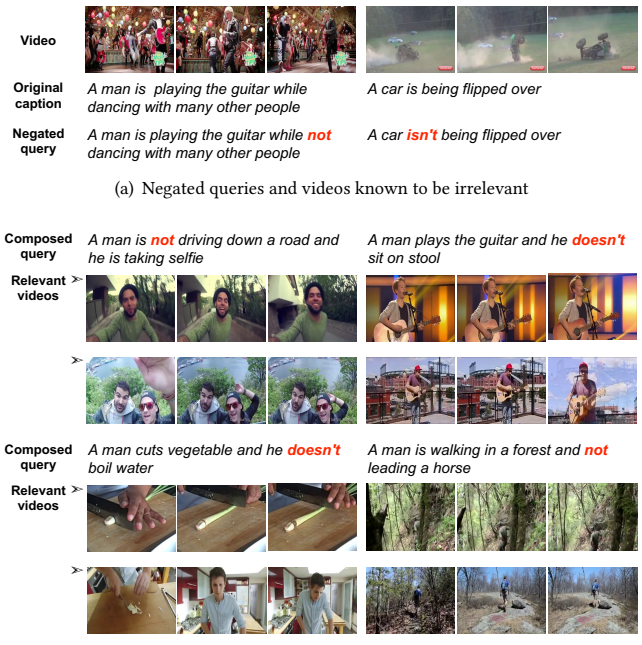
Input: A video caption q
Output: A list of (subject, VP) pairs *s_vp_list*

```
#Tag patterns for specific types of chunks
grammar = '''
 NP: {<DT|JJ|NN.*>*<NN.*>} # A noun-phrase chunk
 PP: {<IN|RP><NP>} # A prepositional-phrase chunk
 VP: {<VB.*><NP|PP|CLAUSE>*} # A verb-phrase chunk
 CLAUSE: {<NP><VP>}  # A clause chunk
 '''
chunk_parser = nltk.RegexpParser(grammar)
tokens = nltk.word_tokenize(q)
tagged_tokens = nltk.pos_tag(tokens)
chunked_text = chunk_parser.parse(tagged_tokens)

s_vp_list = []
for vp in chunked_text.VPs:
    subject = find_subject(checked_text, vp)
    s_vp_list.append((subject, vp))
```

In order to ensure both linguistic and semantic soundness, our composed queries consist of a subject followed by two VPs, one used as positive, while the other used as negative. Given the above

[2]https://github.com/nltk/nltk



(a) Negated queries and videos known to be irrelevant

(b) Composed queries and videos known to be relevant

Figure 2: Illustration of (a) negated and (b) composed queries in the proposed evaluation protocol. Data source: MSR-VTT.

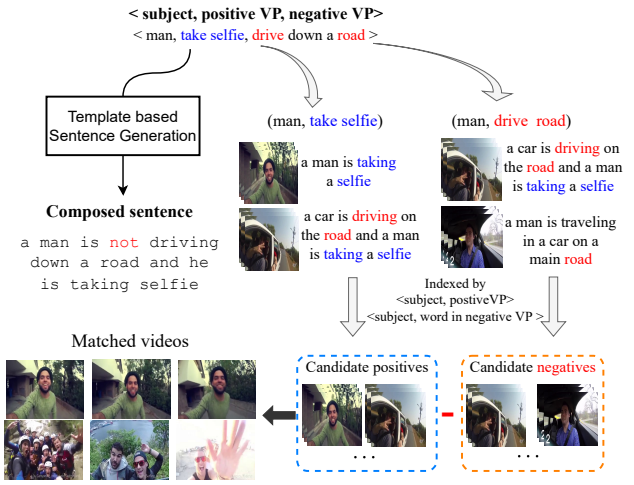


Figure 3: Key data flow of constructing a composed query and its matched videos. Given a subject (*man*), a positive VP (*take selfie*) and a negative VP (*drive down a road*), we use template-based sentence generation to obtain a composed query (*a man is taking selfie and he is not driving down a road*). To find matched videos in the training data, we conduct phrase-level text retrieval on the video captions to identify a set of candidate positive videos and word-level text retrieval to identity a set of candidate negative videos. The matched videos are obtained by set difference.

triplet, a novel query is produced by template-based sentence generation. For instance, given <*man*, *take selfie*, *drive down a road*>, we have the following two queries: *a man is taking selfie and he is not driving down a road* and *a man is not driving down a road and he is taking selfie*. As illustrated in Fig. 3, in order to find reference videos in a given training set, we use the positive VP to conduct phrase-level text retrieval on the video captions to identify a set of candidate positive videos. In order to exclude false positives, we favor precision over recall. Therefore, we use each word from the negative VP to perform word-level text retrieval to identity videos that are positive *w.r.t.* the word and thus being possibly negative *w.r.t.* the composed query. By a set-difference operation between the positive video set and the negative set, matched videos are found. It is possible that the operation may produce an empty set. In this case, the composed query will be discarded. By doing so, we effectively remove queries that describe scenes that are either counter-fact or rarely occur in the real world.

3.1.3 Re-purposed Datasets. We perform negated / composed query construction on MSR-VTT and VATEX. For MSR-VTT, we adopt two data-split editions. One is provided by the dataset developers [32], with 3k test videos, while the other is specified by Yu *et al.* [35] with 1k test videos. The two editions are referred to as MSR-VTT3k and MSR-VTT1k, respectively. For VATEX, we follow the data split[3] by Chen *et al.* [4]. The amount of original, negated and composed queries per test set is summarized in Table 1.

[3]The number of videos used in this work is slightly less than the official number, as some videos were no longer available when downloading.

Table 1: Public datasets re-purposed by this paper for evaluating text-to-video retrieval (T2VR) with negation (nT2VR).

Dataset	Test videos	Test queries		
		Original	*Negated*	*Composed*
MSR-VTT3k [32]	3,000	59,800	59,668	18,157
MSR-VTT1k [35]	1,000	1,000	923	3,697
VATEX [29]	1,398	13,980	7,339	8,394

For a quick quality assessment, we randomly sampled 300 composed queries, and manually checked whether the matched videos are truly relevant *w.r.t.* the queries. About 90% of the queries are correctly associated with relevant videos. The composed queries are of sufficient quality for reliable evaluation.

3.2 Evaluation Criteria

For the original test queries, we report the commonly used Recall at Rank N ($R@N$, $N = 1, 5, 10$), *i.e.* the percentage of test queries that have their answers successfully retrieved among the top-N ranked videos. In addition, we report Mean Inverted Rank (MIR), which reflects how the reference videos are positioned in the overall ranking list. The same metrics are used to evaluate a model's performance on the composed queries.

For each negated query q^-, we can trace back to its original query q, and thus know that the video associated with q shall be negative *w.r.t.* q^-. Hence, the difference between the video's rank *w.r.t.* q

and that *w.r.t.* q^- reflects a model's sensitivity to the introduced negation. In that regard, we report $\Delta R@N$, computed as

$$\Delta R@N(q^-) = R@N(q) - R@N(q^-). \tag{1}$$

In a similar manner we compute ΔMIR.

Ideally, a model more sensitive to negation shall produce larger values regarding both $\Delta R@N$ and ΔMIR. However, using the Δ metrics alone is insufficient. Consider, for instance, a trivial solution that simply reverses the ranking list. Larger ΔR or ΔMIR shall not be interpreted as better retrieval performance. Therefore, the performance on the composed query set shall be treated as primary, while the performance on the negated query set is secondary.

4 PROPOSED LEARNING-BASED METHOD

In order to make our paper more self-contained, we first present some preliminaries concerning current deep learning based T2VR methods in Section 4.1. How to handle negation in a learning-based manner is depicted in Section 4.2.

4.1 Preliminaries

At a high level, a deep learning based T2VR model $\mathcal{M}$ works as follows. Given a textual query q and a short video x, the model uses a text encoder, denoted by $\mathcal{M}_t$, and a visual encoder, denoted by $\mathcal{M}_v$, to project the query and the video into a d-dimensional common space. We use $\mathcal{M}_t(q)$ and $\mathcal{M}_v(x)$ to indicate the resultant textual and visual embedding vectors, respectively. A text-video similarity $s(x, q)$ is typically computed as the cosine similarity between the two embeddings [7, 20]. Accordingly, T2VR on a video collection is achieved by first scoring each video by $s(x, q)$ and then sorting them in descending order to acquire the top-ranked videos.

For model training, a set of manually captioned videos are required for jointly optimizing $\mathcal{M}_t$ and $\mathcal{M}_v$ so that a video x^+ relevant *w.r.t.* a given query shall have a larger similarity than an irrelevant video x^-, *i.e.* $s(x^+, q) > s(x^-, q)$. Such a constraint is commonly implemented by minimizing a triplet ranking loss ℓ_{tri} with hard-negative mining [7, 10, 13, 31, 36]. Given a caption q and a video x^+ it is describing, let $x^\#$ be the hardest negative video, practically selected from a given mini-batch. The loss ℓ_{tri} is calculated as

$$\begin{cases} x^\# & = \arg\max_{x^-}(s(x^-, q) - s(x^+, q)) \\ \ell_{tri}(q, x^+, x^\#) & = \max(0, m_0 + s(x^\#, q) - s(x^+, q)), \end{cases} \tag{2}$$

where m_0 is a positive hyper-parameter controlling the margin.

4.2 Negation Learning

Given (x^+, q) as paired video and caption, we can effectively augment the training data by using the negator described in Section 3.1.1 to generate a soft-negative caption q^- *w.r.t.* the video x. Intuitively, we shall have $sim(x^+, q) > sim(x, q^-)$. So a relatively straightforward strategy to perform negation learning (NL) is to compute another ℓ_{tri} for the triplet $< x^+, q, q^- >$,

$$\ell_{tri}(x^+, q, q^-) = \max(0, m_1 + s(x^+, q^-) - s(x^+, q)), \tag{3}$$

where m_1 is a margin parameter. Adding this auxiliary loss to the primary loss in Eq. 2, we obtain a joint loss $\ell_{snl}(q, x^+)$ as

$$\ell_{snl}(q, x^+) = \ell_{tri}(q, x^+, x^\#) + \lambda_1 \ell_{tri}(x^+, q, q^-), \tag{4}$$

with λ_1 as a weight. The model $\mathcal{M}$ is trained by minimizing $\ell_{snl}(q, x^+)$. We term this strategy *Simple Negation Learning* (SNL).

We see from Eq. 3 that SNL treats q^- as a common negative caption *w.r.t.* the video. Recall that q^- is derived from q by negating one of its clauses. The unchanged part inherited from q, *e.g. a man is playing the guitar* as illustrated in Fig. 2, remains semantically relevant to the video content. This means the negative pair (x^+, q^-) shall maintain certain similarity. In other words, when viewing x^+ as a pivot point in the common space where q shall be more close to x^+, q^- shall not be pushed too far away from x^+. To that end, while $s(x^+, q)$ shall be larger than $s(x^+, q^-)$, there needs to be an upper boundary on their difference, *i.e.* $s(x^+, q) - s(x^+, q^-) < m_2$, with $m_1 < m_2 < 2$. We modify Eq. 3 to take the new constraint into account, resulting in a *bidirectionally* constrained loss ℓ_{bcl} as

$$\begin{aligned} \ell_{bcl}(x^+, q, q^-) = \quad & \max(0, m_1 + s(x^+, q^-) - s(x^+, q)) + \\ & \max(0, -m_2 - s(x^+, q^-) + s(x^+, q)). \end{aligned} \tag{5}$$

Similarly, given the original caption q, we expect that its cross-modal similarity to its relevant video x^+ shall be larger than its uni-modal similarity to its negated variant q^-. Meanwhile, the gap between $s(x^+, q)$ and $s(q, q^-)$ shall be bounded. To that end, we compute ℓ_{bcl} for the triplet $< q, x^+, q^- >$ as

$$\begin{aligned} \ell_{bcl}(q, x^+, q^-) = \quad & \max(0, m_3 + s(q, q^-) - s(q, x^+)) + \\ & \max(0, -m_4 - s(q, q^-) + s(q, x^+)), \end{aligned} \tag{6}$$

where m_3 and m_4 are margin parameters with $0 < m_3 < m_4 < 2$.

Note that $\ell_{bcl}(x^+, q, q^-)$ and $\ell_{bcl}(q, x^+, q^-)$ respectively use the video x^+ and the original caption q as a pivot in the common space to exploit the negated information. By jointly minimizing the two losses, the model $\mathcal{M}$ is trained to find a proper embedding for the soft negative q^- *w.r.t.* both the video x and the original caption q. Accordingly, we term the improved strategy *Bidirectional Negation Learning* (BNL), with the corresponding loss defined as

$$\ell_{bnl}(q, x^+) = \ell_{tri}(q, x^+, x^\#) + \lambda_2(\ell_{bcl}(x^+, q, q^-) + \ell_{bcl}(q, x^+, q^-)), \tag{7}$$

where λ_2 is a small positive weight for balancing the primary and the auxiliary losses.

4.3 Choice of the T2VR Model

We instantiate $\mathcal{M}$ with CLIP (ViT-B/32) [28]. Originally developed for text-image matching, CLIP consists of a BERT for text embedding and a Vision Transformer (ViT) for image embedding. To deal with the video input, we use ViT to extract features per frame, and aggregate the frame-level features to the video level by mean pooling[4] for subsequent cross-modal similarity learning and matching. We use CLIP-*snl* and CLIP-*bnl* to indicate CLIP trained with ℓ_{snl} and ℓ_{bnl}, respectively.

Note that our NL methods are model-agnostic, so other end-to-end alternatives to CLIP can in principle be used. We leave this for future exploration.

5 EXPERIMENTS

5.1 Implementation Details

Hyperparameters used in this work are empirically set as follows and fixed throughout our experiments, unless stated otherwise. The

[4] Mean pooling can be replaced by attention-based pooling for better performance [21].

Table 2: Performance on the original, negated and composed query sets of the re-purposed MSR-VTT3k. The boolean operation is not applicable to the original queries. Our CLIP-*bnl* tops the performance on the composed query set, while being sensitive on the negated query set.

Models	Original (↑)				Negated (↑)				Composed (↑)			
	*R*1	*R*5	*R*10	*MIR*	Δ*R*1	Δ*R*5	Δ*R*10	Δ*MIR*	*R*1	*R*5	*R*10	*MIR*
W2VV++ [19]	11.4	29.9	40.7	0.208	0.3	0.4	0.3	0.003	6.6	23.0	33.6	0.154
SEA [20]	12.4	32.0	43.4	0.224	0.1	0.3	0.3	0.002	7.5	24.3	34.9	0.164
CLIP [28]	21.2	40.8	50.2	0.309	1.5	2.5	2.9	0.020	6.9	24.2	35.6	0.160
CLIP* (*this paper*)	27.7	53.0	64.2	0.398	0.5	1.1	1.1	0.008	11.4	33.3	46.2	0.225
CLIP4Clip [25]	**28.9**	**54.4**	**65.1**	**0.410**	0.8	1.5	1.2	0.010	11.3	33.3	45.6	0.222
Boolean operation:												
W2VV++	–	–	–	–	10.5	26.1	34.6	0.182	8.9	23.7	32.5	0.166
SEA	–	–	–	–	11.9	29.1	38.2	0.202	7.5	19.8	27.9	0.142
CLIP	–	–	–	–	18.8	37.5	46.2	0.278	5.9	16.7	23.9	0.118
CLIP*	–	–	–	–	25.3	47.1	56.1	0.353	13.5	33.7	45.5	0.236
CLIP4Clip	–	–	–	–	**27.2**	**51.0**	**59.9**	**0.380**	8.0	22.9	32.0	0.158
CLIP-*bnl* (*this paper*)	28.4	53.7	64.6	0.404	5.0	6.9	6.9	0.057	**15.3**	**40.0**	**53.3**	**0.274**

Table 3: Performance on the re-purposed MSR-VTT1k.

Models	Original (↑)				Negated (↑)				Composed (↑)			
	*R*1	*R*5	*R*10	*MIR*	Δ*R*1	Δ*R*5	Δ*R*10	Δ*MIR*	*R*1	*R*5	*R*10	*MIR*
W2VV++	24.7	50.4	62.2	0.371	1.2	-0.5	0.8	0.007	10.7	32.9	46.0	0.218
SEA	27.2	54.3	65.8	0.398	-0.5	-0.9	-1.6	-0.007	12.2	34.6	47.0	0.232
CLIP	31.6	54.2	64.2	0.422	1.4	1.4	1.5	0.017	12.9	35.0	46.2	0.237
CLIP*	41.1	69.8	79.9	0.543	0.0	1.7	1.0	0.006	17.3	46.8	61.2	0.310
CLIP4Clip	**43.9**	**70.6**	**80.2**	**0.560**	1.2	-1.7	0.0	0.008	15.0	43.1	57.8	0.281
Boolean operation:												
W2VV++	–	–	–	–	21.3	40.9	51.2	0.310	11.2	27.9	36.9	0.196
SEA	–	–	–	–	23.9	47.6	54.1	0.344	10.9	26.6	35.6	0.188
CLIP	–	–	–	–	26.4	46.2	56.8	0.354	6.3	18.4	25.9	0.129
CLIP*	–	–	–	–	35.9	59.5	65.2	0.463	17.6	42.0	52.0	0.291
CLIP4Clip	–	–	–	–	**40.0**	**61.9**	**69.1**	**0.495**	8.5	25.6	34.9	0.171
CLIP-*bnl*	42.1	68.4	79.6	0.546	12.2	11.7	14.4	0.121	**24.8**	**57.6**	**68.8**	**0.391**

margin parameter m_0 for the primary retrieval loss (Eq. 2) is set to 0.2 according to VSE++ [10]. The lower and upper boundaries, *i.e.* m_1 and m_2 for $\ell_{bcl}(x^+, q, q^-)$ (Eq. 5) are set to 0.1 and 0.6, while m_3 and m_4 for $\ell_{bcl}(q, x^+, q^-)$ (Eq. 6) are set to 0.1 and 0.3. The weight λ_1 for SNL and λ_2 for BNL are both set to 1e-3.

Our deep learning environment is PyTorch (1.7.0) [27] plus NVIDIA GEFORCE RTX 3090 GPUs. We perform SGD based training, with RMSProp as the optimizer. The learning rate is initially 1e-6, decayed by a factor of 0.99 per epoch. We use an early stopping strategy which stops training when no validation performance increase is achieved in two consecutive epochs.

5.2 Evaluating Current T2VR Models

5.2.1 T2VR Model Selection. We choose the following models that have PyTorch training code publicly available:

• W2VV++[5], MM19 [19]: This model learns to project text and video into a latent space, by using bag-of-words (bow), w2v and GRU as its text encoders and ResNeXt-101 / ResNet-152 pre-trained on ImageNet as its visual encoders.
• SEA[6], TMM21 [20]: SEA exploits four text encoders (bow, w2v, GRU, BERT) in a multi-space similarity learning framework. Its visual encoders are the same as W2VV++.
• CLIP[7], ICML21 [28]: A text-image matching model pre-trained on 400 million image-text pairs collected from the Internet.
• CLIP*: We fine-tune the pre-trained CLIP on each of the three training sets (Table 1), using the retrieval loss as expressed in Eq. 2.

[5]https://github.com/li-xirong/w2vvpp
[6]https://github.com/li-xirong/sea
[7]https://github.com/openai/CLIP

Table 4: Performance on the re-purposed VATEX.

Models	Original (↑)				Negated (↑)				Composed (↑)			
	*R*1	*R*5	*R*10	*MIR*	Δ*R*1	Δ*R*5	Δ*R*10	Δ*MIR*	*R*1	*R*5	*R*10	*MIR*
W2VV++	40.5	76.2	84.6	0.561	0.4	0.9	0.1	0.003	12.4	33.7	46.2	0.233
SEA	41.8	78.5	87.0	0.578	-0.3	1.0	0.3	0.000	12.5	34.3	47.5	0.238
CLIP	41.4	72.9	82.7	0.555	1.9	2.1	2.2	0.018	10.5	28.3	41.3	0.201
CLIP*	56.8	88.4	**94.4**	0.703	0.2	0.4	0.7	0.004	14.2	39.2	53.3	0.266
CLIP4Clip	**61.5**	**88.8**	94.0	**0.734**	0.8	0.3	0.6	0.006	14.3	38.4	51.5	0.263
Boolean operation:												
W2VV++	–	–	–	–	31.5	57.1	61.6	0.421	11.6	31.6	42.1	0.215
SEA	–	–	–	–	33.1	60.3	65.2	0.446	12.0	29.7	39.5	0.209
CLIP	–	–	–	–	32.5	57.2	64.5	0.431	5.0	18.0	25.6	0.116
CLIP*	–	–	–	–	45.7	69.6	71.9	0.554	14.1	34.4	45.1	0.243
CLIP4Clip	–	–	–	–	**52.6**	**72.7**	**75.8**	**0.609**	8.7	25.8	34.3	0.171
CLIP-*bnl*	57.6	88.3	94.0	0.708	14.0	11.7	8.6	0.125	**16.6**	**39.9**	**53.9**	**0.284**

• CLIP4Clip[8], arxiv21 [25]: An end-to-end model which transfers the knowledge of the CLIP model for T2VR.

Among them, W2VV++ and SEA are feature re-learning based, while the others are all end-to-end.

5.2.2 Results. The performance of the selected T2VR models on the re-purposed MSR-VTT3k, MSR-VTT1k and VATEX is shown in Table 2, 3, and 4, respectively. The CLIP series (CLIP / CLIP* / CLIP4Clip), due to their end-to-end learning ability, clearly outperform the two feature re-learning alternatives (W2VV++ / SEA) on the original query set. However, the performance difference between the CLIP series and feature re-learning on the negated query set is much smaller, suggesting that they are insensitive to the negation. Note that the pre-trained CLIP has the relatively largest Δ*MIR* of 0.020, see Row#3 in Table 2. Similar results can also be observed from Table 3 and 4. Recall that different from CLIP, all the other models have been re-trained on the video-description data. These results are consistent with our earlier observation that the video descriptions lack negation. Learning from such data makes the models even more insensitive to negation in queries.

Our CLIP-*bnl* performs relatively lower than the SOTA model (CLIP4Clip) on the original query set. Note that we have no intention to beat the SOTA. The inclusion of the SOTA in our experiments is mainly to answer the question raised in the beginning, *i.e.* , *do the (current) models understand what the users do not want*? The experiments show that despite its leading performance on the original queries, CLIP4Clip is unaware of negation in queries, as demonstrated by its small and on the negated query set and its lower R1/R5/R10/MIR scores on the composed query set. Consider the MIR metric for instance, CLIP-*bnl* outperforms CLIP4Clip with a clear margin: 0.222 → 0.274 on MSR-VTT3k (23.4% relative improvement), 0.281 → 0.391 on MSR-VTT1k (39.1%) and 0.263 → 0.284 on VATEX (8.0%). Such performance gaps are deemed to be significant in the literature of T2VR.

[8]https://github.com/ArrowLuo/CLIP4Clip

5.3 NL versus Alternatives

5.3.1 Baselines. We implement the boolean operation [30] for each of the T2VR models previously evaluated. The operation requires decomposing a given query into a positive subquery and a negative subquery. Handling a composed query is relatively straightforward, as we know which part of the query is positive and which is negative. For a negated query, although the negation cue is known, the negation scope followed by the cue is unknown. We resort to neg-BERT [18] to automatically detect the negation scope. The detected result is used as the negative subquery, while the remaining part of the query is used as the positive subquery.

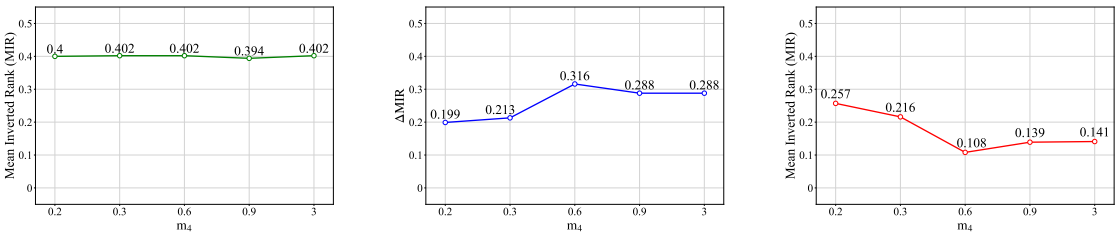
5.3.2 Results. As shown in Table 2, 3 and 4, the T2VR models with the boolean operation produces much larger response on the negated query set, when compared to their counterparts w/o the operation. This indicates that the boolean operation makes the models more sensitive to negation. However, the higher sensitivity is obtained at the cost of the undesired performance drop on the composed query set. Consider CLIP4Clip, the top performer on the original query set for instance. With the boolean operation, its MIR score on the composed query decreases: 0.222 → 0.158 on MSR-VTT3k, 0.281 → 0.171 on MSR-VTT1k, and 0.263 → 0.171 on VATEX. We conclude that the boolean operation is not effective for dealing with the negation in composed queries. By contrast, our CLIP-*bnl*, obtained by fine-tuning CLIP with the proposed bidirectional negation learning, shows superior performance on the composed query set. In addition, CLIP-*bnl* also exhibits higher sensitivity on the negative set against its NL-free counterpart, *i.e.* CLIP* (Δ*MIR* 0.057 versus 0.008, 0.121 versus 0.006, and 0.125 versus 0.004 on MSR-VTT3k, MSRV-VTT1k, and VATEX, respectively).

5.4 Ablation Study on NL

In order to verify the necessity of BNL against SNL, we try with varied implementation choices of the auxiliary loss. Per choice, the related margin parameters are set based on the performance on the held-out validation set. The ablation study is conducted on MSR-VTT3k.

Table 5: The influence of the auxiliary loss. Each row is the performance of a specific CLIP model trained by weightedly adding the corresponding auxiliary loss to the primary retrieval loss. Dataset: MSR-VTT3k.

Auxiliary loss	Original (↑)				Negated (↑)				Composed (↑)			
	$R1$	$R5$	$R10$	MIR	$\Delta R1$	$\Delta R5$	$\Delta R10$	ΔMIR	$R1$	$R5$	$R10$	MIR
– (Row#4, Table 2)	27.7	53.0	64.2	0.398	0.5	1.1	1.1	0.008	11.4	33.3	46.2	0.225
Simple Negation Learning (SNL):												
$\ell_{tri}(x^+, q, q^-)$	**28.8**	**54.2**	**65.1**	**0.408**	1.8	2.5	2.5	0.021	12.9	35.1	48.1	0.241
$\ell_{tri}(q, x^+, q^-)$	28.1	53.5	64.6	0.402	**21.4**	**38.8**	**44.5**	**0.288**	7.4	19.4	27.4	0.141
Bidirectional Negation Learning (BNL):												
$\ell_{bcl}(x^+, q, q^-)$	27.5	52.7	63.8	0.395	2.7	3.7	3.6	0.030	14.6	38.9	51.5	0.264
$\ell_{bcl}(q, x^+, q^-)$	28.0	53.3	64.2	0.400	15.8	26.0	28.1	0.199	14.8	36.1	50.4	0.257
$\ell_{bcl}(x^+, q, q^-) + \ell_{bcl}(q, x^+, q^-)$	28.4	53.7	64.6	0.404	5.0	6.9	6.9	0.057	**15.3**	**40.1**	**53.3**	**0.274**



(a) Original query set (b) Negated query set (c) Composed query set

Figure 4: Performance curves *w.r.t.* the upper boundary m_4 in $\ell_{bcl}(q, x^+, q^-)$. The auxiliary loss, with its lower boundary m_3 fixed as 0.1. Dataset: MSR-VTT3k.

5.4.1 SNL or BNL?. As Table 5 shows, the best choice for SNL is $\ell_{tri}(x^+, q, q^-)$ (with the video as a pivot), scoring MIR of 0.408 on the original query set and 0.241 on the composed query set. The best choice of BNL is the joint use of $\ell_{bcl}(x^+, q, q^-)$ and $\ell_{bcl}(q, x^+, q^-)$, scoring MIR of 0.404 on the original and 0.274 on the composed. With both are better than the baseline w/o using any auxiliary loss, BNL is better than SNL in terms of the overall performance.

Interestingly, when comparing Row#2 and Row#3 in Table 5, using the original query q as the pivot for loss computation yields a model that is clearly more sensitive to the negation than that using the video x^+ as the pivot (ΔMIR 0.288 versus 0.021). A similar phenomenon can also be observed in BNL (ΔMIR 0.199 versus 0.030). Recall that the similarity between the query (video) pivot and the soft negative q^- is computed in a text-to-text (video-to-text) manner. Hence, the results suggest that negation learning by text-to-text matching easily pushes the soft negative far away and adversely affects the learned common space. For this reason, we see that adding the upper-boundary constraint is important, increasing MIR on the composed set from 0.141 to 0.257 (Row#3 versus Row#5 in Table 5).

5.4.2 Influence of the Upper Boundary in $\ell_{bcl}(q, x^+, q^-)$. Following the above discussion about the upper boundary m_4, we further study its influence when using $\ell_{bcl}(q, x^+, q^-)$ as the auxiliary loss. As the performance curves in Fig. 4 show, while its impact on the original query set seems to be marginal, lowering its value obtains better performance on the composed query set. The result again justifies the necessity of BNL.

6 CONCLUSIONS

To conquer the novel task of negation in text-to-video retrieval (nT2VR), we propose a new evaluation protocol together with a learning based method for negation-aware T2V model training. Our experiments on two re-purposed datasets, *i.e.* MSR-VTT and VATEX, allow us to draw conclusions as follows. For the existing T2VR models evaluated in this paper, *i.e.* W2VV++, SEA, CLIP and CLIP4Clip, they are all found to be unaware of negation in queries. Also, manipulating their retrieval results by boolean operations does not work. Negation learning by text-to-text matching easily pushes a soft negative description far away from its original description and adversely affects the learned common space. Bidirectional negation learning is thus necessary. Re-training the CLIP model by the proposed learning method clearly improves its ability to handle queries with negation. In addition, its performance on the original benchmarks is also improved. We believe this work opens up new possibilities for multimedia retrieval.

Acknowledgments. This work was supported by NSFC (No. 62172420, No. 62072463), BJNSF (No. 4202033), and Public Computing Cloud, Renmin University of China.

REFERENCES

[1] George Awad, Asad A. Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas Diduch, Jeffrey Liu, Yvette Graham, Gareth J. F. Jones, , and Georges Quénot. 2021. Evaluating Multiple Video Understanding and Retrieval Tasks at TRECVID 2021. In *TRECVID Workshop*.

[2] Aozhu Chen, Fan Hu, Zihan Wang, Fangming Zhou, and Xirong Li. 2021. What Matters for Ad-hoc Video Search? A Large-scale Evaluation on TRECVID. In *ICCV Workshop on ViRal*.

[3] David Chen and William Dolan. 2011. Collecting Highly Parallel Data for Paraphrase Evaluation. In *CVPR*.

[4] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. 2020. Fine-Grained Video-Text Retrieval With Hierarchical Graph Reasoning. In *CVPR*.

[5] Matthew Cooper, John Adcock, Robert Chen, and Hanning Zhou. 2005. FXPAL at TRECVID 2005. In *TRECVID Workshop*.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[7] Jianfeng Dong, Xirong Li, Chaoxi Xu, Xun Yang, Gang Yang, and Xun Wang. 2021. Dual Encoding for Video Retrieval by Text. *TPAMI* 44, 8 (2021), 4065–4080.

[8] Maksim Dzabraev, Maksim Kalashnikov, Stepan Komkov, and Aleksandr Petiushko. 2021. MDMMT: Multidomain Multimodal Transformer for Video Retrieval. In *CVPR Workshop on HVU*.

[9] Allyson Ettinger. 2020. What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. *TACL* 8 (2020), 34–48.

[10] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In *BMVC*.

[11] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. 2021. CLIP2Video: Mastering Video-Text Retrieval via Image CLIP. *arXiv preprint arXiv:2106.11097* (2021).

[12] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-Modal Transformer for Video Retrieval. In *ECCV*.

[13] Ning Han, Jingjing Chen, Guangyi Xiao, Hao Zhang, Yawen Zeng, and Hao Chen. 2021. Fine-grained Cross-modal Alignment Network for Text-Video Retrieval. In *ACMMM*.

[14] Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by Understanding Not: Modeling Negation in Language Models. In *NAACL*.

[15] Fan Hu, Aozhu Chen, Ziyue Wang, Fangming Zhou, Jianfeng Dong, and Xirong Li. 2022. Lightweight Attentional Feature Fusion: A New Baseline for Text-to-Video Retrieval. In *ECCV*.

[16] Andrej Karpathy and Li Fei-Fei. 2017. Deep Visual-Semantic Alignments for Generating Image Descriptions. *TPAMI* 39, 4 (2017), 664–676.

[17] Nora Kassner and Hinrich Schütze. 2020. Negated and Misprimed Probes for Pretrained Language Models: Birds Can Talk, But Cannot Fly. In *ACL*.

[18] Aditya Khandelwal and Suraj T. Sawant. 2020. NegBERT: A Transfer Learning Approach for Negation Detection and Scope Resolution. In *LREC*.

[19] Xirong Li, Chaoxi Xu, Gang Yang, Zhineng Chen, and Jianfeng Dong. 2019. W2VV++: Fully Deep Learning for Ad-hoc Video Search. In *ACMMM*.

[20] Xirong Li, Fangming Zhou, Chaoxi Xu, Jiaqi Ji, and Gang Yang. 2021. SEA: Sentence Encoder Assembly for Video Retrieval by Textual Queries. *TMM* 23 (2021), 4351–4362.

[21] Xirong Li, Yang Zhou, Jie Wang, Hailan Lin, Jianchun Zhao, Dayong Ding, Weihong Yu, and Youxin Chen. 2021. Multi-Modal Multi-Instance Learning for Retinal Disease Recognition. In *ACMMM*.

[22] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*.

[23] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. 2019. Use What You Have: Video Retrieval Using Representations from Collaborative Experts. In *BMVC*.

[24] Jakub Lokoč, Tomáš Souček, Patrik Veselý, František Mejzlík, Jiaqi Ji, Chaoxi Xu, and Xirong Li. 2020. A W2VV++ Case Study with Automated and Interactive Text-to-Video Retrieval. In *ACMMM*.

[25] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2021. CLIP4Clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860* (2021).

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

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *ICML*.

[29] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. VATEX: A Large-Scale, High-Quality Multilingual Dataset for Video-and-Language Research. In *ICCV*.

[30] Jiaxin Wu and Chong-Wah Ngo. 2020. Interpretable Embedding for Ad-Hoc Video Search. In *ACMMM*.

[31] Peng Wu, Xiangteng He, Mingqian Tang, Yiliang Lv, and Jing Liu. 2021. HANet: Hierarchical Alignment Networks for Video-Text Retrieval. In *ACMMM*.

[32] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video And Language. In *CVPR*.

[33] Xun Yang, Jianfeng Dong, Yixin Cao, Xun Wang, Meng Wang, and Tat-Seng Chua. 2020. Tree-Augmented Cross-Modal Encoding for Complex-Query Video Retrieval. In *SIGIR*.

[34] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL* 2 (2014), 67–78.

[35] Youngjae Yu, Jongseok Kim, and Gunhee Kim. 2018. A Joint Sequence Fusion Model for Video Question Answering and Retrieval. In *ECCV*.

[36] Pengpeng Zeng, Lianli Gao, Xinyu Lyu, Shuaiqi Jing, and Jingkuan Song. 2021. Conceptual and Syntactical Cross-modal Alignment with Cross-level Consistency for Image-Text Matching. In *ACMMM*.

A APPENDICES

A.1 Rules for Query Construction

Negated Query Construction. Table 6 shows example rules to generate negated queries.

Table 6: Rule-based negated query construction.

Original Query	**Identified Word** (POS)	**Negated Query Sample**
Some guys are driving a car and **met** an accident in a road	met (VBD)	Some guys are driving a car and did not meet an accident in a road
A cartoon alien character **finds** another character	finds (VBZ)	A cartoon alien character does not find another character
A man is **running** around and playing a guitar	running (VBG)	A man is not running around and playing a guitar
A man **is** running around and playing a guitar	is (AUX)	A man isn't running around and playing a guitar
A father and son **are** playing with each others' hair	are (AUX)	A father and son aren't playing with each others' hair
A live concert **with** a woman as the lead singer	with (ADP)	A live concert without a woman as the lead singer

Composed Query Generation. We use templates to generate composed queries. According to whether the pronoun of subject can be determined, we use two sets of templates. For example, given < kids, do A, do B>, we first get the pronoun of 'kids', and then randomly choose one of following queries:

- Kids do A and they don't do B.
- Kids don't do B and they do A.
- Kids doing A and not doing B.
- Kids not doing B and they doing A.
- Kids are doing A and not doing B.
- Kids are not doing B and they are doing A.

When the subject is in third-person singular and gender cannot be determined, the pronoun of the subject is unknown. We then use a set of slightly different templates. For example, given <A kid, do A, do B>, we randomly choose one of the following queries:

- A kid does A and doesn't do B.
- A kid doesn't do B while does A.
- A kid doing A and not doing B.
- A kid not doing B while doing A.
- A kid is doing A and not doing B.
- A kid is not doing B while doing A.

Table 7: Influence of the auxiliary-loss weight λ_1 on SNL. Each row corresponds to a specific CLIP model trained by weightedly adding the corresponding auxiliary loss to the primary retrieval loss. Dataset: MSR-VTT3k.

Auxiliary loss	λ_1	Original (↑)				Negated (↑)				Composed (↑)			
		R1	*R5*	*R10*	*MIR*	Δ*R1*	Δ*R5*	Δ*R10*	Δ*MIR*	*R1*	*R5*	*R10*	*MIR*
– (Row#4, Table 2)	–	27.7	53.0	64.2	0.398	0.5	1.1	1.1	0.008	11.4	33.3	46.2	0.225
$\ell_{tri}(x^+, q, q^-)$	0.001	**28.8**	**54.2**	**65.1**	**0.408**	1.8	2.5	2.5	0.021	**12.9**	**35.1**	**48.1**	**0.241**
	0.01	28.5	53.8	64.7	0.405	11.3	16.7	17.5	0.135	10.5	30.6	44.0	0.209
$\ell_{tri}(q, x^+, q^-)$	0.001	28.1	53.5	64.6	0.402	21.4	38.8	44.5	0.288	7.4	19.4	27.4	0.141
	0.01	27.9	53.3	64.3	0.399	**25.3**	**47.5**	**55.9**	**0.352**	3.2	10.2	15.8	0.075

Table 8: Influence of the upper boundary (m_2 / m_4) and the auxiliary-loss weight λ_2 on BNL. Dataset: MSR-VTT3k.

Auxiliary loss	Upper boundary	λ_2	Original (↑)				Negated (↑)				Composed (↑)			
			R1	*R5*	*R10*	*MIR*	Δ*R1*	Δ*R5*	Δ*R10*	Δ*MIR*	*R1*	*R5*	*R10*	*MIR*
– (Row#4, Table 2)	–	–	27.7	53.0	64.2	0.398	0.5	1.1	1.1	0.008	11.4	33.3	46.2	0.225
$\ell_{bcl}(x^+, q, q^-)$	m_2													
	0.2	0.001	28.2	53.6	**64.6**	0.403	0.9	1.3	1.3	0.011	12.1	35.2	47.9	0.236
	0.2	0.01	27.8	53.0	64.0	0.397	3.3	4.6	4.6	0.038	11.3	30.9	43.1	0.214
	0.3	0.001	27.8	53.1	64.3	0.398	4.3	5.6	5.8	0.048	13.9	37.3	50.2	0.256
	0.3	0.01	28.4	53.7	**64.6**	0.404	6.5	9.0	9.2	0.074	12.7	32.3	44.6	0.228
	0.6	0.001	27.5	52.7	63.8	0.395	2.7	3.7	3.6	0.030	14.6	**38.9**	**51.5**	**0.264**
	0.6	0.01	**28.6**	**53.7**	64.5	**0.405**	9.5	13.9	14.4	0.112	12.4	33.7	46.8	0.232
	0.9	0.001	27.4	52.9	64.0	0.395	3.5	5.3	5.1	0.041	14.1	38.8	52.0	0.262
	0.9	0.01	28.0	53.4	**64.6**	0.400	11.7	18.0	19.3	0.143	10.4	30.8	43.5	0.210
$\ell_{bcl}(q, x^+, q^-)$	m_4													
	0.2	0.001	28.0	53.3	64.2	0.400	15.8	26.0	28.1	0.199	**14.8**	36.1	50.4	0.257
	0.2	0.01	28.2	53.5	**64.6**	0.402	11.2	16.9	17.3	0.133	10.0	29.4	40.6	0.198
	0.3	0.001	28.3	53.5	64.3	0.402	17.0	27.5	30.0	0.213	11.8	30.6	42.8	0.216
	0.3	0.01	28.3	53.4	64.2	0.402	15.5	24.5	26.1	0.191	7.4	23.7	33.6	0.159
	0.6	0.001	28.3	53.5	64.3	0.402	23.1	42.6	49.7	0.316	5.6	14.3	19.8	0.108
	0.6	0.01	28.4	53.4	64.2	0.402	**26.6**	**49.3**	58.1	**0.368**	1.0	4.6	8.8	0.039
	0.9	0.001	27.2	52.9	64.0	0.394	21.3	38.7	44.4	0.288	7.1	19.5	28.3	0.139
	0.9	0.01	28.4	53.6	**64.6**	0.404	26.4	49.1	**58.2**	0.367	1.1	4.5	8.0	0.039

A.2 Hyper-parameter Evaluation

Influence of the auxiliary-loss weight λ_1 on SNL. As Table 7 shows, setting the auxiliary-loss weight λ_1 as 0.001 yields better performance on both original and composed query set. Setting λ_1 as 0.01 make model more sensitive to negation but at the cost of undesired performance drop on the two other query sets.

Influence of the upper boundary (m_2 / m_4) and the auxiliary-loss weight λ_2 on BNL. As Table 8 shows, upper boundary and auxiliary learning weight jointly influence negation learning task. Generally, with looser upper boundary and larger auxiliary-loss weight, model exhibits more sensitiveness to negation, but their performance not necessarily increases on the composed and original query sets. Their influence over original query set is smaller than composed query set. Using $\ell_{bcl}(x^+, q, q^-)$ alone (with the video as a pivot), top performance on composed query set is achieved by setting m_2 as 0.6 and λ_2 as 0.001. Using $\ell_{bcl}(q, x^+, q^-)$ alone (with the original query as a pivot), top performance on composed query set is achieved by setting m_4 as 0.2 and λ_2 as 0.001. The result suggests that setting auxiliary-loss weight as 0.001 is more appropriate, meanwhile using the query pivot requires a tighter upper boundary than using the video pivot.

A.3 NL for Text-to-Image Retrieval (T2IR)

To see to what extent can our findings be generalized to the image domain, we reproduce our research on Flickr30k [34] and MS-COCO [22], with results shown in Table 9 and 10. We simply adopt the same hyper-parameters as we have used for T2VR, which could be suboptimal for T2IR. Again, we observe that CLIP-*bnl* outperforms the baselines on the composed query set with a clear margin, showing the viability of the proposed negation learning method. Although our paper is targeted at T2VR, the proposed negative learning method also works for T2IR with negation.

Table 9: Results on re-purposed Flickr30k (data split: [34]).

Models	Original (↑)				Negated (↑)				Composed (↑)			
	R1	*R5*	*R10*	*MIR*	Δ*R1*	Δ*R5*	Δ*R10*	Δ*MIR*	*R1*	*R5*	*R10*	*MIR*
CLIP	59.0	84.6	91.0	0.705	2.7	2.0	1.2	0.024	18.9	41.2	56.0	0.303
CLIP*	**75.1**	**93.3**	**96.1**	**0.832**	1.2	0.4	0.2	0.009	23.4	50.9	65.4	0.365
CLIP (boolean)	–	–	–	–	52.3	69.4	71.5	0.594	8.5	20.1	28.3	0.150
CLIP*(boolean)	–	–	–	–	**65.2**	**72.2**	**70.2**	**0.677**	16.1	41.2	51.6	0.278
CLIP-*bnl*	74.8	93.1	96.2	0.829	18.5	10.7	6.8	0.149	**26.4**	**55.7**	**70.8**	**0.398**

Table 10: Results on re-purposed MS-COCO (data split: [16]).

Models	Original				Negated				Composed			
	R1	*R5*	*R10*	*MIR*	Δ*R1*	Δ*R5*	Δ*R10*	Δ*MIR*	*R1*	*R5*	*R10*	*MIR*
CLIP	28.8	54.1	65.0	0.408	1.6	2.6	2.3	0.020	11.8	28.9	40.6	0.210
CLIP*	**45.6**	**72.8**	**82.3**	**0.579**	3.2	1.9	1.7	0.025	14.0	38.2	52.7	0.261
CLIP (boolean)	–	–	–	–	24.3	43.4	50.1	0.330	7.0	17.7	24.9	0.127
CLIP*(boolean)	–	–	–	–	**39.5**	**57.1**	**61.6**	**0.467**	12.8	32.0	42.5	0.222
CLIP-*bnl*	44.7	71.8	81.4	0.570	18.0	18.7	16.1	0.176	**17.3**	**43.0**	**56.1**	**0.298**