NAIST-IS-DD1561005

Doctoral Dissertation

Linking Videos and Languages: Representations and Their Applications

Mayu Otani

March 15, 2018

Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Mayu Otani

Thesis Committee:

Professor Kiyoshi Kiyokawa	(Supervisor)
Professor Hirokazu Kato	(Co-supervisor)
President Naokazu Yokoya	(Nara Institute of Science and Technology)
Professor Janne Heikkilä	(University of Oulu)
Professor Tomokazu Sato	(Shiga University)
Associate Professor Yuta Nakashima	(Osaka University)

Linking Videos and Languages: Representations and Their Applications[*]

Mayu Otani

Abstract

Mimicking the human ability to understand visual data (images or videos) is a long-standing goal of computer vision. To achieve visual content understanding by a computer, many recent works attempt to connect visual and natural language data including object labels and descriptions. This attempt is important not only for visual understanding but also for broad applications such as content-based visual data retrieval and automatic description generation to help visually impaired people.

While connecting visual data with natural language, *e.g.*, predicting labels of visual concepts and captioning, was once challenging, the recent growth of computational resources and vision-language datasets has led to a significant improvement in these tasks. In particular, deep neural network-based approaches has become able to annotate captions accurately for static images. However, as scales of video datasets are so far limited, and videos are more computationally expensive, understanding visual content in videos still remains challenging.

The goal of this dissertation is to develop cross-modal representations, which enable us to associate videos with natural language. We explore two directions for constructing cross-modal representations: hand-crafted representations and data-driven representation learning. The experiments demonstrate the proposed representations can be applied to a wide range of practical tasks including query-focused video summarization and content-based video retrieval with natural language queries.

[*] Doctoral Dissertation, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1561005, March 15, 2018.

Chapter 1 describes the background of research for connecting videos and languages and detail the goal of this dissertation. In Chapter 2, we summarize prior attempt for cross modal-representation learning and their applications. Chapter 3 introduces a hand-crafted representation that encodes objects in videos and noun words in sentences. We also define semantic similarity between the object-based representation. The object-based representation is applied to video summarization based on user text. In Chapter 4, cross-modal representation learning is explored. We introduce deep models that map videos and sentences to a common feature space, where semantically relevant videos and sentences are located in a nearby space. Different from the object-based representation, this approach incorporates various concepts including objects, actions, and attributes. The performance of the learned representation is evaluated on several tasks: unsupervised video summarization, content-based video and sentence retrieval, and video captioning. The models are further improved by exploiting web images, which help to disambiguate the semantics of sentences. Chapter 5 describes a method to learn frame-level representations for videos. This model is designed to capture dynamics of content within a video. The main challenge of training the model is the lack of video-sentence datasets with frame-level annotations. We alleviate the lack of training data by synthesizing training examples from existing video-description datasets. We apply this frame-level representation to a task of content-based video retrieval for multi-clip videos, which we call fine-grained video retrieval. The experimental results demonstrate that our cross-modal representation is useful for content-based video retrieval with a natural language query.

Keywords:

representation learning, cross-modality, neural network, video retrieval, video summarization

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Once humans take a brief look at visual data (images or videos), they can easily and quickly list various concepts in the image and describe the visual content with natural language. Mimicking this human ability, *i.e.*, understanding and describing visual content, in a computer is a key technique for various applications such as content-based image or video retrieval and automatically describing visual content to help visually impaired people understand the visual content.

One approach for connecting visual and natural language data is to design a cross-modal embedding space. Figure 1.1 illustrates the idea of cross-modal embedding space. Both images and texts are represented as points in a common space so that those with similar semantics are located at close points. For example, an image of zebras in the field, as well as a sentence "a flock of zebras grazing" should be mapped to nearby points.

One straightforward approach to constructing cross-modal representation is to use visual concept recognition techniques. We can obtain a list of visual concepts from an image by visual concept classification or detection techniques. As visual concepts are often described with nouns or verbs in natural language, we can compute the semantic similarity between text and visual data by matching words in a text and detected concept labels. Based on this assumption, some work represents visual data by a set of concept labels and natural language data by a set of words. This cross-modal representation is often used as an intermediate representation in description generation methods [14, 22].

While extensive research efforts had been made so far for designing cross-

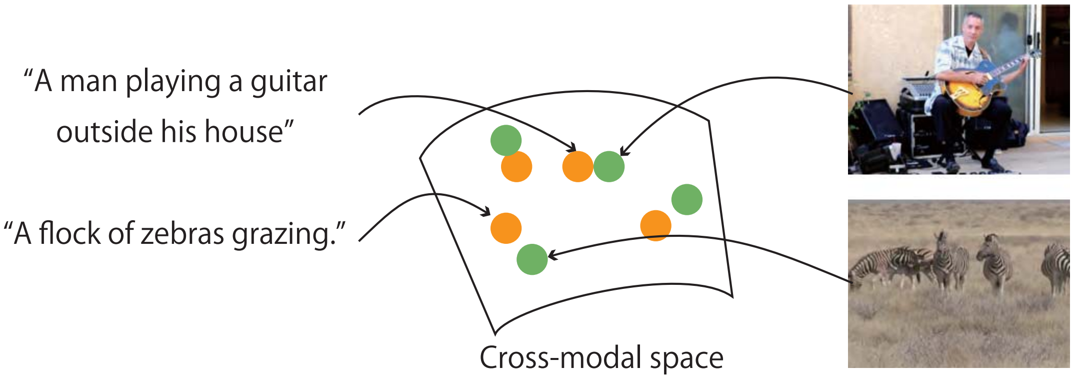

"A man playing a guitar outside his house"
"A flock of zebras grazing."
Cross-modal space

Figure 1.1. Illustration of cross-modal embedding space.

modal representations, it was difficult to connect visual and natural language data because of the limited performance of visual concept recognition models. Visual concept recognition has been a challenging task, although it looks quite simple. Inferring visual concepts of an image involves finding patterns that might be relevant to the visual concepts. Since humans are not aware of how they find and generalize the visual patterns, implementing how to recognize visual concepts in a computer has been quite difficult.

The recent emergence of large-scale datasets and deep neural networks (DNNs) facilitate visual concept recognition. In computer vision, convolutional neural network-based classification models have shown substantial improvement in visual concept recognition tasks including object recognition [42, 80, 26] and action recognition [35, 33]. Now, the state-of-the-art models can even distinguish thousands of visual concepts [71].

An important advantage of the DNN-based approach is the integration of the whole process involved in a task in a deep model, which can be trained in an end-to-end manner. For example, previous object recognition involves several processes: low-level feature extraction, feature transformation, and classification. They are separately designed, and tuning low-level processing (*e.g.*, low-level feature extraction) to maximize the performance of final output is hardly feasible. On the other hand, DNN-based approach integrates these process into one deep model, and the whole process can be optimized by fitting to large-scale data. Given pairs of input and correct output, deep models can learn useful features and how to use them.

This also accelerated the research for tasks that involve different modalities, such as image or video captioning [6, 107]. Recent works propose to associate embeddings of visual data and natural languages with DNNs. In end-to-end learning, how to extract features from different modalities, as well as how to fuse them can be learned seamlessly. This leads the improvement in learning cross-modal embedding spaces for visions and language [39, 108]. In DNN-based approach, one may not need to extract concepts from images or natural language explicitly. Instead, one will model how to map images and text to a common space.

The goal of this dissertation is to develop cross-modal representations for videos and natural languages. The representation should capture complex semantics, and their similarity should follow human intuition on semantic similarity. Most works in this direction have tried to connect static images and short phrases or sets of keywords [39, 34], and videos and natural language have still significant room to explore. Different from static images, videos have additional challenges to capture semantics because they have temporal changes. Due to the temporal changes, the semantics of videos are more complex. This complexity of content makes modeling video understanding more difficult. Similar challenge exists in natural language understanding. Since a sentence in natural language may include various words, and the semantics of each word highly depends on context, modeling the semantics of sentences is also difficult.

As we mentioned above, techniques to connect videos and text have some practical applications. To evaluate the performance of our cross-modal representations, we will apply them to several tasks, which involve videos and natural languages, such as query-focused video summarization, video captioning, and content-based video retrieval. By showing the results of these applications, we will investigate the capability of our cross-modal embedding space.

This dissertation is organized as follows. First, we discuss related works for connecting videos and languages in Chapter 2. We introduce object-focused representations for videos and paragraphs in Chapter 3. In this work, we develop video summarization based on user text, which uses the proposed representations and the similarity metric for creating video summaries based on the input text.

In Chapter 4, we address joint representation learning for videos and sentences. We also propose sentence embedding method that exploits a web image search engine, which helps to disambiguate semantics of sentences. Learned cross-modal embedding spaces are evaluated on the task of content-based video retrieval. We also show that the learned embedding space is beneficial for unsupervised video summarization. In Chapter 5, we address encoding a video into a sequential vector representation, instead of a single vector representation. We demonstrate a content-based video retrieval application, which finds video parts relevant to a natural language query. Finally, Chapter 6 summarizes this dissertation and remaining challenges. We also discuss a future path of this work.

Chapter 2

Related Work and Contributions

The work in this dissertation is motivated by many previous works that address to link vision and language modalities. This chapter gives the overview of existing cross-modal representations and applications that involve visual and language data.

1. Cross-modal Representations for Videos and Languages

Some early works proposed to use a set of concept labels as cross-modal representations for static images and text [14, 51]. Farhadi *et al.* [14] introduced triplets of concept labels (object, action, and scene) as representations, which represent the abstract semantics of images and sentences. For videos, the approach by Lin *et al.* [51] associates a parsed semantic graph of a query sentence and visual cues based on object detection and tracking.

These works require explicit concept detection to construct representations. Therefore, they cannot handle images or text with unseen concepts. To achieve more flexible representations, some works propose to develop a common embedding space, in which visual and language data can be mapped [16, 82, 36]. This approach enables us to compute the semantic similarity between images and text based on the distance in the embedding space without explicit concept detectors. For example, Socher *et al.* [82] proposed to embed low-level image representations and word vectors of object labels into a common embedding space with neural network-based models. They demonstrated classification of unseen visual concepts in the embedding space, which is called as zero-shot learning.

The recent success of deep convolutional neural networks (CNNs) together with large-scale visual datasets [69, 6, 75] has led to several powerful models for image understanding [10, 103, 111]. These models showed not only significant improvement in object classification, but also highly generalized visual representations obtained from hidden layers of the deep models [10]. Deep neural networks have also been used in the field of natural language processing [45, 39]. These works demonstrated that neural network-based models are capable of encoding semantics of text. For example, Kiros *et al.* [39] proposed sentence representation learning using recurrent neural networks (RNNs). They also demonstrated joint learning of image and sentence embedding models, which convert images and sentences to cross-modal representations.

Cross-modal representation learning using deep neural networks is explored in many tasks [52, 16, 36, 108, 117]. Frome *et al.* [16] proposed image classification by computing similarity between joint representations of images and labels, and Zhu *et al.* [117] addressed alignment of movie scenes with sentences in a book using joint representations for video clips and sentences. Their approach also computes the similarity between sentences and subtitles of video clips to improve the performance of video-sentence alignment.

2. Applications

Video Summarization

In this dissertation, we develop video summarization methods as applications of the proposed representations in Chapters 3 and 4. Video summarization is a technique to generate a compact representation of long videos, which help users quickly understand the content. Various methods for video summarization have been proposed for different domains, including sports videos [4], movies [76, 43], documentaries [89], e-sports [83] and user videos [24]. There are also surveys of the literature available [94, 59, 56]. Video summaries are categorized into two types, *i.e.*, storyboards and video skimming. Storyboards are static video summaries, which consist of keyframes [96, 20, 27]. Dynamic video skimming involves generating a short video, which consists of excerpts from the original videos [63, 110]. Our work in this dissertation falls into the category of dynamic video skimming.

To automatically select video excerpts from input videos, various ideas to asses the importance of video clips have been proposed. Attractiveness is a widely employed selection criterion, representing how well a clip attracts the attention of the audience. Attractiveness is often computed from dynamics of pixel values in early video summarization works. Specifically, temporal changes in representations (*e.g.*, the Hessian of pixel values [43] or coordinates of blobs [9]) are tracked, and clips with significant changes are selected [43, 64]. Rather than using low-level representations, more sophisticated saliency-based approaches have also been employed [54, 12]. For example, Ma *et al.* [54] proposed the combination of visual and audio saliency, with which highly salient clips are selected for a video summary. Gygli *et al.*[24] proposed a combination of multiple properties, such as saliency, aesthetics, and camera motion, which are then used to select attractive video clips.

Another criterion for video clip selection is representativeness; video clips in a summary should be less redundant but cover most of the original content. Many methods have been proposed to find representative yet diverse clips [9, 115, 25, 19, 108]. One major approach to retrieving representative clips is video-clip clustering. Earlier work by Gong *et al.* [20] proposed clustering of the video frames in the input videos using singular value decomposition. With their method, clips that include frames close to the cluster centers are included in the video summary. Gygli *et al.* [25] cast the selection of representative clips as a k-medoids problem, which can be efficiently optimized due to its sub-modularity. Zhao *et al.* [115] proposed an online video summarization method. Their method generates a video summary by picking out video clips that are able to reconstruct the remaining clips. For diverse clip selection, determinantal point processes are attracting the attention of video summarization research [19, 108]. As this approach is often used for unsupervised video summarization, we also employ representativeness criterion in Chapter 4 to demonstrate how our representation can benefit a basic video summarization method.

Attractiveness and representativeness are general criteria that can be applied to videos in any domain. However, an ideal video summary can depend much on the domain of the video. Several works of Video summarization for a specific domain have demonstrated the effectiveness of domain knowledge. For example, a work for American football video summarization detects replay shots and scoring events [4], and egocentric video summarization uses a set of daily object detectors [53, 47]. Hu *et al.* [29] demonstrated character-based summarization for TV series using a speaker-identification method. Potapov *et al.* [68] proposed a supervised video summarization method that takes account of event categories such as “birthday party.” Given a training dataset of videos and associated event categories, event classifiers are trained to predict the importance of a clip based on its event category. Lee *et al.* [47] proposed a method for learning the importance of objects in egocentric videos.

Another interesting research direction in video summarization is text-focused summarization, which controls the content of video summaries using textual cues. For example, Babaguchi *et al.* [4] incorporated user profiles that reveal the user’s favorite teams, players, and events. Sharghi *et al.* [78] proposed to extract video clips based on the relevance to keywords. Research in this line attempts to associate video content and text, such as scripts and query words, to generate a video summary based on the input text. In Chapter 3, we propose an object-based representation for videos and text for text-focused video summarization.

Content-based Video and Language Retrieval

Due to the explosive growth of images and videos on the web, visual retrieval has become a hot topic in computer vision and machine learning [8, 56, 46, 49]. Early work addressed content-based video retrieval by detecting predefined concepts in videos, such as objects, actions, and events [81, 102]. A single visual concept may not be enough to spot the desired video, so users are more likely to query with their combinations. Video retrieval by natural language queries provides an intuitive way to make a combination of concepts in a specific context represented in a query. One possible approach is to detect visual concepts and match them to keywords extracted from a natural language query [105, 44, 97, 51], but as they require pre-trained concept detectors, such as [26, 116, 93], unseen or rare concepts might be missed.

To overcome such limitations, Socher *et al.* [82] proposed to learn to embed images and concept labels into a common space, which can handle unseen concepts. Several approaches in this direction have been proposed on both image retrieval [16, 39] and video retrieval [117, 108, 65]. Xu *et al.* [108] proposed a deep neural network for video retrieval by sentence queries and vise versa. They embed a video clip and a sentence into a common space to compute the similarity between them. Yu *et al.*'s approach [113] learns a similarity metric between a whole video content and a query sentence. In contrast to these methods, we address to estimate the relevance that may vary within a video in Chapter 5.

A similar task has been studied in the community of action localization, which finds when a certain action occurs in a video [40, 17, 31, 104]. While action localization focuses on human actions, our video retrieval tasks in Chapter 5 use a natural language sentence, which describes various concepts including actions, as a query. Therefore, the action localization can be regarded as a special case of video retrieval that localizes content in a video.

3. Video and Natural Language Datasets

To promote research to associate videos and natural language data, the research community has provided various datasets involving videos and natural language data, such as descriptions [77, 72, 106, 5, 114], video titles [114, 84], and visual concept labels [47, 70, 1]. Chen *et al.* [5] provide 1,967 short YouTube video clips capturing a single activity. Each video in this dataset is annotated with descriptions. Xu *et al.* [106] released a larger-scale YouTube video dataset, which contains 10K video clips collected with a video search engine and natural language descriptions annotated by crowdsource workers.

There are several datasets for specific domains. Movie datasets aligned with descriptions are introduced in [72, 73, 91, 55]. Senina *et al.* [77] collected cooking video clips and their descriptions. Zeng *et al.* collected 18K user-generated videos and their titles [114]. The averaged length of the videos in this dataset is approximately 1.5 minutes (longer than most other datasets), and they are not edited.

The movie datasets in [72, 77] have alignment of description and video frames. Their vocabulary and content in video clips are fairly different from other videos, such as online videos or broadcast programs, since the movies include fantasy, sci-fi, etc. Several works collect videos by asking crowdsource workers to capture videos according to given scenarios [79, 21]. Compared to web videos, videos in these datasets are well controlled for a task of activity recognition and more suitable for learning-based approaches.

4. Contributions

In this dissertation, we develop cross-modal representations for videos and languages. There are two different approaches to develop cross-modal representations. One is manually designing representations, and the other is a learning-based approach which automatically learns cross-modal representation extractors from data. We investigate hand-crafted cross-modal representations in Chapter 3 and representation learning in Chapters 4 and 5. For each cross-modal representations, we develop applications and evaluate the effects of our cross-modal representations.

In Chapter 3, we develop object-focused representations for videos and text. This representation is designed to capture the high-level semantics of events by summarizing objects in video clips and text. We define semantic similarity between video and text based on this representation. Using this representation, we develop a video summarization method, which generates video summaries according to user text. Video summarization is done by selecting a subset of videos which maximizes the semantic similarity to user text.

The hand-crafted representation in Chapter 3 is limited to a fixed list of object categories, and do not consider other concepts including actions, or attributes. In Chapter 4, we address representation learning for videos and sentences that can handle larger vocabulary. To handle various concepts, we employed deep models that map videos and sentences to a common space. This approach does not require explicit concept recognition. Our approach in Chapter 4 is close to the work by Xu *et al.* [108]. They represent a sentence by a subject, verb, and object (SVO) triplet, and embed sentences as well as videos to a common embedding space using deep neural networks. The main difference between ours and the work [108] is the use of an RNN to encode a sentence. The use of an RNN enables our model to encode all words in a sentence and capture details, such as an object's attributes and scenes. We demonstrate that this representation may improve unsupervised video summarization. We also show that deep representations sometimes fail to capture the semantics of sentences. This failure comes from the ambiguity of words and limited capability of sentence encoding to model long dependency. We propose to help sentence encoding by using web images relevant to a sentence. We evaluate the effect of web images on the task of content-based video retrieval. We also demonstrate that our training scheme using web images benefits video encoding model. We train video captioning model that uses our representations and show that our cross-modal representation can efficiently encode semantics of videos.

We expand the representation learning method to produce sequential representations for videos in Chapter 5. This sequential output is useful to capture how semantic changes within a video. We thus develop a fine-grained video retrieval method that finds parts of a video relevant to a query sentence. This study in Chapter 5 is closely related to the works by Tapaswi *et al.* [87] and Zhu *et al.* [117], which aim to align book text and movie scenes, as well as query-focused video summarization by Sharghi *et al.* [78]. Both methods search for a part of a long video using a natural language query. The main difference between ours and these works is that ours have fewer assumptions about target videos and queries. To align book chapters or sentences to movie scenes, these previous approaches [87, 117] assume that the movie comes with closed captions and that the book text and the movie follow a similar timeline. Sharghi *et al.*'s approach [78] only uses a limited set of nouns as queries and does not accept more generic queries, *i.e.*natural language queries. Our task has neither rich metadata of videos nor rough temporal locations.

The contributions of this dissertation are summarized as follows:

- We propose several cross-modal representations for videos and sentences. Two different approaches to cross-modal representations are explored in this work. Specifically, we introduce hand-crafted representations based on the occurrence of objects in videos and text. We also propose representation learning method for videos and sentences. The proposed models are capable of encoding various concepts including objects, actions, and scenes.

- The cross-modal representations are applied to practical applications. First, we develop video summarization using user text, which creates a video summary according to the content of user text. Our object-focused representations are used to compute semantic similarity between videos and text. We also demonstrate that our deep representations are helpful in the task of content-based video retrieval.

Chapter 3

Object-based Representations for Summarizing Personal Videos Using Blog Text

1. Overview

This chapter proposes object-based representations to capture the semantics of videos and text. We assume that objects in a video clip provide rich cues to understand events in a video, and nouns in text also tell key concepts of text's content as well. Based on this assumption, we construct an object-based representation that encodes objects in videos and nouns extracted from text. We also define a similarity metric with this object-based representation, which enables us to compute the semantic similarity between a video and text.

As an application of our object-based representation, we develop a video summarization method for personal videos. Video summarization extracts a subset of video clips from a long video to produce a shorter video. To select video clips for a video summary, most existing methods rely on predefined criteria, such as small-redundancy [115] or attractiveness of visual content [54, 12]. Therefore, each of them offers the same summary, provided that the same video clips are given as the input.

However, users often have some ideas about storylines that they want to express in their videos. For example, video blogs, which consist of user-generated videos and supporting text, are edited to express author's ideas, experiences, etc. Hereafter, we refer to these ideas as the author's "intentions." Different blog authors need different video summaries, even when they have the same set of original video clips. Most existing video summarization methods do not take such intentions into account. Some methods can modify a video summary to each user based on user preferences [4] or observations about users (*e.g.*, brain waves) [2]. Nevertheless, these methods do not offer explicit authorial control over the content of a video summary.

One approach to enable users to control output video summary is text-based video summarization [78, 67]. Given textual cues, such as keywords or descriptions, text-based video summarization collects video clips relevant to the textual cues. In this work, as an application of our object-based representations, we develop a video summarization method that edits a long video according to scripts written by a user. Specifically, our video summarization system takes a text written for a video blog post and unedited videos as input and produces a video summary that has semantically relevant content to the blog post. During video clip selection, we optimize the content similarity between a video summary and the blog post, which can be computed with our object-based representation.

The main contributions of this chapter are summarized as follows:

1. We develop an object-based representation for video clips and text. We define a similarity metric with our cross-modal representation. This similarity metric can be used for evaluating the content similarity between videos and text.

2. We present a video summarization method for video blogs. The proposed method produces video summaries according to textual input. This method is based on the assumption that good video summaries for video blogs reflect the author's intentions, which most previous works have not taken into account.

3. Most previous work uses low-level visual features for clip selection, whereas our method uses high-level cross-model representation: objects in video clips and words in an input text. By maximizing the content similarity

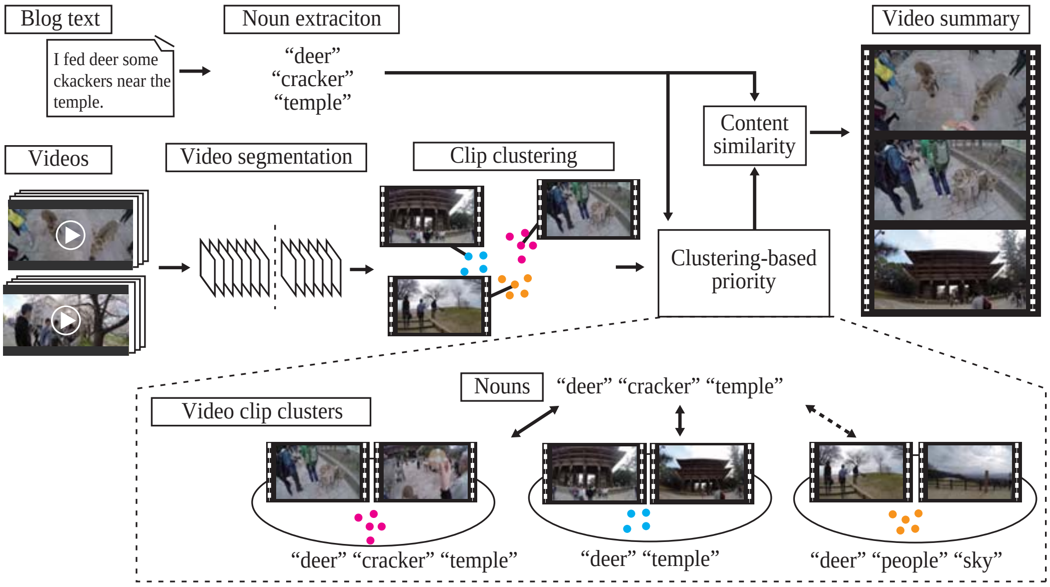


Figure 3.1. Overview of the proposed text-based video summarization method. Given text written by a user, our method selects video clips based on the content of the text, such that the video summary reflects the user's intentions.

based on our cross-modal representation, the proposed video summarization method can retrieve video clips that show events described in the text.

4. We conducted user studies to evaluate our video summarization method. Participants compose their video blog post and answer a questionnaire regarding the composition process. The results have shown that participants preferred our video summarization method for video blog authoring.

5. We also compared our video summarization method with several baseline methods regarding the content coverage and relevance to the input text. Experimental results suggest important properties of video summaries for video blog applications.

2. Text-based Video Summarization

Our video summarization method takes videos with timestamps and the text written by the blog author as input and generates a video summary. The problem of video summarization can be cast as a problem of selecting the optimal subset of video clips. In this study, we design an objective function based on the content similarity between a subset of clips and the input text. By selecting clips that have high content similarity to the input text, our method generates a video summary reflecting the blog author's intentions. Figure 3.1 illustrates an overview of our method. Our method first extracts nouns from the input text. The videos are then segmented and clustered into groups, each of which corresponds to an event. Based on these clusters, we compute the priority of clips; highly prioritized clips are more likely to be included in the video summary. After computing the priority, a video summary is produced by selecting the optimal subset of clips.

2.1 Object-based Representations for Videos and Text

Encoding Text Since objects in videos are often described with nouns, we extract nouns from the input text. The input text is represented by an N-dimensional vector $\mathbf{y}$, where N is a vocabulary size, and assume that noun n corresponds to object n. We set $y_n = 1$ if noun n is included in the input text and 0 otherwise. For noun extraction, we apply parts-of-speech tagging to the input text [92]. Furthermore, we remove predefined stop words because common words are hardly informative.

Encoding video clips We first perform video segmentation on lengthy input videos. Because our method selects clips based on their objects, we set clip boundaries where objects appear or disappear. To find these clip boundaries, we employ the method by Huang *et al.*[30]. Their method tracks the number of keypoint matches and identifies local minima. These local minima often correspond to frames around which objects appear or disappear. Thus, we divide the video at such frames.

Each video clip after video segmentation is represented by object labels and their importance. Object-detection methods, such as [18], can automatically find objects in the clips; however, to focus on our clip selection performance without focusing on the performance of the object-detection method, we manually annotate object labels in this study, rather than detecting them automatically. To do so, we extracted the middle frame of each clip as a keyframe and annotated

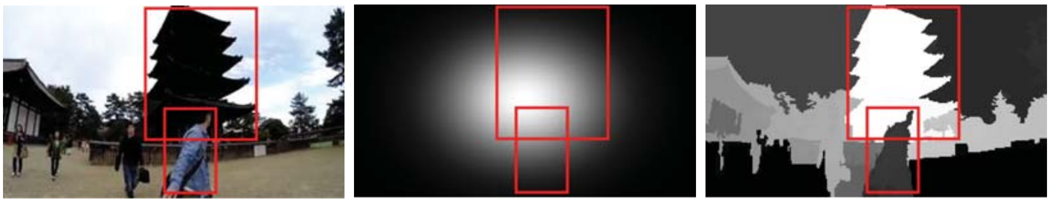


Figure 3.2. Maps of location-based object importance (a) and saliency-based object importance (b).

the object labels.

In this thesis, we test two types of object importance: location-based object importance, and saliency-based object importance. Location-based object importance is simply based on the location and the size of the bounding box of each object. The computation of location-based importance relies on some heuristics: *viz.*, (i) an important object is more likely to be located near the center of the frame, and (ii) it occupies a large area. Based on these heuristics, the importance $x_{m,n}$ of object n in a clip m is defined as

$$x_{m,n} = \int_{\omega \in \Omega_n} \mathcal{N}(\omega | \mu, \Sigma) \mathrm{d}\omega, \tag{3.1}$$

where Ω_n is object n's bounding box, and $\mathcal{N}$ is the normal distribution whose mean μ is the frame's center position and whose variance Σ is a predefined parameter.

The other type of object importance incorporates saliency maps. Because salient objects are likely to be visually important, we employ the average of saliency values over a bounding box as the saliency-based object importance. In this thesis, we use saliency maps based on Yan *et al.*'s method [109]. Saliency maps are computed based on local contrast values and center bias, *i.e.*, areas near the center of an image are more likely to be important. To get stable results, their method generates multiple image layers, which are coarse representations at different levels, and compute a saliency map for each layer. Saliency maps in different scales are fused to produce a final output. Note that the proposed method can use any other method to obtain saliency maps, such as [3, 66, 62], without significant modification. Figure 3.2 shows the maps of location-based object importance and saliency-based object importance, where brighter areas are regarded as more important.

After computing the importance, the input videos are represented by a set of clips $X = \{\mathbf{x}_m \in \mathrm{R}^N \mid m = 1, \ldots, M\}$, where $\mathbf{x}_m$ is a vector representation of the clip m. N is the number of object categories, and each element $x_{m,n}$ denotes the importance of object n in that clip.

2.2 Text-based Clip Selection

Let $\boldsymbol{\psi}(S)$ be a function that gives an N-dimensional vector representation of a subset of clips $S \subseteq X$, given by

$$\boldsymbol{\psi}(S) = \sum_{x_m \in S} p_m(\mathbf{y})\mathbf{x}_m, \tag{3.2}$$

where $p_m(\mathbf{y})$ and $\mathbf{x}_m$ denote a priority value of the clip m conditioned on the input text and an N-dimensional vector representation for clip m, respectively. The priority value represents how relevant the clip is to the input text, which is computed with cluster-based content similarity.

With the video summary representation, we formulate the problem of selecting a subset of clips $S^* \subseteq X$ as:

$$S^* = \operatorname*{argmax}_{S \subseteq X} O(\boldsymbol{\psi}(X), \mathbf{y}), \tag{3.3}$$

$$\text{s.t.} \sum_{\mathbf{x}_m \in S} l_m \leq L. \tag{3.4}$$

Here, L is the length of the resulting summary, which is given by the user, and l_m is the length of clip m. The objective function to be maximized in video summarization is a linear combination of two terms as follows:

$$O(\boldsymbol{\psi}(S), \mathbf{y}) = o_{\text{sim}}(\boldsymbol{\psi}(S), \mathbf{y}) + \alpha o_{\text{cov}}(\boldsymbol{\psi}(S)), \tag{3.5}$$

where o_{sim} is the content similarity between S and the input text $\mathbf{y}$, and o_{cov} is the content coverage. Moreover, α is a parameter that balances these two terms. Selecting a subset with high content similarity reflects the blog author's intentions in the resulting summary, and the content-coverage term encourages the summary to include various content, provided that it is relevant to the input text.

The following sections detail the clip priority, the content-similarity term, and the content-coverage term.

Clip Priority

Our method uses content similarity based on the objects in each video clip and the nouns in the input text. However, this can be unreliable, because the clip usually contains a subset of objects that appear in the event. For example, suppose the input text pertains to a certain event in which a certain object is involved. If this object is not very specific to the event, even though it appears throughout the input video, content similarity based solely on objects and nouns can pick out all clips that come with the object.

To find clips that are more relevant to the input text, we introduce clustering-based clip priority. First, we assume that an event is temporally concentrated, *i.e.*, clips capturing the same event have similar timestamps. Under this assumption, we can cluster clips based on their timestamps and the objects in them. For clustering, we use affinity propagation [15]. The similarity between two clips $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as

$$A(\mathbf{x}_i, \mathbf{x}_j) = \exp\left[-\frac{\lambda \min(|\tau_i - \tau_j|, \theta)}{M}\right] + \gamma J(\mathbf{x}_i, \mathbf{x}_j), \tag{3.6}$$

where τ_i is the temporal frame index of the middle frame in clip i, and M denotes the total number of frames in the input videos. Here, $J(\cdot,\cdot)$ gives the weighted Jaccard similarity, defined as

$$J(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_n \min(x_{i,n}, x_{j,n})}{\sum_n \max(x_{i,n}, x_{j,n})}. \tag{3.7}$$

In Eq. (3.6), λ controls the reduction in temporal similarity, and θ is a threshold for the temporal distance $|\tau_i - \tau_j|$. We suppose that clips extracted from different videos are temporally distinct. Thus, the temporal distance $|\tau_i - \tau_j|$ of such clips is set to a threshold θ. Moreover, γ is a parameter to balance the temporal similarity with the object based similarity. The number of clusters is automatically determined from data and self-similarity $A(\mathbf{x}_i, \mathbf{x}_j)$. Low self-similarity values result in a small number of clusters. We set the self-similarity values to the median of the pair-wise similarities as suggested in [15].

We assume that a cluster is relevant to the input text when the nouns corresponding to the objects in the cluster are included in the input text. Thus, we again use the weighted Jaccard similarity between a cluster and the input text to determine the priority of all clips in the cluster. Let $\mathbf{c}_i$ be a representation of the cluster that includes clip i, each element of which represents whether the corresponding object appears in the cluster. More specifically, we set $c_{i,n} = 1$ if any clip in the cluster has $x_{m,n} > 0$, and $c_{i,n} = 0$ otherwise. Using this, the priority value of clip i is computed as

$$p_i(\mathbf{y}) = J(\mathbf{c}_i, \mathbf{y}). \tag{3.8}$$

Content-similarity Term

We quantify the content similarity between the set S of clips and the input text y using the weighted Jaccard similarity in Eq. (3.7). This computes the similarity between object labels in selected videos and nouns in the input text as follows:

$$o_{\mathrm{sim}}(\mathbf{x}, \mathbf{y}) = J(\boldsymbol{\psi}(S), \mathbf{y}). \tag{3.9}$$

This similarity indirectly relies on priority through $\boldsymbol{\psi}(S)$. The value increases when S includes clips with high priority that have objects in common with the input text.

Content-coverage Term

If content coverage is not considered, some relevant clips can be rejected when their objects do not appear explicitly in the input text. This can result in a summary that is entirely composed of clips with similar content. To avoid this, our method encourages the inclusion of relevant clips that cover diverse content. Coverage of the original content is a criterion that is widely used in summarization tasks [89, 95, 86]. Figure 3.3 illustrates the idea of content-coverage in this study. Insofar as our goal is to generate a summary that reflects the blog au-

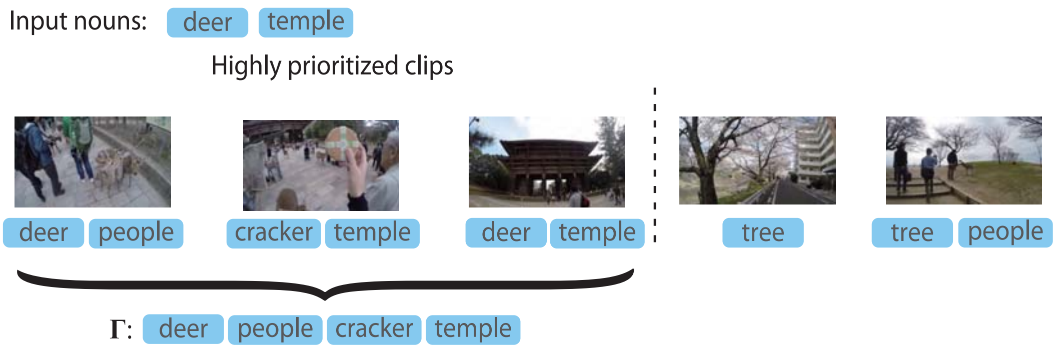


Figure 3.3. Illustration of content restriction for the content-coverage term.

thor's intentions, we list objects annotated to highly prioritized video clips. The coverage of the set of objects is rewarded during clip selection.

Let $\mathbf{\Gamma} = (\gamma_1, \ldots, \gamma_N)$ represent a set of objects in highly prioritized clips, where $\gamma_n = 1$ if clip $\mathbf{x}_i \in X$ whose $p_i > \rho$ has $x_{i,n} > 0$ and $\gamma_n = 0$ otherwise. We define the coverage $o_{\text{cov}}(\boldsymbol{\psi}(S))$ using the weighted Jaccard similarity in Eq. (3.7) to compute similarity between sets of objects in selected clips and prioritized ones as follows:

$$o_{\text{cov}}(\boldsymbol{\psi}(S)) = J(\boldsymbol{\psi}(S), \mathbf{\Gamma}). \tag{3.10}$$

This term represents how well S covers the content of highly prioritized clips.

2.3 Clip Selection

The proposed clip selection algorithm obtains a suboptimal subset of clips S^* in the manner of dynamic programming inspired by [57]. During subset selection, we iteratively update a video summary by adding a clip with the constraint of the summary length (Algorithm 1). Let $S^*_{m,l}$ be a subset of clips, which are selected such that their total length is limited to l. To obtain $S^*_{m,l}$, we evaluate the objective function in Eq. (3.5) with $\mathbf{x}_m \cup S_{m-1,l-l_m}$, where l_m is the length of clip $\mathbf{x}_m$. We then update $S^*_{m,l} = x_m \cup S_{m-1,l-l_m}$ if the value increases; otherwise $S^*_{m,l} = S_{m-1,l-l_m}$. We store each intermediate result and its corresponding value. To obtain a video summary, we select the best of the stored subsets and concatenate its clips in the order of their timestamps.

Algorithm 1 Clip selection

```
Require: video clips $X = \{\mathbf{x}_1, \dots, \mathbf{x}_M\}$, text $\mathbf{y}$, summary length $L$
Ensure: $S \subseteq X$
  $S_{i,0} = \emptyset \ \forall i = 1, \dots, M$
  for $m = 1$ to $M$ do
    for $l = 1$ to $L$ do
      $S' = S_{m-1,l}$
      $S'' = S_{m-1,l-l_m} \cup \mathbf{x}_m$
      if $O(\boldsymbol{\psi}(S'), y) > O(\boldsymbol{\psi}(S''), \mathbf{y})$ then
        $S_{m,l} = S'$
      else
        $S_{i,l} = S''$
      end if
    end for
  end for
  $S^* = \mathrm{argmax}_{S_{m,l}} O(\boldsymbol{\psi}(S_{m,l}), \mathbf{y})$
```

3. Evaluation and Discussion

Assessing the quality of video summaries is a challenging problem itself. Most previous methods are evaluated based on user studies [37, 53] or by comparing the resulting summaries with manually created reference summaries [68, 24, 50]. Since our task (*i.e.*, video summarization for video blogs) is a novel video summarization task, there is no established way to evaluate the performance of our method. Therefore, we opt to conduct a user study.

The present user study consists of two parts. First, a participant is asked to score multiple video summaries for a given blog post regarding their suitability to the blog post. To investigate our video summaries in detail, we administer an additional questionnaire regarding other properties, including redundancy, content coverage, and the relevance of the summary to the input text. Thus, we evaluate the video summaries from the perspective of the video blog viewers. The second part of the user study involves collecting blocks of text written by the participants and generating video summaries using the text. The participants are

T1	On a warm day in March, we went to Nara Park. Before getting to Nara Park, we went to Saho river. There were cherry trees along the river. The river is well known for cherry blossom, and many people visit during the season of blossom. I took many videos of other students. One of the students, Nakashima used a special camera for his study. He took some videos, carrying the camera along the river. It was a beautiful place and I want to visit there next spring again.
T2	We went to Nara Park. A lot of deer were around the Nandaimon. There were also a few cracker shops, and many tourists enjoyed feeding deer. I bought some crackers and deer immediately gathered around me.
T3	Nandaimon is a famous gate in the Nara Park. I saw a statue of Nandaimon. There were many people.

Figure 3.4. Original texts used in the experiment.

Table 3.1. Input and methods evaluated.

	Input	Method
(a)	Videos	Uniform sampling
(b)	Videos	Cluster-based
(c)	Videos and text	Proposed method
(d)	Videos and text	Description-based w/o content coverage
(e)	Videos and text	Description-based w/o content coverage and preference

asked to score the video summaries based on their text. Consequently, this part evaluates the video summaries from the perspective of the blog authors.

3.1 Evaluation from the Viewers' Perspective

Because this is the first attempt to use video summarization for video blogs, we investigated whether blog viewers believed that the video summaries generated by our method were suitable for a given blog post. We also evaluated video summaries in terms of several properties, such as redundancy and content coverage, which are widely used criteria in the domain of video summarization.

To compile a dataset, we recorded multiple videos of a short trip, totaling 80 min. As input text, we used the three blocks of text shown in Figure 3.4, each of which describes different scenes from the input videos. We compared our method

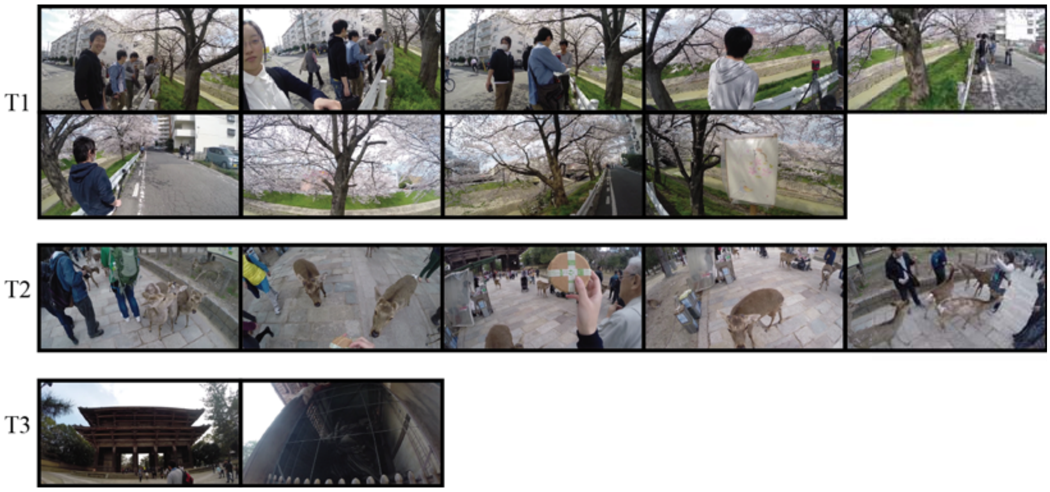
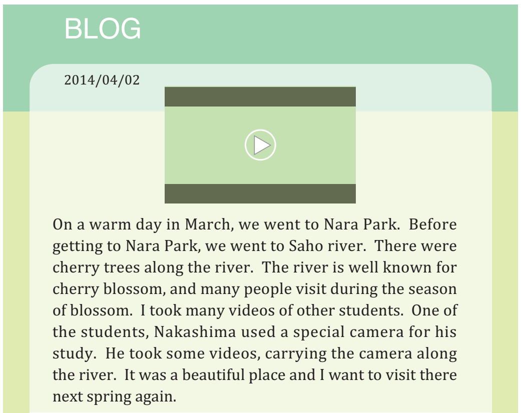


Figure 3.5. Keyframes of our video summaries for each input text.

to multiple baseline methods (see Table 3.1). Methods (a) and (b) generate video summaries without text. Uniform sampling (a) is a simple yet effective way to produce video summaries, and this method is widely employed as a baseline. We sampled 2-sec. clips with uniform intervals. The clustering-based method (b) utilizes the clustering results described in Section 2.2. With this method, clips are selected from cluster representatives, such that they include as many objects as possible. We also compared some variants of our method. Method (c) is our full method. Method (d) is basically our text-based method, but with the content coverage term o_{cov} excluded (*i.e.*, $\alpha = 0$). In addition to the exclusion of the coverage term o_{cov}, method (e) also excludes clip priority by setting the priority values of all clips to 1. All of these variants used location-based object importance.

For location-based object importance, the parameters were set to $\Sigma = \text{diag}(8w, 8h)$, where w and h are the width and the height of the frame, respectively. Other parameters were heuristically determined as follows: $\alpha = 0.25$, $\lambda = 5$, $\theta = 3600$, $\gamma = 0.25$, $\rho = 0.1$, and $L = 20$. Here, θ corresponds to 60 sec., because our input videos were 60 fps. We generated video summaries using methods (c)–(e) for each input text. In total, we generated 11 videos. Keyframes of the clips selected with our full method are shown in Figure 3.5. These resulting summaries show that our method selects clips from different scenes based on the content of the input



2014/04/02

On a warm day in March, we went to Nara Park. Before getting to Nara Park, we went to Saho river. There were cherry trees along the river. The river is well known for cherry blossom, and many people visit during the season of blossom. I took many videos of other students. One of the students, Nakashima used a special camera for his study. He took some videos, carrying the camera along the river. It was a beautiful place and I want to visit there next spring again.

Figure 3.6. An example of a video blog post shown to participants in the user study.

texts.

We recruited 20 participants from both genders; all participants were in their 20s or 30s. They reviewed a video blog post (see Figure 3.6) and were asked to score each video in terms of how well the video suited the blog post. The scores ranged from 1 to 5, where 1 means that the video definitely does not suit the blog post, and 5 means that it suits the post very well. The participants were divided into three groups. Group 1 (G1), Group 2 (G2), and Group 3 (G3) had eight, six, and six people, respectively. The blog post T1 was displayed for subjects in G1, blog post T2 for G2, and blog post T3 for G3. After reviewing a blog post, subjects rated baseline video summaries and description-based video summaries. Subjects also scored video summaries generated using blog posts for other groups.

Table 3.2 shows the scores for each group. For all groups, our full method (c) was scored as either the first or second best. Variant (d) was also rated highly. Interestingly, the participants in G1 chose clustering-based video summary (b) as most suitable for text T1. In fact, the clustering-based method (b) only accidentally included many clips relevant to T1, which contributed to the high score. Furthermore, we found that only the summary generated by the clustering-based method (b) included scenes just before the events described in T1. Although the inclusion of such clips was not part of the design of the clustering-based method, such clips can lead to a better comprehension of the events by providing context.

Table 3.2. Average scores regarding suitability to a video blog post. Bold values indicate the highest scores for each group.

method	Input text	Group		
		G1	G2	G3
(a) Uniform sampling	None	3.38	1.67	2.67
(b) Cluster-based	None	**4.38**	1.83	2.00
(c) Proposed method	T1	3.38	1.00	1.83
	T2	2.25	**4.33**	2.67
	T3	1.38	1.67	3.67
(d) Description-based w/o content coverage	T1	3.25	1.00	1.83
	T2	2.13	4.17	2.67
	T3	1.13	2.17	**4.00**
(e) Description-based w/o content coverage and preference	T1	2.25	3.00	2.50
	T2	2.00	3.17	3.00
	T3	2.13	2.67	3.17

The effect of such connecting video clips on video summaries is discussed in [53].

The results from comparing the scores among variants of our methods (c)–(e) imply that the content coverage term o_{cov} did not significantly affect the score. On the other hand, the use of clip priority resulted in a significant improvement in the suitability for the video blog. From these results, we conclude that the participants generally preferred our method over other methods. These results also suggest that the inclusion of clips that introduce scenes of interest can further improve the suitability for a blog post. The participants were also asked to score videos in terms of the following three aspects, to investigate the perception of our video summary compared to that of the baselines.

Q1 How well the video matches the input text (relevance to the input text).

Q2 How redundant the video is.

Q3 How well the summarized video covers the content of the entire video.

The scores ranged from 1 to 5. For Q1, a score of 1 means that the video does not represent the text at all, whereas 5 means that it represents the text very well. For

Table 3.3. Average scores of similarity to the input text (Q1). Bold values are the highest scores for each input text.

	Baselines		Our method								
	(a)	(b)	(c)			(d)			(e)		
Text	None		T1	T2	T3	T1	T2	T3	T1	T2	T3
T1	3.45	3.65	**3.90**	1.60	1.30	3.8	1.65	1.15	2.75	1.55	1.65
T2	1.70	1.85	1.10	**4.50**	1.75	1.20	4.45	1.70	2.35	3.70	3.40
T3	1.35	1.20	1.10	1.65	4.70	1.10	1.30	**4.75**	1.55	2.80	3.10

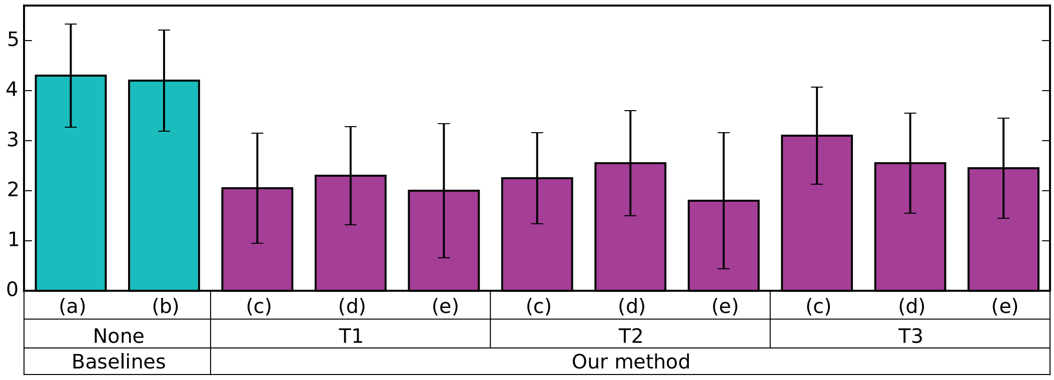


Figure 3.7. Averages and standard deviations of the scores for Q2.

Q2, scores 1 and 5 mean “significantly redundant” and “hardly redundant,” respectively. For Q3, score 1 means significant content is missing, whereas 5 means that most content is covered. The relevance to the input text is an important property for video summarization designed for video blogs. Because redundancy and content coverage is widely used in evaluations of video summaries, we also investigated these properties.

Table 3.3, Figure 3.7, and Figure 3.8 show the results for Q1, Q2, and Q3, respectively. Regarding Q1 (Table 3.3), our summaries received the highest scores in terms of their relevance to the input text.

This means that our method was able to select clips appropriate to the input text. On the other hand, in terms of redundancy (Q2) and content coverage (Q3), our method received lower scores than uniform sampling (a) and the clustering-based method (b). Because our video summaries have multiple clips relevant to

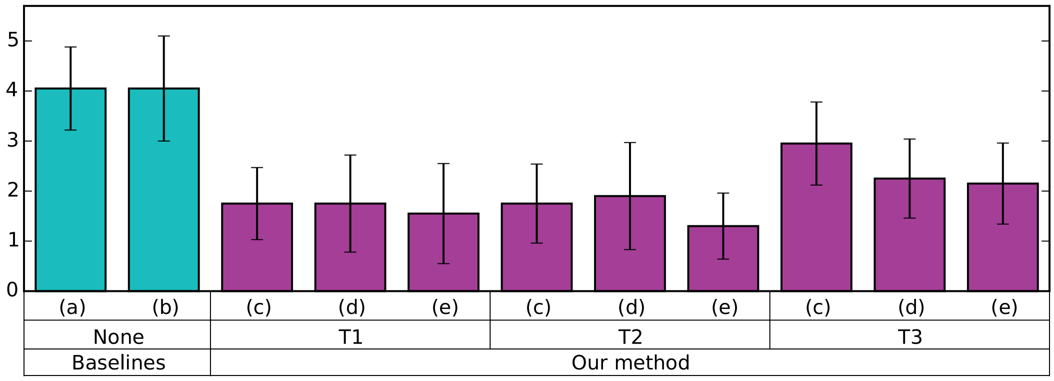


Figure 3.8. Averages and standard deviations of the scores for Q3.

the input texts, the clips can have similar content. This resulted in lower scores for Q2. The score for Q3 was also expected because our method restricts those clips that are included in the summary based on the input text. Although our method was not rated highly for Q2 (redundancy) and Q3 (content coverage), the participants preferred our video summaries for the video blog posts, according to Table 3.2. This indicates that, for blog viewers, the relevance to the input text is more important for video blogs than redundancy or content coverage.

3.2 Evaluation from the Blog Author's Perspective

We also collected texts written by 12 participants, all male and all in their 20s. We asked them to score the video summaries that were generated based on their texts. The participants reviewed all unedited videos in our dataset and wrote a short description of what interested them. By comparing participants' responses, we investigated how their intention was reflected in the video summaries.

The video dataset was the same as that of the previous section. In this evaluation, we compared four methods. Two were the same as the baselines (a) and (b) in the previous section. The other two methods were our proposed method, with location-based object importance and saliency-based object importance. The parameter ρ was set to the minimum of the priority for the top-90% of the clips.

For the first question, participants were asked to rate whether they would want to use the video summary generated by each method for their video blog post (Q4). Scores 1 and 5 indicate "strongly disagree" and "strongly agree,"

Table 3.4. Averages and standard deviations of scores for Q4.

	Our Methods		Baseline Methods	
	Location-based (c)	Saliency-based	Uniform (a)	Clustering-based (b)
Avg.	2.92	**3.50**	3.50	3.33
Std.	1.31	0.80	1.31	1.44

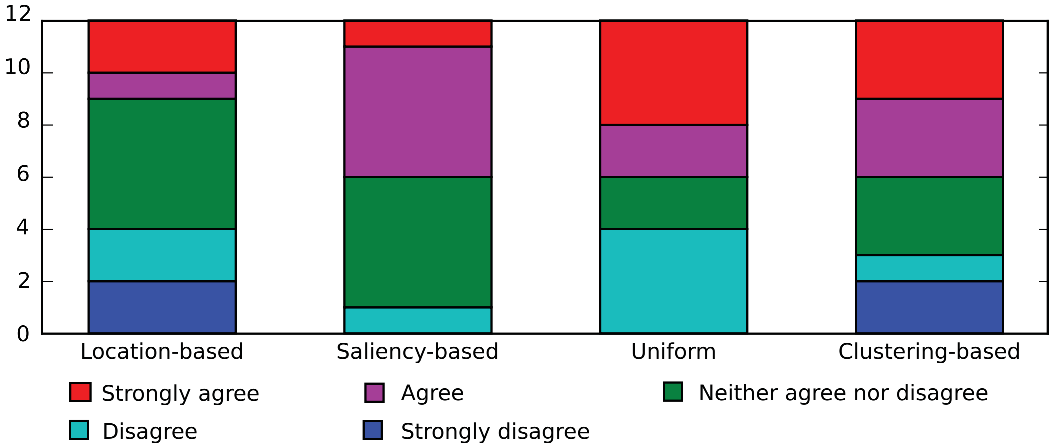


Figure 3.9. Answers for Q4.

respectively. Table 3.4 shows the average and the standard deviation of the scores. Whereas our method with saliency-based object importance and uniform sampling received the same average score, the standard deviation of our method was smaller. Figure 3.9 shows details of the results. This reveals that uniform sampling received both positive and negative responses, whereas only a few participants negatively rated our method with saliency-based object importance.

To identify the factors that affect these scores, we asked the participants to answer an additional questionnaire regarding their assessment of the following properties of video summaries:

- Relevance to the text in the blog post
- Inclusion of more scenes than the text
- Aesthetic quality (composition, camera motion, etc.).

The participants were asked whether these respective properties were important.

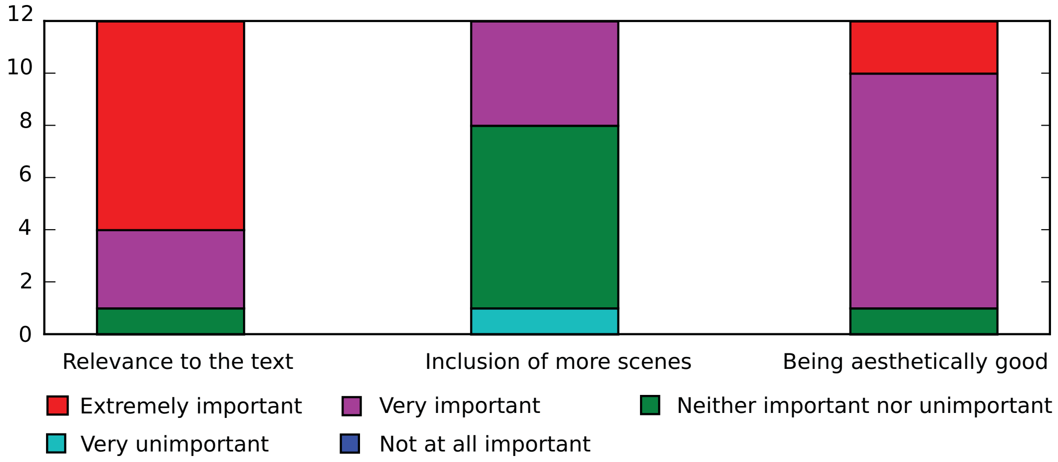


Figure 3.10. Answers regarding the importance of video-summary properties.

Figure 3.10 shows the results. The results show that many participants thought that the relevance to the blog post and the aesthetic quality were important for video summaries. We believe that this is the main reason why the participants preferred saliency-based object importance to location-based object importance.

3.3 Limitation

We proposed an object-based representation to connect paragraph and a set of videos. Since this approach represents videos and sentences by a pre-defined set of objects, our representation fail to capture unseen concepts. Therefore, our video summarization method may be not able to retrieve videos with rare concepts. It should be noted that our method is designed for personal videos that include several scenes or events. Thus this method may not be very effective for some videos, such as sports videos, whose object-based representation hardly change within videos. For example, a video capturing boxing match will show two people and a boxing ring throughout videos, and object-based representation will be hardly distinctive.

4. Summary

We proposed object-based cross-modal representations for videos and text and introduced the semantic similarity using the representations. We demonstrated a text-based video summarization method for video blog authoring as an application of our representation. The proposed video summarization method segments input videos and clusters clips, such that each cluster corresponds to an event. We further proposed clustering-based priority to indicate how relevant the clip is to the input text. We observed that this text-focused priority improves the suitability of video summaries to the input texts. The proposed method selects clips by maximizing the content similarity between the input text and a resulting video summary. The experimental results demonstrate the effectiveness of our method. Moreover, we examined the preferred properties of video summaries for video blogs, and the results show that the relevance to a blog post is considered paramount. The results also suggest that considering the aesthetic quality in addition to relevance to a blog post can further improve video summaries.

Although our method currently utilizes manually annotated object labels, various object-labeling methods have been proposed recently [80, 85, 26]. These methods can be adopted by our object-based representation. Also, our video summarization method can be extended by including clips that introduce and provide context for the scene of interest, and it can be improved by considering the aesthetic quality of the summary.

Chapter 4

Learning Semantic Representations by Linking Videos and Sentences

1. Overview

This chapter describes a representation learning method to associate videos and sentences. Different from the object-based representations described in Chapter 3, we address to incorporate various concepts including actions, scenes and attributes to compute cross-modal representations. Specifically, we construct deep models that convert a video clip and a sentence to vector representations in a common embedding space, where their semantic similarity correlates to their negative distance.

Our method is inspired by previous works by Xu *et al.* [108] and Kiros *et al.* [39]. Xu *et al.* propose a deep model that embeds video clips and a sentence into a common embedding space. While their work picks out the only subject, object, and verb words to represent the semantics of a sentence, we consider all words in a sentence to compute sentence embeddings. Kiros *et al.* demonstrate content-based image and sentence retrieval by mapping them to a common embedding space. We follow their training scheme to train video and sentence embedding models jointly.

Our cross-modal representations are validated on the task of video summarization and content-based video retrieval. In the first experiment, we investigate how our representations, which are trained to capture sentence-level semantics, affect the performance of video summarization. We develop a clustering-based video summarization method, which is a basic unsupervised approach for video summarization, and compare its performance with different video representations. Secondly, we demonstrate content-based video and sentence retrieval between video clips and sentences using our cross-modal representations. Our embedding models are further extended with web image search to disambiguate the semantics of sentences, which can be helpful for content-based video and sentence retrieval applications.

2. Cross-modal Representation Learning for Videos and Sentences

We propose a neural network-based embedding model to extract cross-modal representations for videos and sentences. Our embedding model consists of two subnetworks, each of which encodes videos and sentences, respectively (Figure 4.1). Moreover, we also propose to enhance sentence embedding by exploiting web images.

2.1 Video Embedding

We extract frames from a video at 1 fps as in [108] and feed them to a CNN-based video embedding model. Many recent works on modeling video frames employ different frame rate. For example, Yu *et al.*'s work sample one per ten frames (2.4 fps) to encode movie videos [113], and Na *et al.* employ 6 fps for a video question answering model [61]. We believe the effects of different frame rate is not significant since the details of motions are seldom important for capturing sentence-level semantics.

In our approach, we employ two CNN architectures: 19-layer VGG [80] and GoogLeNet [85], both of which are pre-trained on ImageNet [75]. We replace the classifier layer in each model with two fully-connected layers. Specifically, we compute activations of the VGG's `fc7` layer or the GoogLeNet's `inception 5b`

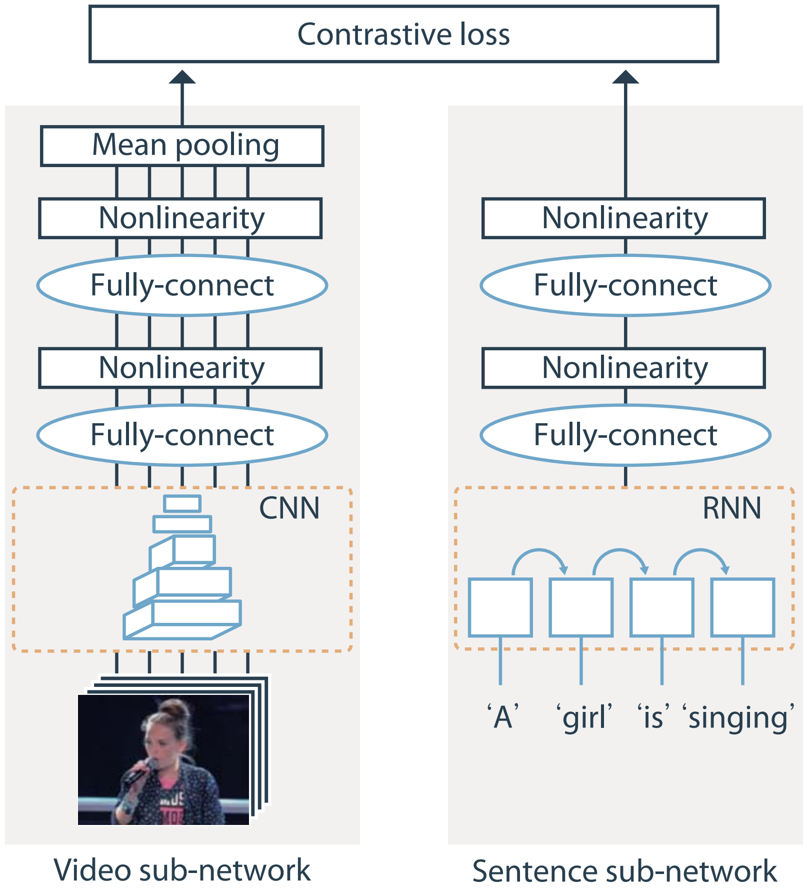


Figure 4.1. The network architecture. Video clips and sentences are encoded into vectors in the same size. Both sub-networks for videos and sentences are trained jointly by minimizing the contrastive loss.

layer and feed them to additional fully-connected layers.

Let $V = \{v_1, \ldots v_N\}$ be a set of frames v_i, where $v_n \in \mathrm{R}^{d_\mathrm{v}}$ is a visual feature extracted from n-th frame (d_v=4,096 for VGG, and d_v=1,024 for GoogLeNet). The video embedding $x \in \mathrm{R}^{d_\mathrm{e}}$ is computed by:

$$x = \operatorname*{mean}_{v \in V}[\tanh(W_{\mathrm{v}_2} \tanh(W_{\mathrm{v}_1} v + b_{\mathrm{v}_1}) + b_{\mathrm{v}_2})]. \tag{4.1}$$

Here, $W_{\mathrm{v}_1} \in \mathrm{R}^{d_\mathrm{h} \times d_\mathrm{v}}$, $b_{\mathrm{v}_1} \in \mathrm{R}^{d_\mathrm{h}}$, $W_{\mathrm{v}_2} \in \mathrm{R}^{d_\mathrm{e} \times d_\mathrm{h}}$, and $b_{\mathrm{v}_2} \in \mathrm{R}^{d_\mathrm{e}}$ are the learnable parameters of the fully-connected layers. mean[·] denotes a mean pooling, which take the average of input vectors.

2.2 Sentence Embedding

For the sentence sub-network, we use skip-thought vector model by Kiros *et al.* [39], which encodes a sentence into 4800-dimensional vectors with an RNN. Similarly to the video sub-network, we introduce two fully-connected layers with hyperbolic tangent nonlinearity (but without a mean pooling layer) as in Figure 4.1 to calculate a sentence representation. We encode sentences into vector representations using skip-thought that is an RNN pre-trained with a large-scale book corpus [39]. Let $T = \{w_1 \dots w_M\}$ be the input sentence, where $w_t \in \mathrm{R}^{d_\mathrm{w}}$ is a word vector of the t-th word in the sentence. Skip-thought takes a sequence of word vectors as in [39] and produces hidden state $h_t \in \mathrm{R}^{d_\mathrm{s}}$ at each time step t as:

$$
\begin{aligned}
j_t &= \sigma(W_\mathrm{r} w_t + U_\mathrm{r} w_{t-1}), && (4.2)\\
i_t &= \sigma(W_\mathrm{i} w_t + U_\mathrm{i} h_{t-1}), && (4.3)\\
a_t &= \tanh(W_\mathrm{a} w_t + U_\mathrm{a}(j_t \odot h_{t-1})), && (4.4)\\
h_t &= (1 - i_t) \odot h_{t-1} + i_t \odot a_t, && (4.5)
\end{aligned}
$$

where σ is the sigmoid activation function, and $\odot$ is the component-wise product. The parameters $W_\mathrm{r}, W_\mathrm{i}, W_\mathrm{a}, U_\mathrm{r}, U_\mathrm{i}$, and U_a are $d_\mathrm{s} \times d_\mathrm{w}$ matrices. Sentence Y is encoded into the hidden state after processing the last word. We use combine-skip in [39], which is a concatenation of outputs from two separate RNNs trained with different datasets. We denote the output of combine-skip from sentence T by $t_\mathrm{cs} \in \mathrm{R}^{d_\mathrm{c}}$, where d_c=4,800.

We then transform the skip-thought vectors s_Y into a sentence embedding y with two fully-connected layers as:

$$
y = \tanh(W_{\mathrm{s}_2} \tanh(W_{\mathrm{s}_1} t_\mathrm{cs} + b_{\mathrm{s}_1}) + b_{\mathrm{s}_2}), \qquad (4.6)
$$

where $W_{\mathrm{s}_1} \in \mathrm{R}^{d_\mathrm{h} \times d_\mathrm{c}}$, $b_{\mathrm{s}_1} \in \mathrm{R}^{d_\mathrm{h}}$, $W_{\mathrm{s}_2} \in \mathrm{R}^{d_\mathrm{e} \times d_\mathrm{h}}$, and $b_{\mathrm{s}_2} \in \mathrm{R}^{d_\mathrm{e}}$ are the learnable parameters of sentence embedding.

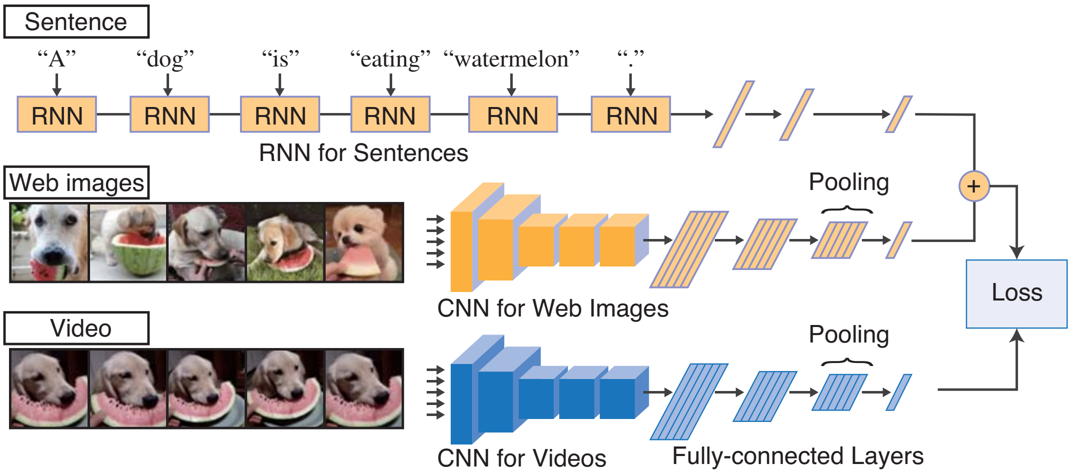


Figure 4.2. Illustration of our video and sentence embedding with web images. The orange component is the sentence embedding model that takes a sentence and corresponding web images as input. Video embedding model is denoted by the blue component.

2.3 Sentence Embedding with Web Images

In addition to the sentence embedding model, we propose to extend the sentence embedding with web images. To enhance the sentence embedding, we retrieve relevant web images that are expected to disambiguate semantics of the sentence. For example, the word "keyboard" can be interpreted as a musical instrument or an input device for computers. If the word comes with "play," the meaning of "keyboard" narrows down to a musical instrument. This means that a specific combination of words can reduce the possible visual concepts relevant to the sentence, which may not be fully encoded even with the state-of-the-art RNN-based approach like [39].

We propose to take this into account by using web image search results. Since most image search engines use surrounding text to retrieve images, we can expect that they are responsive to such word combinations. Consequently, we retrieve web images using the input sentence as a query and download the results. The web images are fused with the input sentence by applying a two-branch neural network as shown in Figure 4.2.

This sentence embedding model consists of two branches that merge the outputs of a CNN-based network for web images and an RNN-based network for a sentence described in Section 2.2. Before computing the sentence embedding, we download top-K results of web image search with the input sentence as a query. Let $Z = \{z_1 \dots z_K\}$ be a set of web images. We utilize the same architecture as the video embedding and compute an intermediate representation $e_\mathrm{z} \in \mathrm{R}^{d_\mathrm{e}}$ that integrates the web images as:

$$e_\mathrm{z} = \operatorname*{mean}_{z \in Z}[\tanh(W_{\mathrm{z}_2} \tanh(W_{\mathrm{z}_1} z + b_{\mathrm{z}_1}) + b_{\mathrm{z}_2})], \tag{4.7}$$

where $W_{\mathrm{z}_1} \in \mathrm{R}^{d_\mathrm{h} \times d_\mathrm{v}}$, $b_{\mathrm{z}_1} \in \mathrm{R}^{d_\mathrm{h}}$, $W_{\mathrm{z}_2} \in \mathrm{R}^{d_\mathrm{e} \times d_\mathrm{h}}$, and $b_{\mathrm{z}_2} \in \mathrm{R}^{d_\mathrm{e}}$ are the leanable parameters of the two fully-connected layers.

Once the outputs e_z are computed, the sentence embedding using web images y_z is computed as:

$$y_z = \frac{1}{2}(y + e_\mathrm{z}). \tag{4.8}$$

By this simple mixture of y and e_z, the sentence and web images directly influence the sentence embedding.

2.4 Joint Learning of Embedding Models

We jointly train both embedding models for videos and sentences using pairs of videos and associated sentences in a training set by minimizing the contrastive loss [7]. In our approach, the contrastive loss decreases when embeddings of videos and sentences with similar semantics get closer to each other in the embedding space, and those with dissimilar semantics get farther apart.

The training process requires a set of positive and negative video-sentence pairs. A positive pair contains a video and a sentence that are semantically relevant, and a negative pair contains irrelevant ones. During training, we get a positive pair by sampling a video and its description. Let $\{(x_n, y_n) \mid n = 1, \dots, N\}$ be the set of positive pairs. Given a positive pair (x_n, y_n), we sample irrelevant sentences and compute their embeddings $\mathcal{Y}' = \{y'_1 \dots y'_{N_\mathrm{c}}\}$, as well as, videos $\mathcal{X}' = \{x'_1 \dots x'_{N_\mathrm{c}}\}$ from the training set, which are used to build two sets of negative pairs $\{(x_n, y') \mid y' \in \mathcal{Y}'\}$ and $\{(x', y_n) \mid x' \in \mathcal{X}'\}$. Our embedding models for sentences and videos are jointly optimized by minimizing the contrastive loss

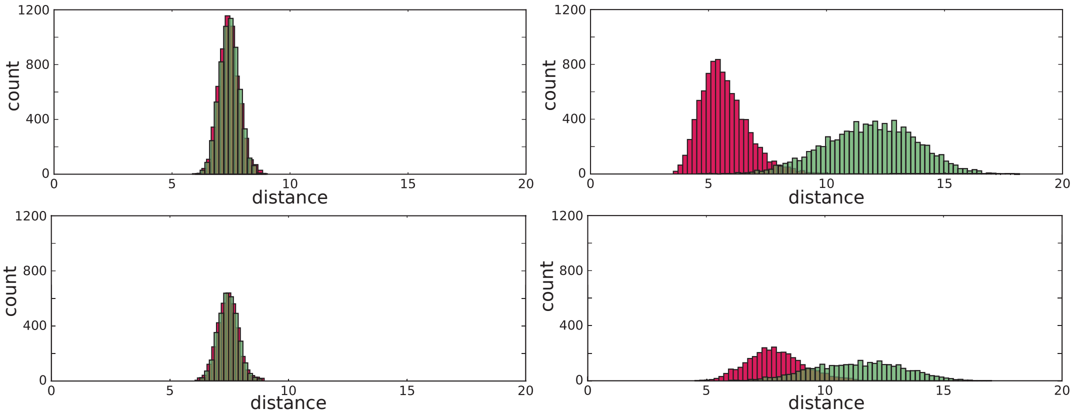


Figure 4.3. Histograms of pairwise distances before training (left) and after training (right). Top row: Histograms of the training set. Bottom row: Histograms of the test set. Red represents positive pairs and green does negative pairs.

defined as:

$$Loss(x_n, x_n) = \frac{1}{1+2N_c}\Bigg\{d(x_n, y_n) + \sum_{y' \in \mathcal{Y}'} \max(0, \alpha - d(x_n, y')) + \sum_{x' \in \mathcal{X}'} \max(0, \alpha - d(x', y_n))\Bigg\}, \quad (4.9)$$

where $d(\cdot, \cdot)$ denotes euclidean distance between embeddings. The hyperparameter α is a margin. Negative pairs with smaller distances than α are penalized. Margin α is set to the largest distance of positive pairs before training so that most negative pairs influence the model parameters at the beginning of training.

Figure 4.3 shows the histograms of distances of positive and negative pairs before and after training. The initial distance distributions of positive and negative pairs overlap. After training, the distributions are pulled apart. This indicates that the training process encourages videos and sentences in positive pairs to be mapped to closer points and those in negative ones to farther points.

Figure 4.4 shows a 2D plot of learned deep representations, in which the dimensionality of the semantic space is reduced using t-SNE [98] and a keyframe of each video clip is placed at the corresponding position, visualizing learned representations of video clips. This plot demonstrates that our embedding model for videos successfully locates semantically relevant videos at closer points. For

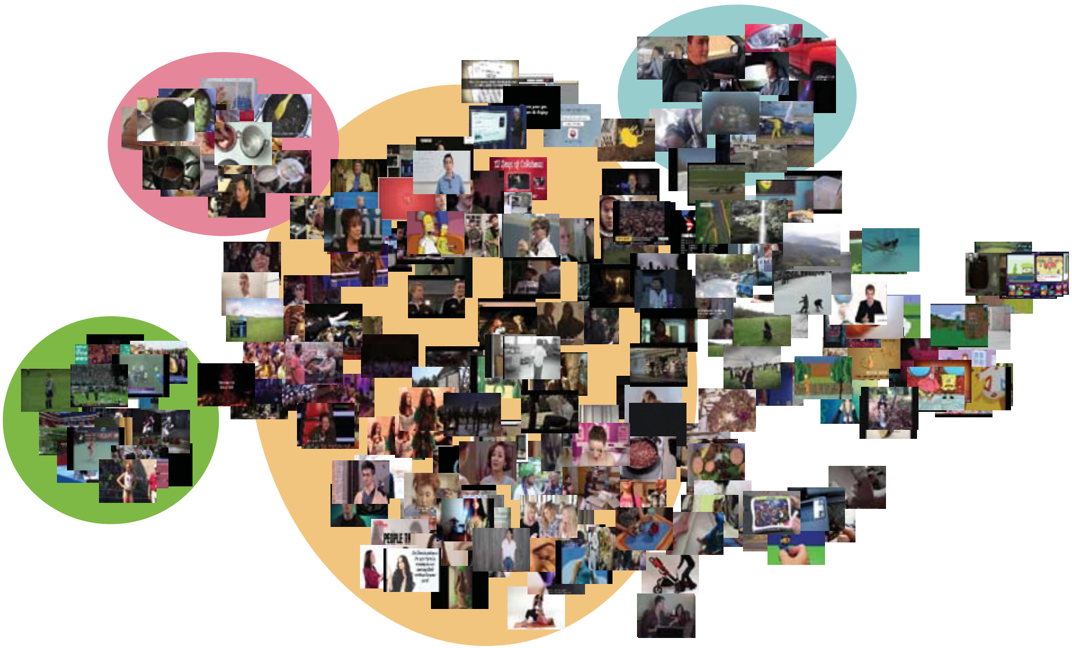


Figure 4.4. Two-dimensional deep representation embedding with keyframes of corresponding videos, where the representation dimensionality is reduced with t-SNE. The videos located on each colored ellipsis show similar content, *e.g.*, cars and driving people (blue), sports (green), talking people (orange), and cooking (pink).

example, the group of videos around the upper left area (pink) contains cooking videos, and another group on the lower left (green) shows various sports videos.

We also show the examples of positive and negative pairs with corresponding distances in Figure 4.5. Note that these distances are computed with the model that involves web images for sentence embedding. The positive pairs (a) and (b) are easy cases, in which sentences explicitly describe the video contents. The pair (c) is an example of hard cases. The sentence includes “a man” and “phone”, but the video shows two men, and a phone is occluded by a hand.

The pairs (d) and (e) are hard negative cases. The pair (d) shows partial matches of contents, such as the action “mixing” and the object “yolk.” Another negative pair (e) has a video and a sentence about cooking, although there is disagreement about details. As shown in these examples, the closer a video and

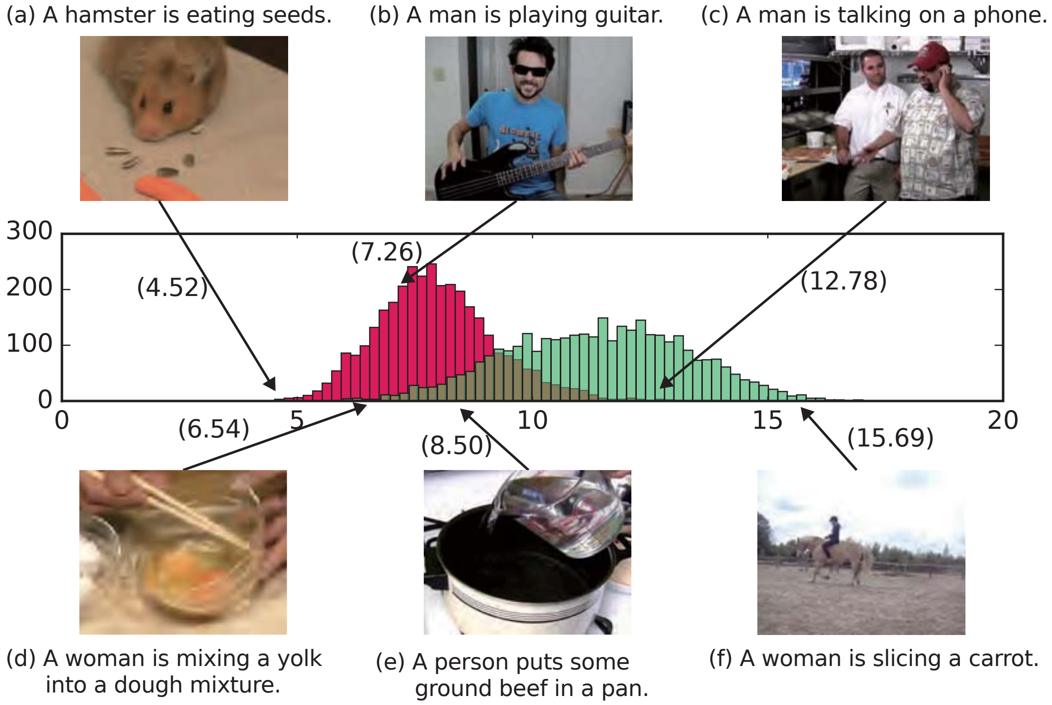


Figure 4.5. Examples of positive (a)–(c) and negative (d)–(f) pairs in the test set with corresponding distances. The values $(\cdot)$ are distances of the pairs. The plot shows the histograms of distances of positive (red) and negative (green) pairs.

a sentence are located in the embedding space, the more relevant they are.

3. Experiments

In the following experiments, we evaluate our cross-domain representations on several tasks: video summarization, content-based video retrieval, and video captioning.

3.1 Video Summarization

In this experiment, we investigate how well our cross-modal representations perform on the task of video summarization. Summarizing real-world videos requires encoding various visual concepts. There are several deep representations

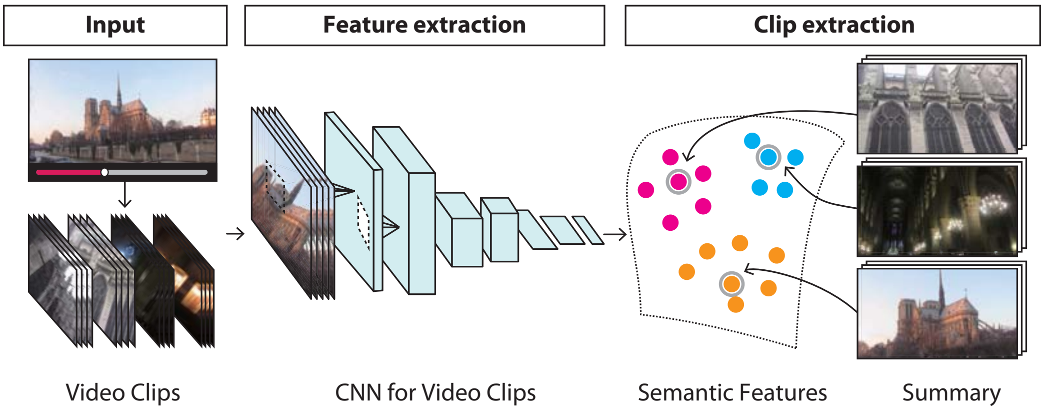


Figure 4.6. Video summarization using deep semantic representations. We extract uniform length video clips from an input video. The clips are fed to a CNN for visual feature extraction and then mapped to points in a semantic space. We finally generate a video summary by sampling video clips that correspond to cluster centers in the semantic space.

using convolutional neural networks (CNNs) [111, 10], which are employed in many recognition tasks including object recognition [10], and video summarization [25]. These deep representations are trained for specific classification tasks, which predict class labels of a particular domain, such as objects and actions. Being different from these deep representations, our cross-modal representations are trained to encode combinations of diverse visual concepts to handle a wide range of video contents. We expect our cross-modal representation well captures sentence-level semantics. In this experiment, we perform video summarization that tries to extract semantically representative video clips using our cross-modal representation.

We evaluated and compared video summaries generated using our cross-modal representations with some baselines. We used the SumMe dataset [24] consisting of 25 videos for evaluation. As the videos in this dataset are either unedited or slightly edited, unimportant or redundant parts are left in the videos. The dataset includes videos with various contents. It also provides manually created video summaries for each video, with which we compare our summaries. We compute the pairwise f-measure that evaluates agreement to reference video summaries

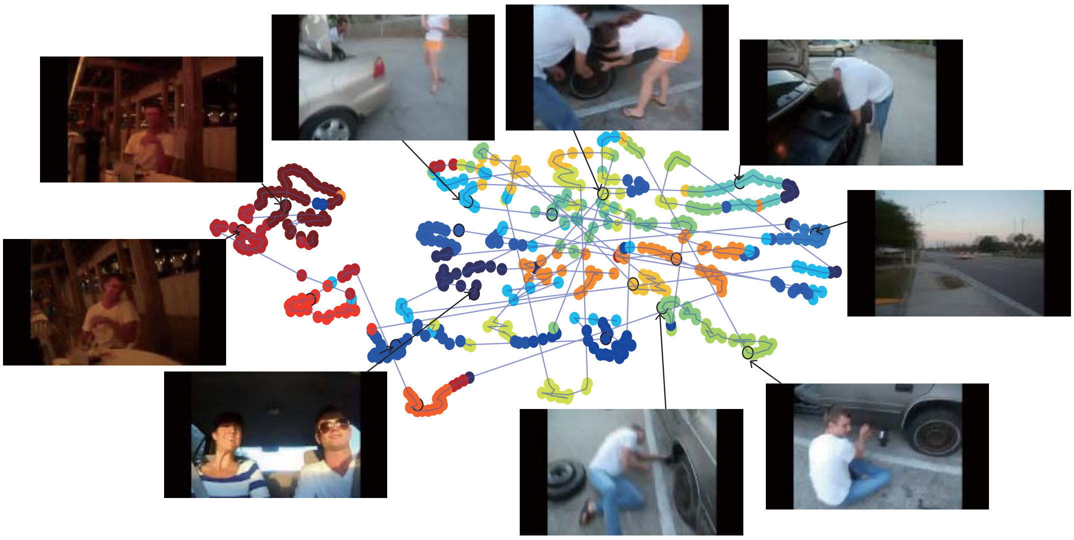


Figure 4.7. A two-dimensional plot of our deep representations calculated from a video, where we reduce the deep representations' dimensionality with t-SNE [98]. Some deep representations are represented by the corresponding video clips' keyframes, and the edges connecting deep representations represent temporal adjacency of video clips. The colors of deep representations indicate clusters obtained by k-means algorithm, *i.e.*, points with the same color belong to the same cluster.

using the code provided in [24]. The pairwise f-measure is defined by:

$$F_1 = 2\frac{PRE * REC}{PRE + REC}, \tag{4.10}$$

where PRE and REC are precision and recall of selected frames to human selection. In this experiment, we do not use any metadata such as textual scripts as in Chapter 3. Therefore, we use only video representations that are learned from video-sentence pairs.

Figure 4.6 shows an overview of video summarization using learned representations. We first extract uniform length video clips from the input video in a temporal sliding window manner and compute their cross-modal representations with our video embedding model. Inspired by [9], we represent the input video as a set of representations, each of which corresponds to a video clip, as shown in Figure 4.7. This representation can capture the semantic distribution of the input video. In Figure 4.7, some clusters can be observed, each of which is expected to contain semantically similar video clips. Based on this assumption, our approach picks out a subset of video clips which correspond to cluster centers.

Generating Video Summary

In this experiment, we generate a video summary given an input video by solving the k-medoids problem [25]. In the k-medoids problem, we find a subset K video clips, which are cluster centers that minimize the sum of the Euclidean distance of all video clips to their nearest cluster centers and K is a given parameter to determine the length of the video summary. Letting $X = \{x_1, \dots, x_L\}$ be a set of cross-modal representations extracted from all video clips in the input video, k-medoids finds a subset $S \subset X$, that minimizes the objective function defined as:

$$F(S) = \sum_{x \in X} \min_{s \in S} \|x - s\|_2^2. \tag{4.11}$$

The optimal subset

$$S^* = \operatorname*{argmin}_S F(S) \tag{4.12}$$

includes the most representative clips in clusters. As shown in Figure 4.4, our video sub-network maps clips with similar semantics to closer points in the semantic space; therefore we can expect that the clips in a cluster have semantically similar content and subset S^* consequently includes most representative and diverse video clips. The clips in S^* are concatenated in the temporal order to generate a video summary.

Implementation Detail

Deep representation computation. We uniformly extract 5-second video clips in a temporal sliding window manner, where the window is shifted by 1 second. Each video clip is re-sampled at 1 frame per second. In this experiment, we use VGG `fc7` layer to extract visual features from frames. We set the unit size of the two fully connected layers to $d_\text{v} = 1000$ and $d_\text{e} = 300$, respectively. This means that our cross-modal representation is a 300-dimensional vector. For the sentence embedding model, the fully-connected layers on the top of the RNN have the same sizes as the video sub-networks. During the training, we fix the network parameters of VGG and skip-thought, but those of the top two fully-connected layers for both video and sentence embedding models are updated. We set N_c to 20 during training. Our DNN was trained over the MSR-VTT dataset [106], which consists of 1M video clips annotated with 20 descriptions for each. We used Adam [38] to optimize the network parameters with the learning rate of 2^{-4} and trained for 4 epochs. We implemented our model using Chainer [90].

Video summarization generation. Given an input video, we sampled 5-second video clips in the same way as the training of our DNN and extracted a deep representation from each clip. We then minimize the objective function in Eq. (4.11) with cost-effective lazy forward selection [24, 48]. We set the summary length to be roughly 15% of the input video's length following [24].

Baselines

We compared our video summaries with the following several baselines as well as recent video summarization approaches:

1. **Manually-created** video summaries are a powerful baseline that may be viewed as the upper bound for automatic approaches. The SumMe dataset provides at least 15 manually-created video summaries whose length is 15% of the original video. We computed the average f-measure of each manually-created video summary with letting each of the rest manually-created video summaries as ground truth (*i.e.*, if there are 20 manually-created video summaries, we compute 19 f-measures for each summary in a pairwise manner and calculate their average). We denote the summary with the highest f-measure among all manually-created video summaries by the best-human video summary.

2. **Uniform sampling** (Uni.) is widely used as a baseline for video summarization evaluation. We segment an input video into 5-second video clips and sample them with a uniform interval so that the durations of sampled video clips total 15% of the original length.

3. **VGG**-based video summary. We also compare to video summaries generated in the same approach as ours except that VGG's `fc7` activations were used instead of our deep representations.

4. **Attention-based** video summary (Attn.) is a recently proposed video summarization approach using visual attention [11].

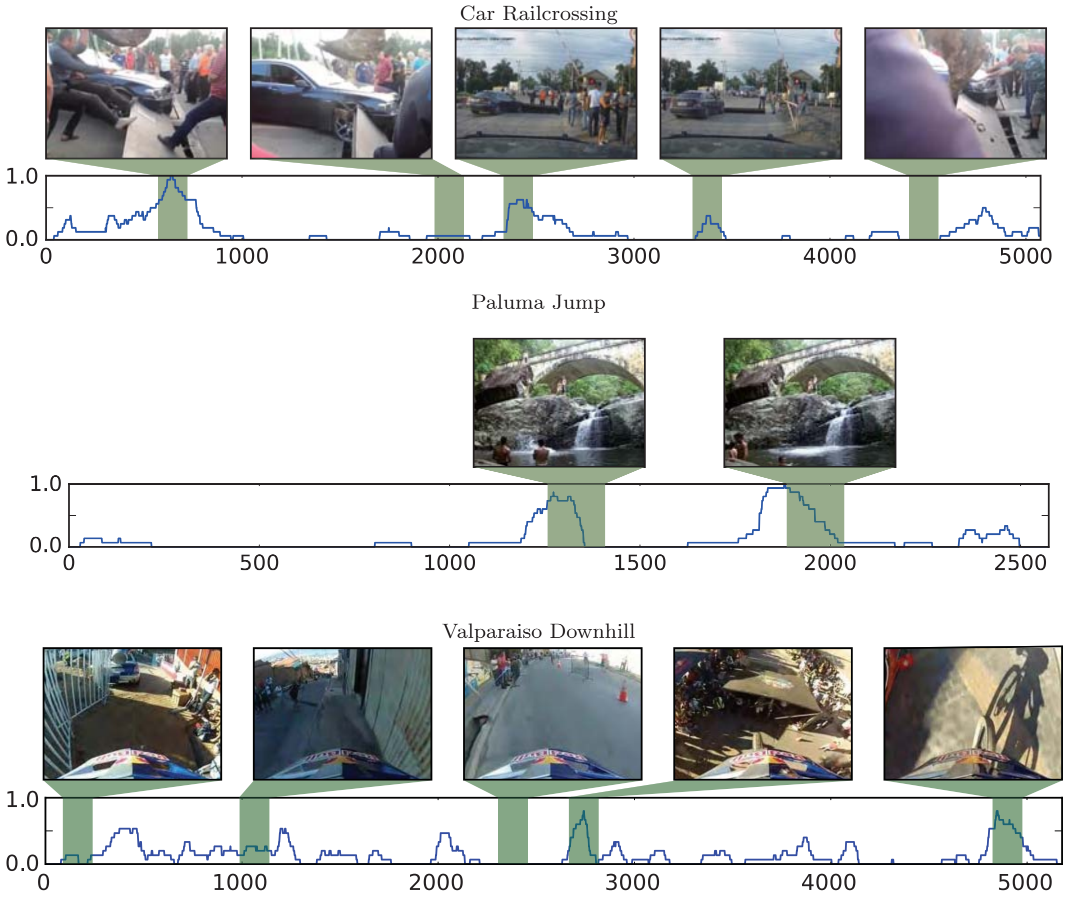
5. **Interestingness-based** video summary (Intr.) refers to a supervised approach [24], where the weights of multiple objectives are optimized using the SumMe dataset.

Results

Several examples of video summaries generated with our approach are shown in Figure 4.8, along with the ratio of annotators who agreed to include each video clip in their manually-created video summary. The peaks of the blue lines indicate that the corresponding video clips were frequently selected to create a video summary. These blue lines demonstrate that human annotators were consistent to some extent. Also, we observe that the video clips selected by our approach (green areas) are correlated to the blue lines. This suggests that our approach is consistent with the selection of human annotators.

The results of the quantitative evaluation are summarized in Table 4.1. In this table, we report the minimum, average, and maximum f-measure scores of manually-created video summaries. Compared with VGG-based summaries, ours significantly improved the scores. Our video summaries achieved 58.8% of the average score of manually-created video summaries, while VGG-based got 40.8%. This result demonstrates the advantage of our deep representations for creating video summaries.

One of the recent video summarization approaches, *i.e.*, interestingness-based one [24], got the highest score in this experiment. Note that the interestingness-based approach [24] uses a supervised technique, in which the mixture weights of various criteria in their objective function are optimized over the SumMe dataset. Our video summaries were generated using a relatively simple algorithm to extract a subset of clips; nevertheless, ours outperformed the interestingness-based for some videos and even got a better mean f-measure score than attention-based.


Car Railcrossing

Paluma Jump

Valparaiso Downhill

Figure 4.8. Clips selected by our approach. Keyframes of selected clips are displayed. The green areas in the graphs indicate selected clips. The blue lines represents the ratio of annotators who selected the clip for their manually-created summary.

Our approach got low scores, especially for short videos, such as "Jumps" and "Fire Domino." Since we extract uniform length clips (5 seconds), in the case of short videos, our approach only extracts a few clips. This may result in a lower f-measure score. This limitation can be solved by extracting shorter video clips or using more sophisticated video segmentation like [76, 24].

We also observed that our approach got lower scores than others on the "St Maarten Landing" and "Notre Dame," which are challenging because of long unimportant parts and diversity of content, respectively. For "St Maarten Land-

Table 4.1. F-measures of manually-created video summaries and computational approaches (our approach and baselines, higher is better). Since there are multiple manually-created video summaries for each original video and thus multiple f-measures, we show their minimum, mean, and maximum. The best score among the computational approaches are highlighted.

	Manually created			Computational approaches				
Video	Min.	Avg.	Max.	Uni.	VGG	Attn.	Intr.	Ours
Air Force One	0.185	0.332	0.457	0.060	0.239	0.215	**0.318**	0.316
Base Jumping	0.113	0.257	0.396	**0.247**	0.062	0.194	0.121	0.077
Bearpark Climbing	0.129	0.208	0.267	0.225	0.134	**0.227**	0.118	0.178
Bike Polo	0.190	0.322	0.436	0.190	0.069	0.076	**0.356**	0.235
Bus in Rock Tunnel	0.126	0.198	0.270	0.114	0.120	0.112	0.135	**0.151**
Car Railcrossing	0.245	0.357	0.454	0.185	0.139	0.064	**0.362**	0.328
Cockpit Landing	0.110	0.279	0.366	0.103	**0.190**	0.116	0.172	0.165
Cooking	0.273	0.379	0.496	0.076	0.285	0.118	0.321	**0.329**
Eiffel Tower	0.233	0.312	0.426	0.142	0.008	0.136	**0.295**	0.174
Excavators River Crossing	0.108	0.303	0.397	0.107	0.030	0.041	**0.189**	0.134
Fire Domino	0.170	0.394	0.517	0.103	0.124	**0.252**	0.130	0.022
Jumps	0.214	0.483	0.569	0.054	0.000	0.243	**0.427**	0.015
Kids Playing in Leaves	0.141	0.289	0.416	0.051	0.243	0.084	0.089	**0.278**
Notre Dame	0.179	0.231	0.287	0.156	0.136	0.138	**0.235**	0.093
Paintball	0.145	0.399	0.503	0.071	0.270	0.281	**0.320**	0.274
Playing on Water Slide	0.139	0.195	0.284	0.075	0.092	0.124	**0.200**	0.183
Saving Dolphines	0.095	0.188	0.242	0.146	0.103	**0.154**	0.145	0.121
Scuba	0.109	0.217	0.302	0.070	0.160	**0.200**	0.184	0.154
St Maarten Landing	0.365	0.496	0.606	0.152	0.153	**0.419**	0.313	0.015
Statue of Liberty	0.096	0.184	0.280	0.184	0.098	0.083	**0.192**	0.143
Uncut Evening Flight	0.206	0.350	0.421	0.074	0.168	**0.299**	0.271	0.168
Valparaiso Downhill	0.148	0.272	0.400	0.083	0.110	0.231	0.242	**0.258**
Car over Camera	0.214	0.346	0.418	0.245	0.048	0.201	**0.372**	0.132
Paluma Jump	0.346	0.509	0.642	0.058	0.056	0.028	0.181	**0.428**
playing ball	0.190	0.271	0.364	0.123	0.127	0.140	0.174	**0.194**
Mean f-measure	0.179	0.311	0.409	0.124	0.127	0.167	**0.234**	0.183
Relative to human avg.	0.576	1.000	1.315	0.398	0.408	0.537	**0.752**	0.588
Relative to human max.	0.438	0.760	1.000	0.303	0.310	0.408	**0.572**	0.447

ing," as our approach is unsupervised, it failed to exclude unimportant clips. For "Notre Dame," generating a summary is difficult because there are too many pos-

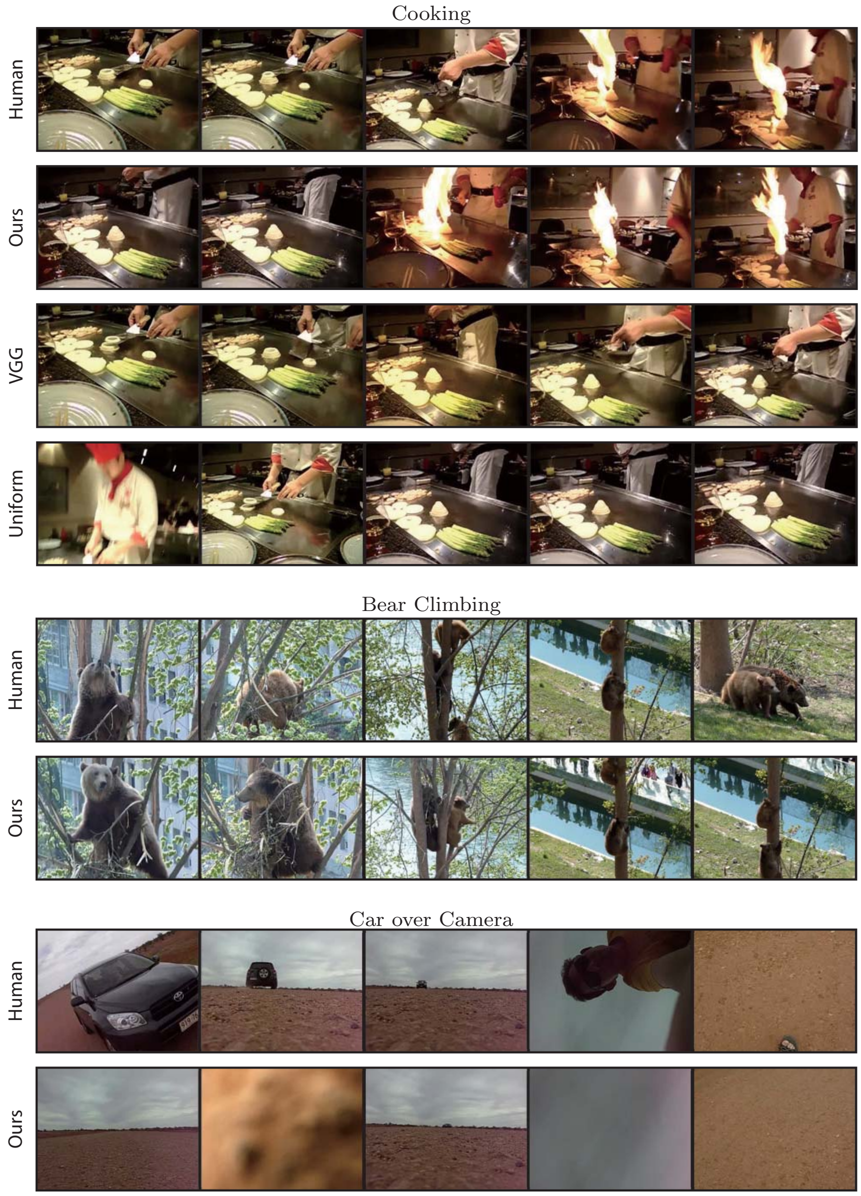


Figure 4.9. Uniformly sampled frames of summaries by different approaches. “Human” means the best-human video summary.

sible clips to be included in a summary. While our summary shares small parts with manually created summaries, it is a challenging example even for human annotators, which is shown in the low scores of manually-created video summaries.

Figure 4.9 shows examples of video summaries created with our approach and baselines. The video "Cooking" shows a person cooking some vegetables while doing a performance. Ours and the best-human video summary include the same scene of the performance with fire, while others do not. On the other hand, ours extracts unimportant clips from the video "Car over Camera." The original video is highly redundant with static scenes just showing the ground or the sky, and such scenes make up large clusters in the semantic space even if they are unimportant. As our approach extracts representatives from each cluster, a video with lengthy unimportant parts results in a poor video summary. We believe that this problem can be avoided by using visual cues such as interestingness [23] and objectiveness [3].

3.2 Content-based Video and Sentence Retrieval

Implementation Detail

With 19-layer VGG, the hidden layer size d_h of embedding models was set to 1,000 and the dimension of the embedding space d_e was set to 300. We set $N_\mathrm{c} = 50$ in this experiment. For model using GoogLeNet, we used $d_\mathrm{h} = 600$ and $d_\mathrm{e} = 300$.

We implemented our model using Chainer [90]. We used Adam [38] for optimization with a learning rate of 10^{-4}. The parameters of the CNNs and skip-thought were fixed. We applied dropout with a ratio of 0.5 to the top two layers of video and sentence embedding models. Our models were trained for 15 epochs, and their parameters were saved at every 100 updates. We took the model parameters whose performance was the best on the validation set.

Experimental Setup

Dataset: We used the YouTube dataset [5] consisting of 80K English descriptions for 1,970 videos. We first divided the dataset into 1,200, 100, and 670 videos for training, validation, and test, respectively, as in [111, 108, 22]. Then, we extracted five-second clips from each original video in a sliding-window manner. As a result, we obtained 8,001, 628, and 4,499 clips for the training, validation, and test sets, respectively. For each clip, we picked five ground truth descriptions out of those associated with its original video.

We collected top-5 image search results for each sentence using the Bing image search engine. We used a sentence modified by lowercasing and punctuation removal as a query. In order to eliminate cartoons and clip art, the image type was limited to photos using Bing API.

Video Retrieval: Given a video and a query sentence, we extracted five-second video clips from the video and computed Euclidean distances from the query to the clips. We used their median as the distance of the original video and the query. We ranked the videos based on the distance to each query and recorded the rank of the ground truth video. Since the test set has 670 videos, the probability of bringing the ground truth video at top-1 by random ranking is about 0.15%.

Sentence Retrieval: For the sentence retrieval task, we ranked sentences for each query video. We computed the distances between a sentence and a query video in the same way as the video retrieval task. Note that each video has five ground truth sentences; thus, we recorded the highest rank among them. The test set has 3,500 sentences.

Evaluation Metrics: We report recall rates at top-1, -5, and -10, the average and median rank, which are standard metrics employed in the retrieval evaluation. We found that some videos in the dataset had sentences whose semantics were almost the same (*e.g.*, “A group of women is dancing” and “Women are dancing”). For the video that is annotated with one of such sentences, the other sentence is treated as incorrect with the recall rates, which does not agree with human judges. Therefore, we employed additional evaluation metrics widely used in the description generation task, *i.e.*, CIDEr, BLUE@4, and METEOR [6]. They compute agreement scores in different ways using a retrieved sentence and a set of ground truth ones associated with a query video. Thus, these metrics give high scores for semantically relevant sentences even if they are not annotated to a query video. We computed the scores of the top-ranked sentence for each video using the evaluation script provided in the Microsoft COCO Evaluation Server [6]. In our experiments, all ground truth descriptions for each original video are used to compute these scores.

Table 4.2. Video and sentence retrieval results. R@K is recall at top K results (higher values are better). aR and mR are the average and median of rank (lower values are better). Bold values denote best scores of each metric.

	Video retrieval					Sentence retrieval				
Models	**R@1**	**R@5**	**R@10**	**aR**	**mR**	**R@1**	**R@5**	**R@10**	**aR**	**mR**
Random Ranking	0.15	0.79	1.48	335.92	333	0.22	0.69	1.32	561.32	439
VGG+VS	6.12	21.88	33.22	58.98	24	7.01	18.66	27.16	131.33	35
VGG+VI	4.03	13.70	21.40	94.62	48	5.67	17.91	28.21	116.86	38
VGG+ALL$_1$	6.48	20.15	30.51	59.53	26	**10.60**	25.22	36.42	85.90	21
VGG+ALL$_2$	5.97	21.31	32.54	56.01	24	8.66	22.84	33.13	100.14	29
GoogLeNet+VS	7.49	22.84	33.10	54.14	22	8.51	21.34	30.45	114.66	33
GoogLeNet+VI	4.24	16.42	24.96	84.48	41	6.87	17.31	30.00	96.78	30
GoogLeNet+ALL$_1$	5.52	18.93	28.90	60.38	28	9.85	**27.01**	**38.36**	**75.23**	**19**
GoogLeNet+ALL$_2$	**7.67**	**23.40**	**34.99**	**49.08**	**21**	9.85	24.18	33.73	85.16	22
ST [39]	2.63	11.55	19.34	106.00	51	2.99	10.90	17.46	241.00	77
DVCT [108]	-	-	-	224.10	-	-	-	-	236.27	-

Table 4.3. Evaluated scores of retrieved sentences. All values are reported in percentage (%). Higher scores are better.

Models	**CIDEr**	**BLEU**	**METEOR**
VGG+VS	30.44	27.16	25.74
VGG+VI	29.00	22.42	22.99
VGG+ALL$_1$	42.52	**30.81**	**27.77**
VGG+ALL$_2$	32.56	27.39	26.58
GoogLeNet+VS	33.82	26.97	25.99
GoogLeNet+VI	35.08	24.56	24.16
GoogLeNet+ALL$_1$	**43.52**	29.99	27.48
GoogLeNet+ALL$_2$	38.08	29.28	26.50

Effects of Each Component of Our Approach

In order to investigate the influence of each component of our approach, we tested some variations of our full model. The scores of the models on the video and sentence retrieval tasks are shown in Table 4.2. Our full model that computes sentence embedding using web images is denoted by ALL$_2$. ALL$_1$ is a variation of ALL$_2$ that computes embeddings with one fully-connected layer with the unit size of d_e. Comparison between ALL$_1$ and ALL$_2$ indicates that the number of fully-connected layers in embedding is not essential.

Our model which does not use web images to compute sentence embeddings is denoted by VS. The comparison between our full model ALL and VS reveals the contributions of web images. VGG+ALL$_2$ had better average rank (aR) than VGG+VS on both video and sentence retrieval, and comparison between GoogLeNet+ALL$_2$ and GoogLeNet+VS also shows a clear advantage of incorporating web images.

We also tested a model without sentences, which is denoted by VI. In VI, the sentence embeddings are computed only from web images, *i.e.*, e_{z}. We investigated the effect of using both sentences and web images by comparing VI to our full model ALL$_2$. The results show that sentences are necessary. The comparison between VI and VS also indicates that sentences provide main cues for the retrieval task.

The scores of retrieved sentences computed by CIDEr, BLEU@4, and METEOR are shown in Table 4.3. In all metrics, our full model using both sentences and web images (ALL$_1$ and ALL$_2$) outperformed to other models (VS and VI). In summary, contributions by sentences and web images were non-trivial, and the best performance was achieved by using both of them.

Some examples of retrieved videos by GoogLeNet+VS, GoogLeNet+VI, and GoogLeNet+ALL$_2$ are shown in Figure 4.10. These results suggest that web images reduced the ambiguity of queries' semantics by providing hints on their visual concepts. For example, with the sentence (1) "A man is playing a keyboard," retrieval results of GoogleNet+VS includes two videos of a keyboard on a laptop as well as one on a musical instrument. On the other hand, all top-3 results by GoogleNet+ALL$_2$ are about musical instruments. We observed that web images retrieved by the query (1) included several images of musical instruments, which looked to be helpful to clarify the semantics of the query. We see web images often affected the retrieval results positively as in the example (3). The model without web images got videos about cooking, but there is disagreement in the details of the query and video content. With web images, our model obtained more relevant videos, which show a person cutting a large piece of meat. However, irrelevant image search results can harm the video retrieval performance. For a query"A monkey is fighting with a man"' in (4) resulted in irrelevant web

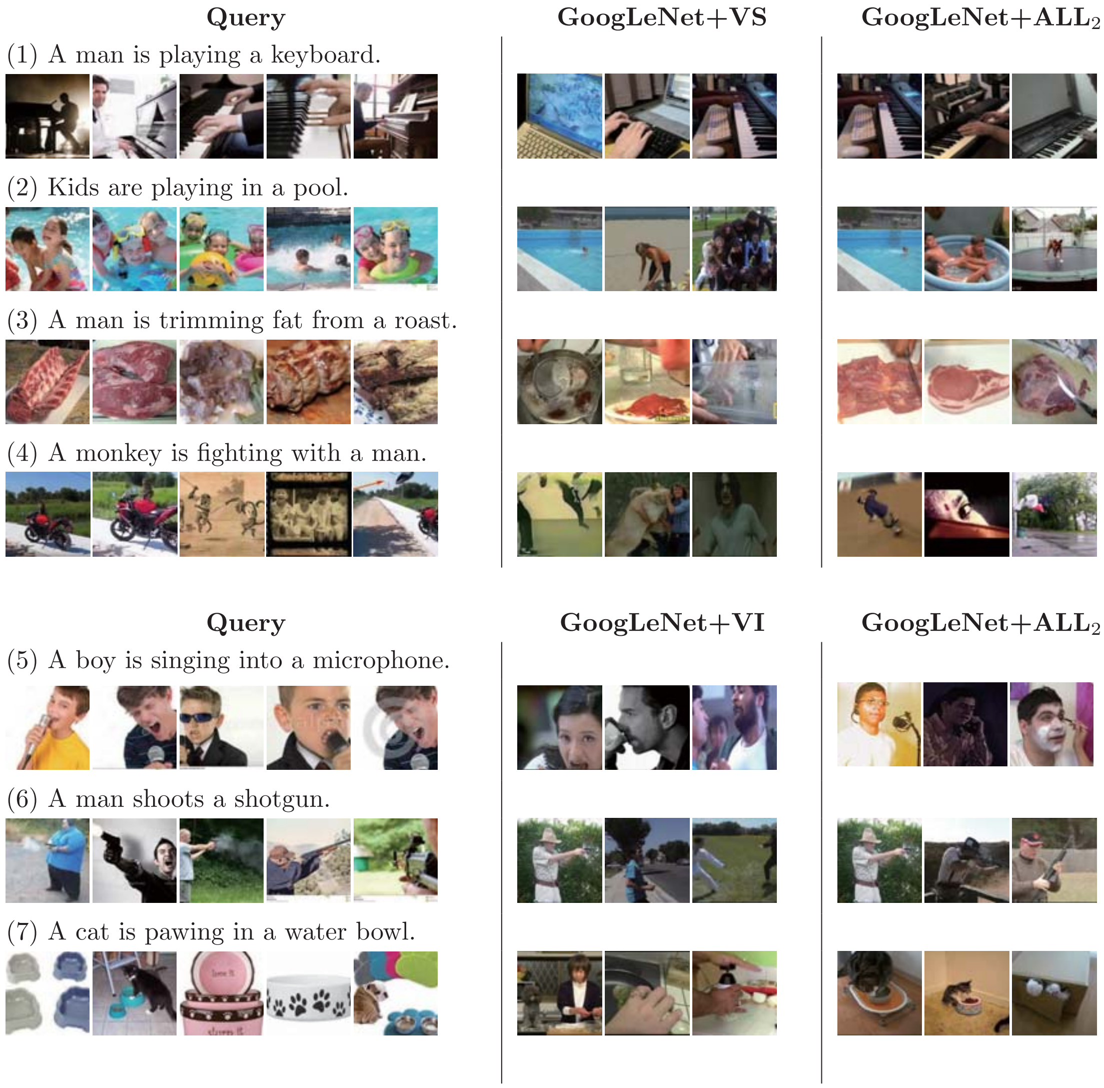

Query
GoogLeNet+VS
GoogLeNet+ALL$_2$
(1) A man is playing a keyboard.
(2) Kids are playing in a pool.
(3) A man is trimming fat from a roast.
(4) A monkey is fighting with a man.
Query
GoogLeNet+VI
GoogLeNet+ALL$_2$
(5) A boy is singing into a microphone.
(6) A man shoots a shotgun.
(7) A cat is pawing in a water bowl.

Figure 4.10. Examples of video retrieval results. Left: Query sentences and web images. Center: Top-3 retrieved videos by GoogLeNet+VS and VI. Right: Top-3 retrieved videos by GoogLeNet+ALL$_2$.

images, and our full model failed to get correct videos.

Compared to GoogLeNet+VI, our full model obtained more videos with relevant content for other queries. These results suggest that both sentence and web images are important for the performance of content-based video retrieval. The example in (7) also got irrelevant images as in (4), but this result indicates that

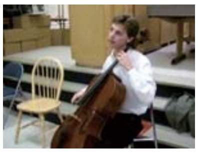
Query Video	GoogLeNet+All$_2$	GoogLeNet+VS
	1. A man is cutting a paper. 2. A man is cutting a paper by hands. 3. Someone is cutting the carrot into small pieces.	1. Someone is cutting the carrot into small pieces. 2. A person cuts a sock with scissors. 3. An oriental lady is cutting a carrot into thin pieces.
	1. A woman is talking while applying eyeshadow. 2. A woman applies Joker makeup to a man's face. 3. A woman is applying cosmetics to a man.	1. A woman is singing. 2. A woman is singing. 3. A woman wearing a headset is singing into a large microphone.
	1. A pair of zebras are playing with each other. 2. The zebras are playing. 3. A pair of zebras is nuzzling.	1. Leopards are congregating. 2. A group of deers are crossing road. 3. A pair of zebras is nuzzling.
	1. A man is playing keyboards. 2. A boy is playing a grand piano. 3. A boy is playing guitar.	1. A little boy is playing piano. 2. A little boy is playing a grand piano. 3. A boy is playing a piano.

Figure 4.11. Examples of top-3 retrieved sentences. Left: Query videos. Center: Top-3 retrieved sentences by GoogLeNet+ALL$_2$. Right: Top-3 retrieved sentences by GoogLeNet+VS.

our model may recover from irrelevant image search results by combining a query sentence.

Some examples of sentence retrieval results are shown in Figure 4.11. While our full model may retrieve sentences that disagree with query videos in details, most of the retrieved sentences are relevant to query videos.

Comparison to Prior Work

The approach for image and sentence retrieval by Kiros *et al.* [39] applies linear transformations to CNN-based image and RNN-based sentence representations to embed them into a common space. Note that their model was designed for the image and sentence retrieval tasks; thus, we extracted the middle frame as a keyframe and trained the model with pairs of a keyframe and a sentence. Xu *et al.* [108] introduced neural network-based embedding models for videos and sen-

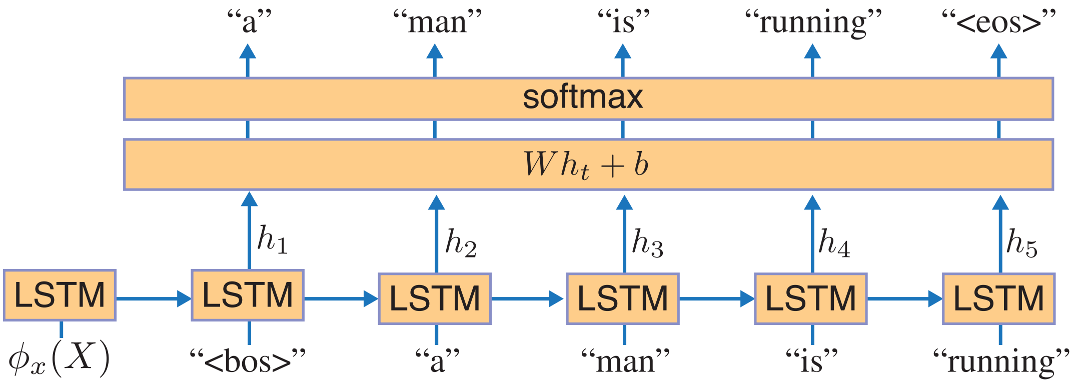


Figure 4.12. Illustration of the decoder model. "<bos>" is a tag denoting the beginning of a sentence, and "<eos>" is the end of a sentence.

tences. Their approach embeds videos and SVO triplets extracted from sentences into an embedding space. Kiros *et al.*'s and Xu *et al.*'s approaches are denoted by ST and DVCT, respectively.

Scores in Table 4.2 indicates that our model clearly outperformed prior work in both video and sentence retrieval tasks. There is a significant difference in performance of DVCT and others. ST and ours encode all words in a sentence, while DVCT only encodes its SVO triplets. This suggests that using all words in a sentence together with an RNN is necessary to get good embeddings.

3.3 Video Description Generation

Automatic description generation for images [101, 13] and videos [74, 100, 111, 99] is another task to associate images or videos with sentences. As an application of our models, we performed the description generation task using our video embeddings. To analyze the information encoded by our video embedding, we trained a decoder that produces descriptions from our video embeddings. A basic approach for description generation is to use long-short term memory (LSTM) that produces a sequence of probabilities over a vocabulary conditioned on visual representations [101, 100]. We trained an LSTM as a decoder of video embeddings (Figure 4.12). The decoder predicts the next word based on word vector w_t at

Table 4.4. Scores of generated sentences. TVNL+Extra Data is the TVNL model pre-trained on the Flickr30k [112] and the COCO2014 [6] datasets.

Models	**CIDEr**	**BLEU**	**METEOR**
TVNL [100]	-	31.19	26.87
TVNL+Extra Data	-	33.29	29.07
DVETS [111]	51.67	41.92	29.60
Ours	41.62	33.69	28.47

each time step t as:

$$[a_t \ i_t \ f_t \ o_t]^T = W_\mathrm{u} w_t + b_\mathrm{u} + W_\mathrm{l} h_{t-1}, \tag{4.13}$$

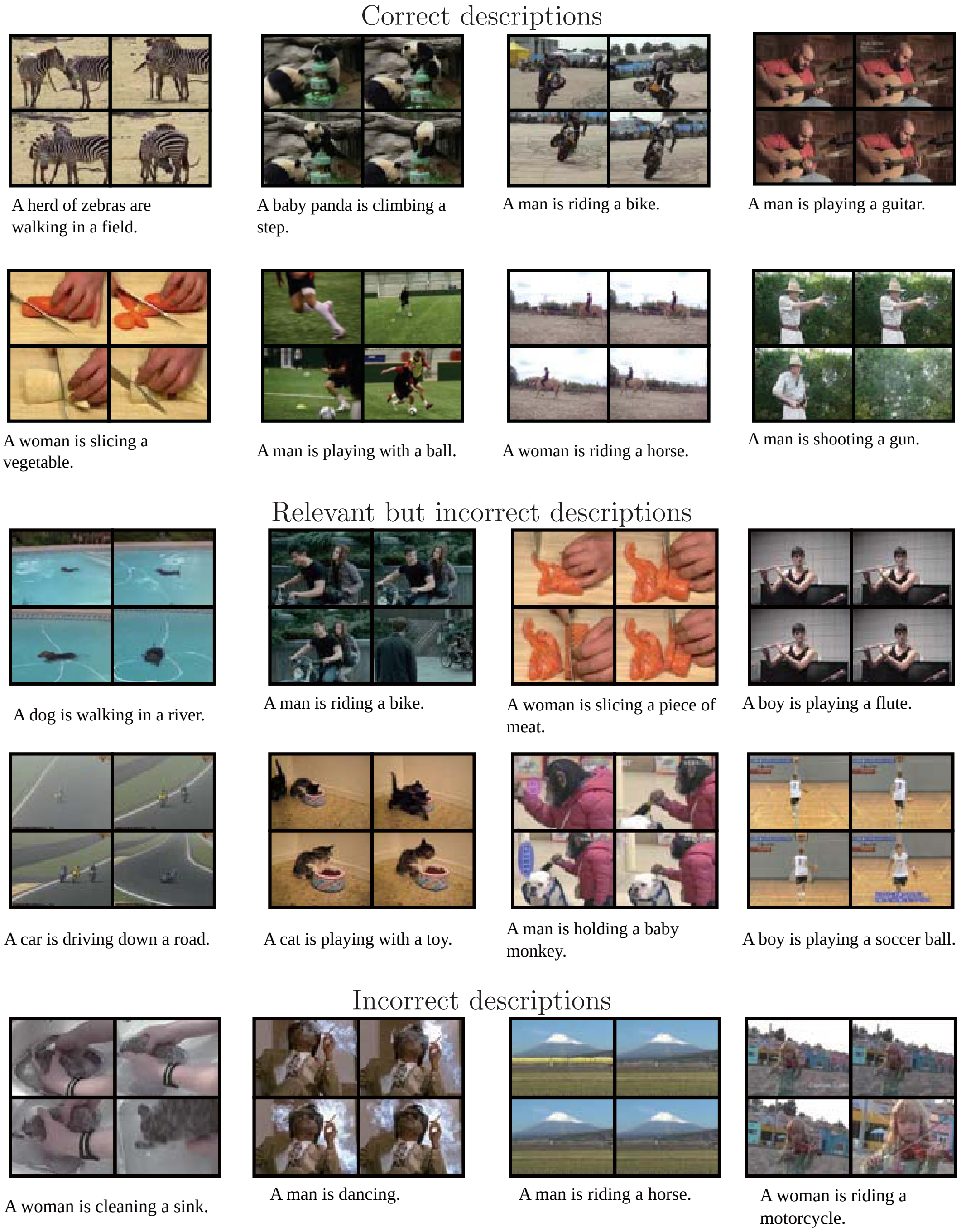
$$c_t = \tanh(a_t)\sigma(i_t) + c_{t-1}\sigma(f_t), \tag{4.14}$$

$$h_t = \tanh(c_t)\sigma(o_t), \tag{4.15}$$

$$p_t = \mathrm{softmax}(W_\mathrm{p} h_t + b_\mathrm{p}), \tag{4.16}$$

where $W_\mathrm{u}, W_\mathrm{l} \in \mathrm{R}^{4\mathrm{d_w} \times \mathrm{d_w}}$ and $b_u \in \mathrm{R}^{4d_\mathrm{w}}$ are parameters of the LSTM, and $[a_t \ i_t \ f_t \ o_t]^T$ is a column vector that is a concatenation of $a_t, i_t, f_t, o_t \in \mathrm{R}^{d_\mathrm{w}}$. The matrix W_p and the vector b_p encode the hidden state into a vector with the vocabulary size. The output p_t is the probabilities over the vocabulary. We built a vocabulary consisting of all words in the YouTube dataset and special tags, *i.e.*, begin-of-sentence ("<bos >") and end-of-sentence ("<eos >"). The generative process is terminated when "<eos >" is produced. We trained the decoder using the YouTube dataset. We computed the video embedding $\phi_\mathrm{v}(X)$ using GoogLeNet+ALL$_2$ as an input to the LSTM at $t = 0$. We trained the decoder by minimizing the cross-entropy loss. During training, we fixed the parameters of our embedding models.

Figure 4.13 shows generated sentences. Although video embeddings were trained for retrieval tasks and not fine-tuned for the decoder, we observed that most generated sentences were semantically relevant to their original videos. The results show that our model can produce correct descriptions for videos with diverse content, such as animals, sports, cooking *etc.* These results suggest that our model can encode various concepts into our representation. Although many descriptions involve disagreement in details, the descriptions are relevant to video



Correct descriptions

A herd of zebras are walking in a field.

A baby panda is climbing a step.

A man is riding a bike.

A man is playing a guitar.

A woman is slicing a vegetable.

A man is playing with a ball.

A woman is riding a horse.

A man is shooting a gun.

Relevant but incorrect descriptions

A dog is walking in a river.

A man is riding a bike.

A woman is slicing a piece of meat.

A boy is playing a flute.

A car is driving down a road.

A cat is playing with a toy.

A man is holding a baby monkey.

A boy is playing a soccer ball.

Incorrect descriptions

A woman is cleaning a sink.

A man is dancing.

A man is riding a horse.

A woman is riding a motorcycle.

Figure 4.13. Descriptions generated from our video embeddings.

content. The bottom row shows examples that the model generated descriptions irrelevant to the video input.

We evaluated generated sentences with the COCO description evaluation, *i.e.*, CIDEr, BLUE@4, and METEOR. While generated sentences got scores below the state-of-the-art video captioning method DVETS [111], we found that the BLEU and METEOR scores were comparable to other video captioning methods (Table 4.4). This indicates that our model efficiently encoded videos, maintaining their semantics. Moreover, this result suggests that our embeddings can be applied to other tasks that require joint representations of videos and sentences.

4. Summary

We developed neural network-based embedding models for video, sentence, and image inputs which fuse sentence and image representations. We jointly trained video and sentence embeddings using the YouTube dataset.

We demonstrated that our video embeddings can be used in an unsupervised video summarization approach that selects video clips based on the representativeness in the semantic embedding space. We observed that learned representations extracted from videos with similar content make clusters in the semantic space. In our approach, the input video is represented by deep representations in the semantic space, and clips corresponding to cluster centers are extracted to generate a video summary. By comparing our summaries to those created using VGG representations, we showed that the advantage of incorporating our cross-modal representations in video summarization. Furthermore, our results even outperformed the worst human created summaries.

Experiments for content-based video and sentence retrieval demonstrated the advantage of incorporating additional web images in sentence embedding and exhibited that our approach outperforms prior work in both video and sentence retrieval tasks. Furthermore, by decoding descriptions from video embeddings, we demonstrated that rich semantics of videos are efficiently encoded in our video embeddings. The future work includes the development of a video embedding that considers temporal structures of videos. We observed that some sentences result in irrelevant web images, and such sentences may not get advantages of web images. It would also be interesting to investigate which words can work as effective queries for image search. We also expect that filtering out web images in preprocessing of sentence embedding would improve the performance of sentence embedding.

Chapter 5

Representation Learning for Fine-grained Video Retrieval by Sentence Queries

1. Overview

In this chapter, we address to learn time-varying representations for content-based video retrieval (CBVR). The embedding models in Chapter 4 encode a video clip into one feature vector. However, that approach cannot capture the change of semantics along time. In order to represent the change of content within a video, we propose to produce a sequence of feature vectors as a video representation. We expect that this time-varying representation is helpful to model real-world videos such as movies or YouTube videos, which are long and consist of multiple video clips.

As in Chapter 4, we try to map both sentences and videos into a common embedding space, where a video is represented by a sequence of feature vectors. One interesting application of this representation is localizing content in a multi-clip video with a natural language query. Given a description, *e.g.*"She kisses his cheek", we would like to find corresponding short video clips from a long video (Figure 5.1). We call this task as fine-grained video retrieval (FGVR). In contrast to existing CBVR tasks, FGVR aims to handle more complex videos which may have multiple clips and varying content within a video. Thus, we

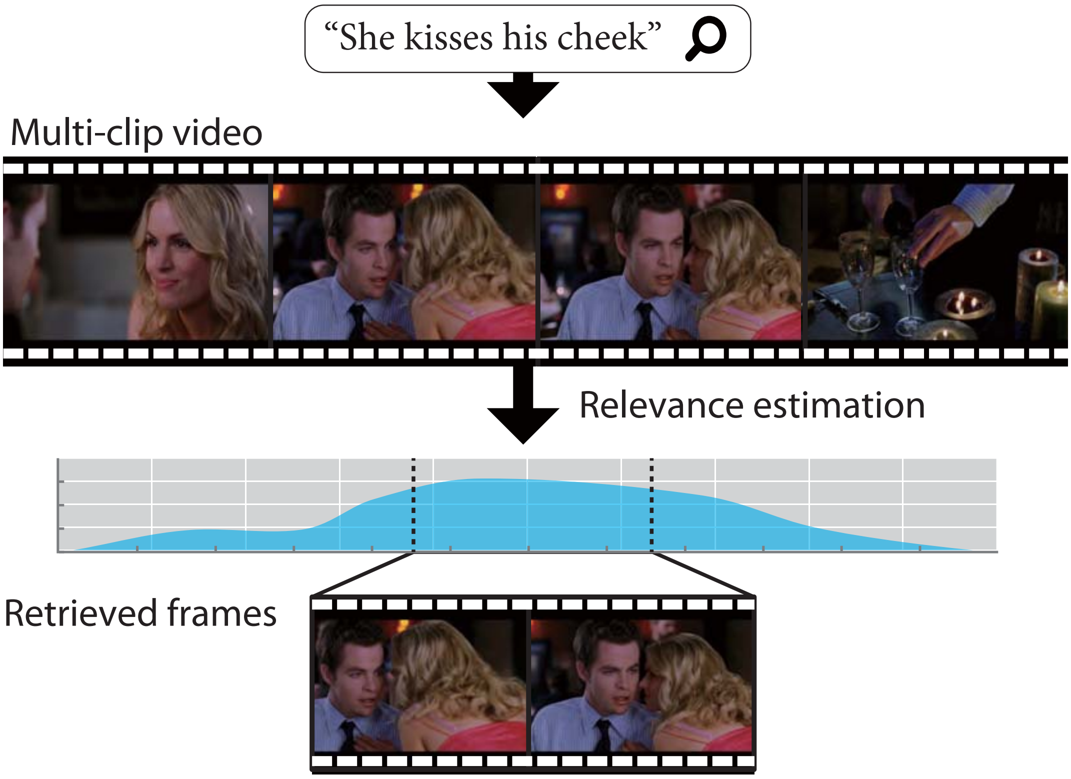


Figure 5.1. Given a natural language query, fine-grained video retrieval finds video frames which the query describes. An input video consists of multiple video clips.

expect that FGVR techniques contribute to a wide range of applications for real-world videos, for example, scene search from a lengthy video, and alignment of roughly annotated metadata and videos.

In this chapter, we describe representation learning by solving the FGVR task. We construct FGVR models that encode videos and sentences into cross-modal representations as in Chapter 4. The models are trained to localize video content which is semantically relevant to a sentence query. The task of FGVR works as strong supervision that makes a model encode time-varying semantics into a sequential representation, as well as, map videos and sentences into a cross-modal embedding space.

Most methods that temporally associate video content with languages, such as action localization, use frame-level labels indicating the start and the end point of the desired content. However, there do not exist many datasets that have multi-clip videos and sentences with temporal annotation. Making a dataset that is large enough to develop recent deep neural network models will require an immense amount of human intervention, thus we facilitate representation learning by synthesizing examples using existing datasets. While we do not have video-sentence datasets with temporal annotation, there are several large-scale datasets that provide videos and their descriptions only [106, 5, 77, 72]. We propose to compile a query sentence and a multi-clip video with temporal annotation from video-description datasets and a training scheme using the synthesized video-query pairs. As our data generation scheme can be applied to any video-description datasets, we can scale training datasets. Importantly, the experimental results demonstrate that our training scheme enables FGVR models to localize query-relevant content in real-world videos, while the models are trained on synthesized videos.

The contribution of this work is summarized as follows:

- We develop several neural network-based models, which produce time-varying representation from a video. The models are trained by solving FGVR tasks so that the model can associate video parts with natural language queries. We evaluate learned representations on the FGVR with two different datasets, which are built from YouTube videos and movies, respectively.

- We propose a new task of video retrieval, *i.e.*, FGVR. This task assumes that a video consists of multiple video clips, which may contain different objects, actions, and scenes. This assumption is more practical because most videos (online videos, broadcast programs, and movies) are edited and consist of multiple video clips.

- We propose to synthesize video and query pairs from existing datasets for video captioning. Our data generation scheme can build FGVR samples from any video captioning datasets. This enables large-scale training datasets, which are essential for developing deep neural network-based methods.

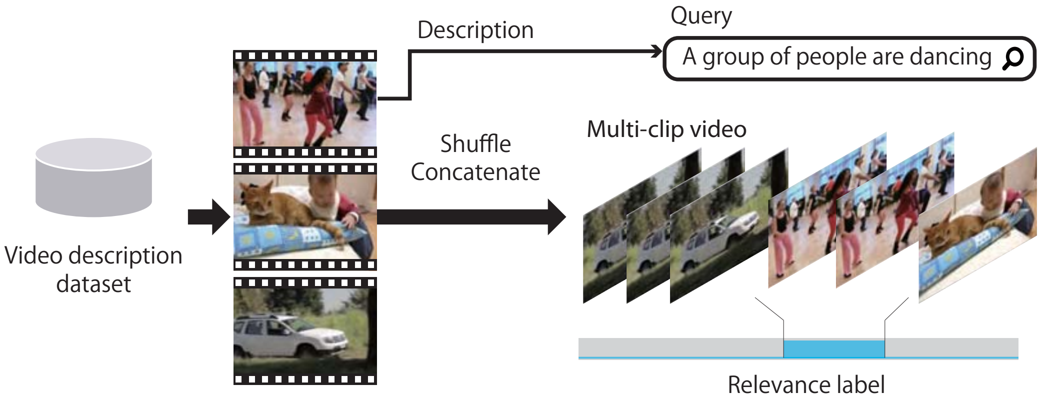


Figure 5.2. FGVR examples are generated from video-description datasets. A video clip associated with a description is combined with randomly sampled videos. This results in a multi-clip video and a sentence which describes only a part of the video.

2. Fine-grained Video Retrieval

2.1 Problem Statement

In the FGVR task, the input is a video consisting of multiple clips and a natural language query. The goal is to retrieve a subset of frames whose content is semantically relevant to the query (Figure 5.1). Specifically, given a sentence and video frames $V = \{v_1, \dots v_T\}$, where v_t is a visual feature extracted from the t-th frame, FGVR estimates relevance scores $R = \{r_1, \dots, r_T\}$ at each time step to retrieve frames. This task is similar to the video retrieval task for finding videos in a database which are relevant to a query. However, video retrieval tasks often implicitly assume that each video in the dataset is short and can be represented by a single query sentence. This assumption is not valid for most videos, *e.g.*, broadcast programs, movies, and even YouTube videos. A majority of these videos are lengthy and come with multiple concepts or scenes. The FGVR task relaxes this assumption; that is only a small part of the target video is relevant to a sentence query.

2.2 Data Generation

Training deep models usually requires large-scale datasets. Since there are no existing datasets for FGVR, we compile training examples for FGVR by extending the existing CBVR datasets. For FGVR examples, videos must 1) consist of multiple clips, 2) have corresponding query sentence related to a part of the video, and 3) be annotated with frame-level relevance labels. Since there is no dataset tailored for this task, we make video and query pairs from a large-scale video-description dataset, such as [106, 72].

The data generation using a video-description dataset is illustrated in Figure 5.2. To get a video consisting of multiple clips, we sample several video clips and their corresponding descriptions. We then choose one of the descriptions as a query sentence and concatenate the video clips in random order. Concatenation of multiple videos results in shot boundaries like most edited videos. The frames in a video clip corresponding to the selected query sentence are labeled as relevant frames, and other frames as irrelevant ones. By doing this, we can generate a number of videos where only a small part of it is relevant to a query sentence. Our data generation scheme can be applied to any dataset which provides videos and descriptions. This enables us to train FGVR methods on diverse videos provided by existing datasets.

2.3 Models for FVGR

We introduce several video embedding models that read video frames and produce a sequence of feature vectors. In order to cover possible models to capture content dynamics, we develop models with clip-level and frame-level video encoding. Each video clip or frame and a query sentence are mapped to a common feature space, and we estimate the relevance between them by computing the similarity of their representations in the feature space. In all methods, we employ the `pool5` layer of ResNet-50 [26] to extract visual features V from video frames.

2.4 Text Embedding Models

For text encoding, we employ two models that encode a sequence of words $\{w_1, \dots, w_N\}$ into a vector representation t, where w_n is a word embeddings.

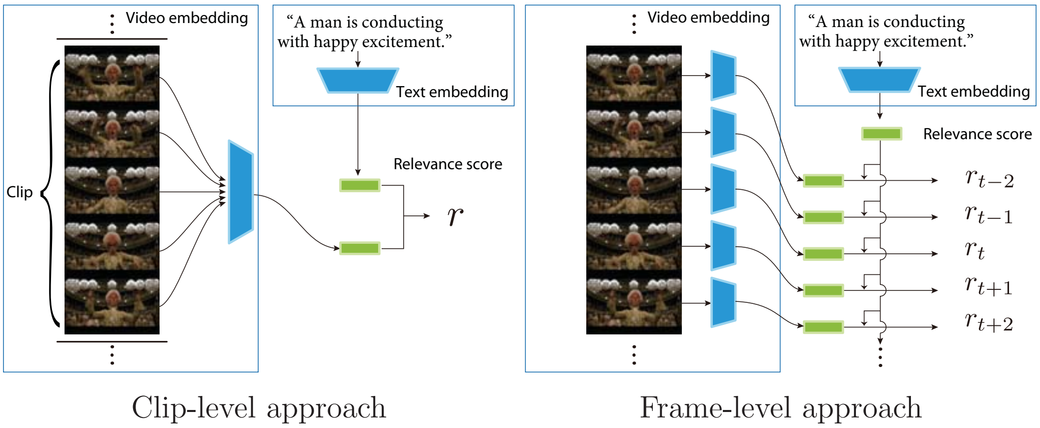


Figure 5.3. Illustration of clip-level (left) and frame-level (right) embedding models. Clip-level model summarizes frames in a clip and produces a feature vector. On the other hand, frame-level approach outputs a feature vector for every frame. These models are trained to localize video parts, which are semantically relevant to a query sentence.

One is the word pooling-based model (**W-Pool**). Input word embeddings are averaged to be transformed with a fully-connected layer as:

$$\tilde{w} = \sum_{n=1}^{N} w_n, \tag{5.1}$$

$$y = \tanh(W_{\text{wp}}\tilde{w} + b_{\text{wp}}), \tag{5.2}$$

where W_{wp} and b_{wp} are parameters of the fully-connected layer and y is a sentence representation.

The other is the word LSTM model (**W-LSTM**) that encodes a sequence of word embeddings with an LSTM layer, *i.e.*,

$$h_n, c_n = \text{LSTM}(w_n, h_{n-1}, c_{n-1}), \tag{5.3}$$

where h_n and c_n are a hidden state and a memory cell of the LSTM layer, respectively. We employ the last hidden state h_N as a representation of the sentence in the common feature space.

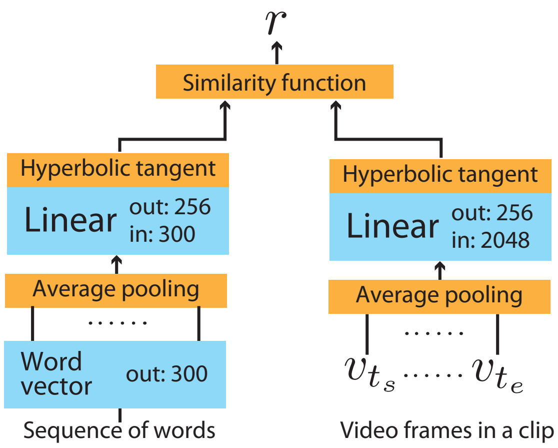


Figure 5.4. The architecture of F-Pool model for video clips and W-Pool model for sentences.

2.5 Dynamic Video Embedding Models

Clip-level Video Embedding

One possible approach is to divide an input video into short video clips and outputs a feature vector for each video clip as illustrated in Figure 5.3 (left). We call this approach as a clip-level approach. We test two temporal video segmentation for this approach: Ground truth video segmentation uses clip boundaries in synthesized videos, and uniform segmentation divides videos with a uniform interval. Similarly to [91], we implement two neural network models that take a sequence of frames $\{v_{t_s}, \ldots, v_{t_e}\}$ in a video clip as input and produce a vector representation x that summarizes the frames.

Frame pooling (F-Pool) summarizes the frames $\{v_{t_s}, \ldots, v_{t_e}\}$ in a video clip by average pooling. The averaged feature vectors are fed to a fully-connected layer. Therefore, the F-Pool model maps a video clip into the common feature

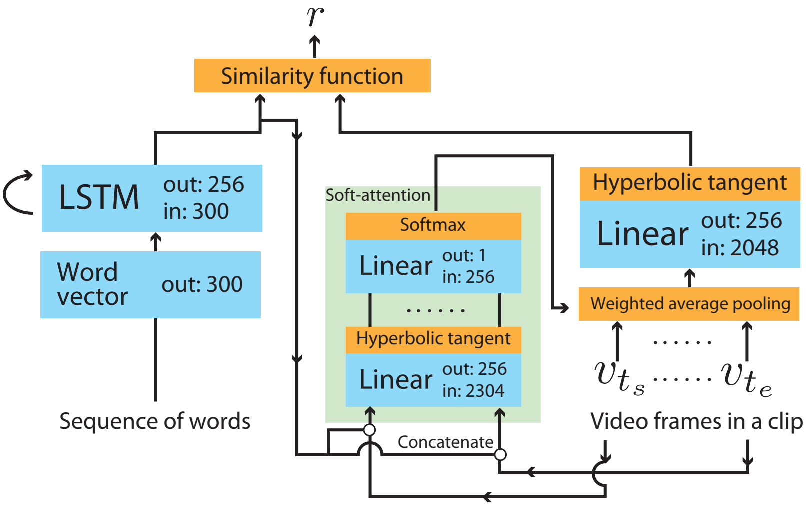


Figure 5.5. The architecture of WA model for video clips and W-Pool model for sentences.

space by

$$\tilde{v} = \sum_{i=t_s}^{t_e} v_i, \tag{5.4}$$

$$x = \tanh(W_{\text{fp}}\tilde{v} + b_{\text{fp}}), \tag{5.5}$$

where W_{fp} and b_{fp} are parameters of the fully-connected layer. Figure 5.4 illustrates this model. In this work, we set the unit size of the fully-connected layer to 256.

Weighted average (WA) incorporates the soft-attention mechanism [111] in frame pooling. The weights a_i of the frame v_i is computed based on the frame feature and a query sentence by

$$e_i = w_{\text{a}}^{\text{T}} \tanh(W_{\text{a}}[y, v_i] + b_{\text{a}}), \tag{5.6}$$

$$a_i = \exp(e_i)/\sum_{j=t_s}^{t_n} \exp(e_j), \tag{5.7}$$

where w_{a}, W_{a}, and b_{a} are learnable parameters, and $[\cdot, \cdot]$ denotes the concatenation of vectors. The vector y is a text embedding computed with a text encoding model

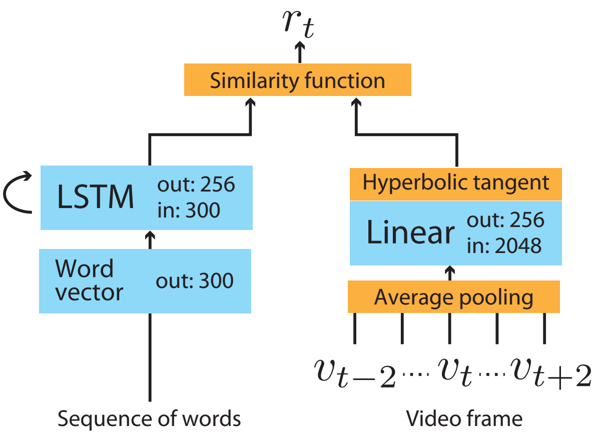


Figure 5.6. Sliding window (SW) model for video frame embedding.v_{t-2} and v_{t+2} indicate the first and the last frame in a temporal window, respectively.

described in Section 2.4. Using the weights, we obtain a weighted sum of frames and feed it to a fully-connected layer to get a clip representation x as:

$$\tilde{v}_{\mathrm{wa}} = \sum_{i=t_s}^{t_e} a_i v_i, \tag{5.8}$$

$$x = \tanh(W_{\mathrm{wa}} \tilde{v}_{\mathrm{wa}} + b_{\mathrm{wa}}), \tag{5.9}$$

where W_{wa} and b_{wa} are parameters of the fully-connected layer. Figure 5.5 shows the WA model with W-LSTM model for sentence embedding. Details including unit sizes are shown in Figure 5.5.

Frame-level Video Encoding

In the clip-level approach, an input video needs to be segmented beforehand; however, segment boundaries are not always available, and temporal video segmentation itself is still a challenging task. Another direction for this task is to read frames and produce a feature vector at each time step as in Figure 5.3 (right). For this approach, we implemented three models that encodes video frames to a sequence of vector representations $\{x_1, \ldots, x_T\}$.

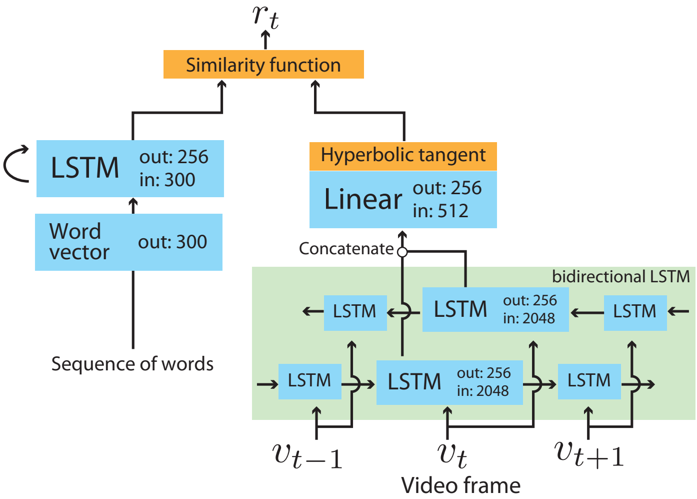


Figure 5.7. Bidirectional-LSTM (biLSTM) model for video frame embedding.

Sliding window (SW) This model reads an input frame sequence in the sliding window fashion. At each time step, we perform average pooling over frames within a temporal window and feed its output to a fully-connected layer in the same way as the F-Pool model. As shown in Figure 5.6, we set the temporal window size to 5 and the model reads frames with a stride of 1.

Bidirectional-LSTM (biLSTM) The biLSTM model utilizes a two-layer LSTM network that reads frames in forward and backward directions as in Figure 5.7. This bidirectional LSTM is employed in several recent works to model video frames [114, 28]. Hidden states at each time step are concatenated and transformed with a fully-connected layer as:

$$x_t = \tanh(W[h_t^{\text{forward}}, h_t^{\text{backward}}] + b), \tag{5.10}$$

where h_t^{forward} and h_t^{backward} are hidden states of the forward-LSTM and the backward-LSTM layers for the input frame v_t, respectively.

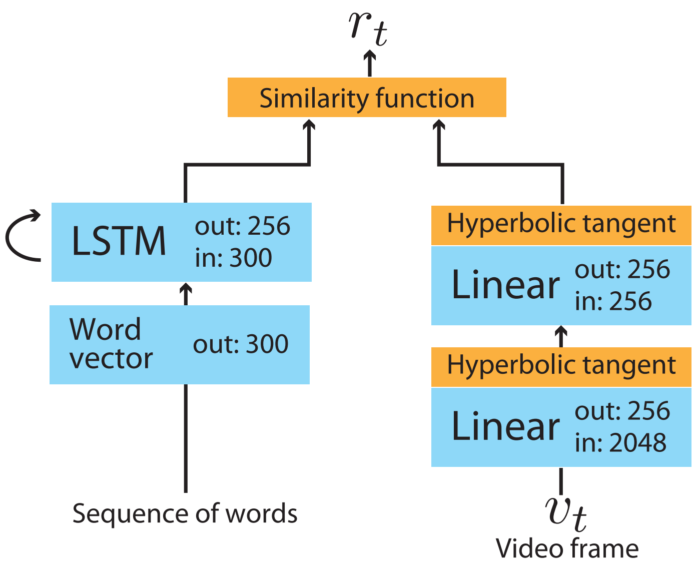


Figure 5.8. Fully-connected (FC) model for video frame embedding.

Fully-connected (FC) This model is a variation of the biLSTM model. We remove the temporal connection by replacing the bidirectional LSTM layers with a fully-connected layer as in Figure 5.8. Therefore, the input frame v_t is transformed by

$$\tilde{v}_t = \tanh(W_1 v_t + b_1), \tag{5.11}$$

$$x_t = \tanh(W_2 \tilde{v}_t + b_2), \tag{5.12}$$

where W_1, W_2, b_1, and b_2 are parameters of the fully-connected layers. This model estimates relevance scores in a frame-by-frame fashion. Therefore, this model is equivalent to frame-level CBVR.

2.6 Similarity Metrics for Relevance Score

After a vector representations for clips or frames are obtained, relevance scores $R = \{r_1, \dots, r_T\}$ are computed. In this study, we test cosine similarity and partial order similarity [91]. With cosine similarity, relevance scores are computed as:

$$r_t = \frac{x_t \cdot y}{\|x_t\| \|y\|}. \tag{5.13}$$

Partial order similarity between two vectors is computed as:

$$r_t = -\| \max(y - x_t, 0) \|^2, \tag{5.14}$$

where u_0 and u_1 are non-negative vectors. Therefore, we compute the absolute values of the outputs of models and apply L2-normalization before computing the partial order similarity. Note that partial order similarity is not order-invariant.

2.7 Training

We train the models described in Section 2.3 using video-sentence pairs synthesized as in Section 2.2. The models for videos and sentences are jointly trained so that the query relevance scores of relevant frames are larger than those of others. We compute an averaged score of relevant and irrelevant frames and update the model to make the difference between the scores larger. During the training, a model is trained by minimizing the loss computed from predicted relevance score $R = \{r_1, \ldots, r_T\}$ and ground truth label $L = \{l_1, \ldots, l_T\}$ as:

$$\text{Loss}(R, L) = \max(-R_{\text{pos}} + R_{\text{neg}} + \mu, 0), \tag{5.15}$$

$$R_{\text{pos}} = \frac{1}{N_{\text{pos}}} \sum_{t=1}^{T} l_t r_t, \tag{5.16}$$

$$R_{\text{neg}} = \frac{1}{N_{\text{neg}}} \sum_{t=1}^{T} (1 - l_t) r_t, \tag{5.17}$$

where N_{pos} and N_{neg} are the number of relevant and irrelevant frames in a video, respectively. l_t is a label representing the frame's relevance/irrelevance to a query sentence and r_t is relevance score, which is computed as in Section 2.6. We set $l_t = 1$ if the frame is relevant, and otherwise 0. The parameter μ is a predefined margin to penalize the smaller difference between the averaged score of relevant and irrelevant frames than the margin. Models of the clip-level approach do not produce frame-level scores, thus we spread a clip-level score to all frames in the clip.

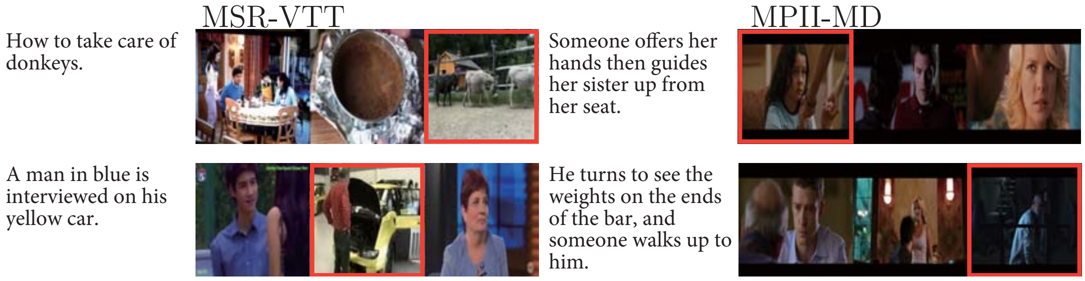

MSR-VTT
MPII-MD
How to take care of donkeys.
A man in blue is interviewed on his yellow car.
Someone offers her hands then guides her sister up from her seat.
He turns to see the weights on the ends of the bar, and someone walks up to him.

Figure 5.9. Query sentence and video pairs, where only keyframes are displayed for videos. The videos are composed by combining multiple video clips from existing video-description datasets. The examples in the left are built from MSR-VTT, and the right column from MPII-MD. The red boxes indicate the frames corresponding to the query sentence.

3. Experiments

We evaluated learned representations on the task of FGVR. We generated FGVR examples from two datasets, MSR Video to Text (MSR-VTT) [106] and the MPII Movie Description dataset (MPII-MD) [73], and investigate the performance of each model.

3.1 Implementation Detail

The model was trained in an end-to-end manner with stochastic gradient decent with the mini-batch size of 100. We used Adam [38] for optimization with the initial learning rate 10^{-3} for MSR-VTT and 10^{-4} for MPII-MD. In all experiments, models were trained for 15 epochs, and we employed a model at the minimum loss on the validation split. During training, we halved the learning rate at the 10th epoch. We adopted gradient clipping with threshold 10.0 and weight decay with weight 0.0005 for MPII-MD. We set the parameter μ for the loss function to 1.0. To extract video frame features, we utilized ResNet-50 pretrained on ImageNet [26]. The word embeddings were initialized with word vectors by [58], which we empirically found helpful for training. We set the output size of video and text encoding models to 256. The window size of SW was 5, and input videos were padded with zeros to keep the output length the same as the number of input video frames. Both of the bidirectional LSTM layers in the biLSTM model have

Table 5.1. Statistics of the original datasets

Dataset	Domain	# video	Avr. duration	# sentence
MSR-VTT	open	10,000	15s	200,000
MPII-MD (LSMDC'16)	movie	118,507	4s	118,507

256 units, and the output vectors were fed to the fully connected layers whose output size was also 256.

3.2 Datasets

We tested video and sentence encoding models on the MSR-VTT and the MPII-MD datasets. Examples of generated video and query pairs are displayed in Figure 5.9. The MSR-VTT dataset includes 10,000 YouTube video clips, and 20 descriptions are annotated for each video clip. MPII Movie Description dataset has 118,507 video clips from movies, and each video clip is annotated with one description. For the MSR-VTT dataset, we used training and test splits provided by the MSR-VTT official web page. For the MPII-MD dataset, we used splits for the LSMDC'16 movie annotation and retrieval task [91]. Word vocabulary is collected from descriptions in the training split. The descriptions were normalized by punctuation removal and lowercasing, then we compiled a vocabulary dictionary by sampling words occurring more than three times in training queries, which results in 8,935 words for the YouTube dataset and 10,066 words for the movie dataset. The videos were down-sampled at 5 fps and rescaled to 244×244. During training, we sampled two video clips for each video-description pair to create FGVR examples as in Figure 5.2. Since most video clips in the datasets have similar durations, which can be a strong prior, the first and the last few seconds of video clips are randomly trimmed so that the video clips have 20-100% of their original length. The average durations of videos compiled from the MSR-VTT dataset is 32 seconds and those from the MPII-MD dataset is 8.6 seconds.

We report statistics of the datasets in Table 5.1. Figure 5.10 shows that all videos compiled from the MSR-VTT dataset are within 10-80 seconds, and the MPII-MD dataset mainly has shorter videos, it also has quite long videos whose

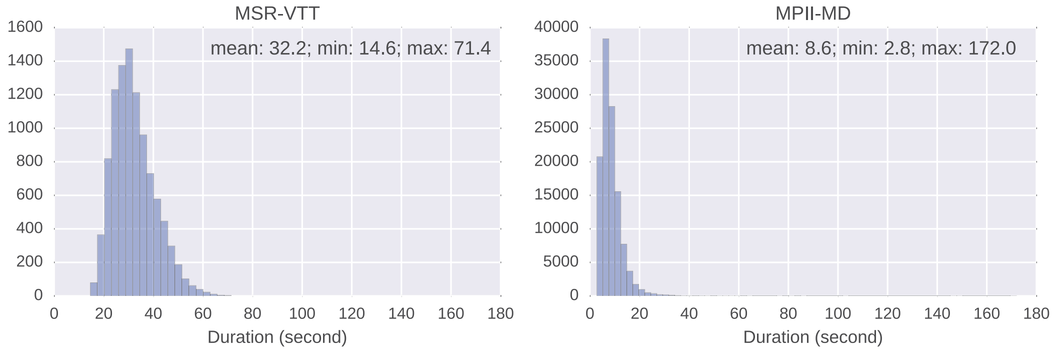


Figure 5.10. The distribution of synthesized videos' durations. The horizontal axis represents durations, and the vertical axis is the number of videos.

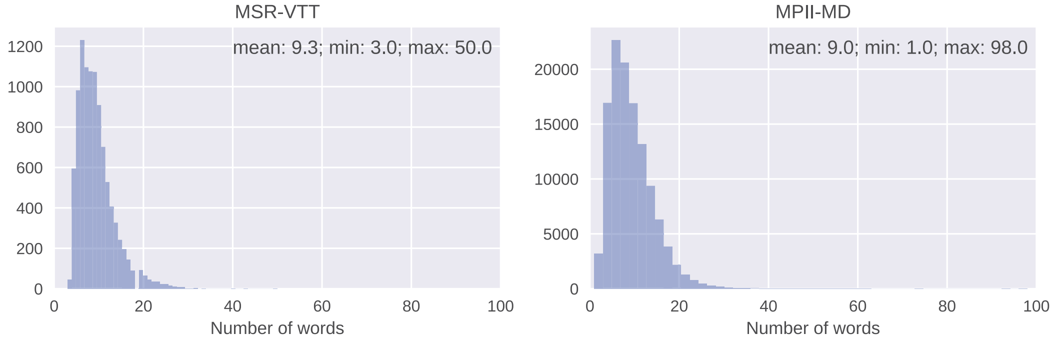


Figure 5.11. The distribution of the number of words in a sentence. The horizontal axis represents the number of words, and the vertical axis is the number of videos.

durations are more than 100 seconds. Note that our video generation incorporates random trimming, so durations of resulting videos are not static. As can be seen in Figure 5.11, the MPII-MD dataset includes some long sentences that have more than 50 words. These sentences tend to describe complex scenes.

3.3 Qualitative Evaluation

We show some examples of relevance scores predicted by a model trained for the FGVR task. Figure 5.12 shows an example of frame-level scores for different queries by the biLSTM model. The video shown in Figure 5.12 was generated

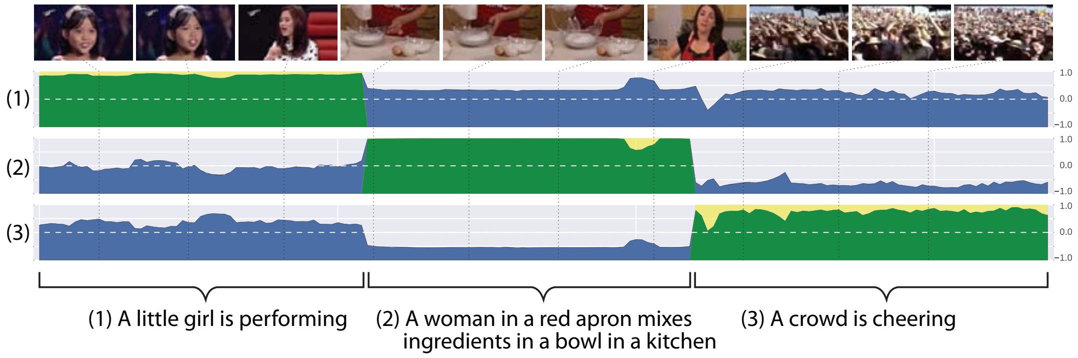


Figure 5.12. Relevance scores of a multi-clipped video for different queries. The horizontal axis represents time. From top to bottom: scores for queries (1), (2), and (3). Blue represents relevance scores and yellow ground truth relevance labels. Overlapping areas are thus green.

from the MSR-VTT dataset. The yellow areas indicate ground truth video clips corresponding to the query sentence. For the query sentences (1) and (2), the model predicted high relevance scores for corresponding frames. Interestingly, for the query (3), frames of a girl with a microphone got high scores as well as the ground truth frames of a crowd. This might be caused by the crowd behind the girl. Within a video clip, we can observe that relevance scores varied according to the content of the frame, *e.g.*, frame without the cooking tools are less relevant than other frames for the query (2). While the models were trained on videos that had only one relevant part, the resulting model gave high scores for multiple parts. This suggests that the training scheme, which uses synthesized videos, does not constrain test videos to have the same structure as training videos, *i.e.*, trained models can be reused to videos with multiple relevant parts.

Relevance scores produced by the baseline models for a video and query pair are shown in Figure 5.13. The top row result was produced by a model with W-Pool text encoding model and others use models with W-LSTM text encoding models. All models in the examples use cosine similarity. For the clip-level approaches (F-Pool and WA), we used the ground truth video clip boundaries. The input video is a short movie excerpt from the MPII-MD dataset. While the input video is generated in a different way from the training scheme (*i.e.*, using randomly sampled video clips), the models of the example still detect relevant

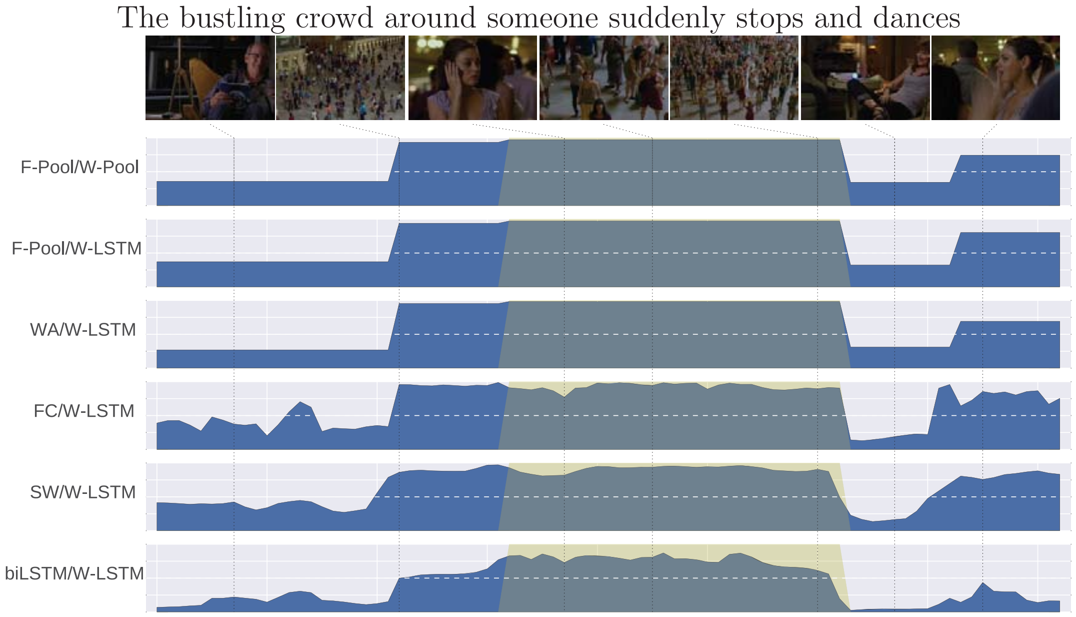
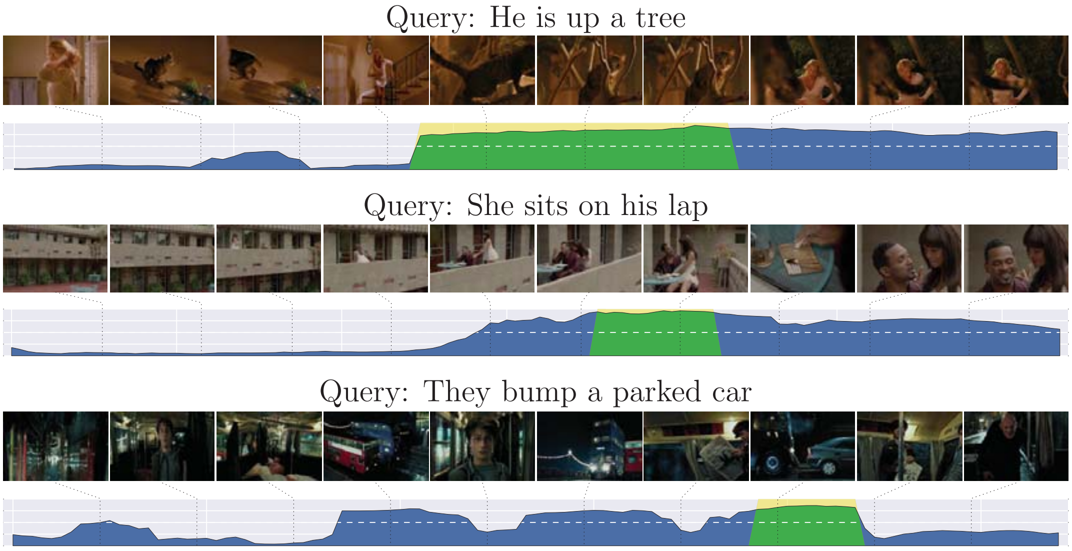


Figure 5.13. Examples of relevance scores by different models. The input video is five successive video clips from a movie in the MPII-MD dataset.

parts in a video. This suggests that the baseline models trained on the synthesized videos can be reused for real-world videos. The three models of the clip-level approaches, which are in the top three rows, do not show significant differences in predicted scores. Compared to the FC model, those with temporal connection produce smoother relevance scores, which may be preferable because relevance scores are not likely to change frequently in most videos.

We show more examples of the biLSTM model on short movie excerpts to demonstrate that the model can be used for real-world videos (see Figure 5.14). The ground truth parts are indicated by yellow areas in the figure. Note that the ground truth labels are based on where the query sentence is originally annotated in a video captioning dataset, and other frames can also be relevant to the query. Moreover, the start and end points of a specific event are ambiguous, especially in movies. The examples of top two rows show that the biLSTM can roughly localize content relevant to the queries. However, the biLSTM model failed by giving high scores for irrelevant frames in the bottom row. The input video of this example shows a dark scene, and the scene changes rapidly. Moreover,



Query: He is up a tree

Query: She sits on his lap

Query: They bump a parked car

Figure 5.14. Examples of relevance score estimation by the biLSTM model on movie videos.

the video has unusual events since it is a fantasy movie. We assume that these characteristics of the input video made it difficult to capture the video content.

3.4 Quantitative Evaluation

We conducted a quantitative evaluation of predicting relevant frames from multi-clipped videos on the MSR-VTT and the MPII-MD datasets. We generated test videos in the same way as in Section 2.2 from test splits of the datasets. For each test sample, we computed frame-level relevance scores of a video to a query sentence, and then evaluated the performance with average precision (AP). We report the mean and the standard deviation (the values in parenthesis) of the AP scores over all test samples in Table 5.2. To compute AP, the clip-level scores were transformed to frame-level scores by simply spreading the clip-level score to all frames in the clip. The scores obtained by random score prediction are reported in the bottom row.

Overall, cosine similarity performs better than partial order similarity in this task. For clip-level approaches, there are no significant differences between models. Note that these scores with ground truth clip boundaries (GT) can be re-

Table 5.2. Mean average precision (AP) scores (%) of FGVR. GT denotes ground truth clip boundaries, and UNI denotes uniform segmentation.

video model /	clip	MSR-VTT		MPII-MD	
sentence model	boundaries	cosine	p-order	cosine	p-order
F-Pool / W-Pool	GT	**86.5 (27.9)**	80.9 (31.5)	**77.7 (33.2)**	73.6 (34.8)
	UNI	81.1 (22.5)	76.0 (25.2)	74.4 (26.6)	70.7 (27.4)
F-Pool / W-LSTM	GT	85.4 (28.7)	79.2 (32.3)	74.8 (34.3)	69.0 (35.8)
	UNI	80.1 (23.1)	75.9 (25.3)	72.5 (27.6)	68.2 (28.4)
WA / W-LSTM	GT	86.4 (28.0)	75.9 (33.7)	75.8 (34.0)	69.0 (35.9)
	UNI	79.7 (23.2)	71.0 (26.7)	72.6 (27.4)	67.4 (28.5)
FC / W-LSTM	—	80.9 (23.7)	75.7 (25.2)	73.1 (27.7)	63.3 (27.6)
SW / W-LSTM	—	83.3 (22.9)	76.3 (25.7)	73.5 (27.9)	69.8 (28.8)
biLSTM / W-LSTM	—	**83.8 (22.7)**	72.5 (25.7)	**76.1 (28.9)**	61.7 (26.5)
by chance	—	47.0 (12.2)		49.4 (17.6)	

garded as a sort of upper bounds of the clip-level approaches. We also report scores obtained by uniformly dividing an input video into three clips (UNI). These results suggest that the performance of clip-level FGVR methods highly relies on temporal video segmentation.

We can also observe that the frame-level approach (FC, SW, and biLSTM models), which does not require temporal video segmentation, achieves good retrieval performance on the MSR-VTT dataset. This suggests that video segmentation is not necessary for FGVR. From the comparison between models for the frame-level approach, we can see that incorporating nearby frames improves the performance. This might be because context obtained from other frames is helpful to understand a video content.

For the MPII-MD dataset, all baselines resulted in lower scores. As videos and sentences in the dataset are more challenging as shown in Figure 5.9. Many of the sentences often describe complex scenes, of which LSTM may have difficulties in encoding the semantics. Moreover, movies often have dark and low-contrast scenes, which may cause failures in understanding the visual content.

4. Summary

In this work, we propose to learn sequential vector representation for videos to encode dynamics of content within a video. Our video embedding model and sentence embedding model are jointly trained by solving the FGVR task to localize video content according to a query sentence, which is a new video retrieval task. This task is based on the idea that developing video retrieval methods to handle untrimmed videos consisting of multiple clips is important for real-world applications. For this task, we present a data generation scheme to scale training video-query pairs. We introduce two lines of approaches, *i.e.*, clip-level and frame-level approaches, and implemented video and sentence embedding models for this task.

In experiments, we present results on two evaluation datasets, which are built from a YouTube video dataset and a movie dataset. The experimental results suggest that the clip-level approach can be improved by leveraging sophisticated video segmentation methods. We also observed that considering temporal context by the sliding window fashion or temporal connections between frames helps to encode video frames. The FGVR results on some videos from movies suggest that our approach can retrieve video parts from real-world videos although our models are trained on generated video-query pairs. We expect that text embedding methods that can handle long sentences, which have complex semantics, will be a key component for further improvement. An FGVR task on manually edited videos, *e.g.*, retrieving a scene from a movie, is a challenging and important topic. We will explore FGVR on manually created videos by modifying such video and text alignment datasets as [72, 77].

Chapter 6

Conclusion

This dissertation has proposed several cross-modal representations for videos and languages. Evaluating the performance of representation is unclear; thus we investigate how our cross-modal representations work in practical tasks, such as video summarization, video captioning, and content-based video retrieval.

We have explored two approaches for developing cross-modal representations. One is manually designing a cross-modal representation for videos and languages, as well as their similarity metric. The other approach is data-driven representation learning. While this approach requires large-scale training data, the approach can automatically learn cross-model representations from data.

Chapter 3 has explored the former approach. We designed object-focused representations for videos and text so that the representation provides a rough idea about events presented in videos and text. We implemented a query-focused video summarization method, which makes a video summarization according to the content of user text. We observed that the semantic similarity metric built upon our object-focused representation is useful for picking out video parts so that a resulting summary has more events relevant to user text. The user study also suggests that users prefer video summaries generated by the proposed video summarization method that creates an output video summary based on the input text.

Chapter 4 proposed data-driven representation learning for videos and sentences. In this work, we developed deep models that map videos and sentences into a common feature space. Moreover, we proposed to extend the sentence encoding model by incorporating web images. We investigated the performance of learned representations in video summarization and content-based video and sentence retrieval. The experiment of unsupervised video summarization suggests that simply replacing visual representation with our cross-modal representation may improve the quality of video summaries. It is also observed that the use of web images in sentence encoding helps to retrieve more relevant items in the experiment of content-based video and sentence retrieval. This result indicates that web images help to disambiguate the semantics of an input sentence.

Video encoding model is extended to capture dynamics within a video in Chapter 5. This model is motivated by a new content-based video retrieval task to find video parts relevant to a query sentence. We proposed to synthesize video-query pairs from video captioning datasets, which can be used to train video and sentence encoding models by solving the fine-grained video retrieval (FGVR). The experimental results suggest that our models can localize content in real-world videos with a natural language query although our models are trained on synthesized examples.

To summarize, we proposed cross-modal representations for videos and natural languages. It is observed that these representations are capable of encoding sentence-level semantics of videos, which is comprised of combinations of various concepts including objects, actions, scenes, *etc.* These representations are validated in practical applications. The experimental results demonstrated that use of our representations can improve the performance on the task of unsupervised video summarization and content-based video and sentence retrieval. Furthermore, we proposed text-focused video summarization and FGVR, which are novel applications using our cross-modal representation.

Cross-modal representations that associate videos and languages enable computers to behave as if they understand videos. However, video and language understanding is an extremely complex process, and only a small part of the process is explored in the research community. Many recent works encode videos and sentences with deep models and they seem to capture diverse concepts including objects, actions, and attributes. However, these large and complex models also make it difficult to know what factors of semantics are captured. One of the next challenges lies in clarifying the limitation of current deep representations, and how we can extend the capability to encode rich semantics. For example, current techniques hardly handle long temporal relationships within a video, such as temporal order and causality of events. The temporal relationships are important factors to understand the complicated semantics of videos, and modeling them must be an important future direction.

One significant criticism for recent video understanding research is that settings of many video understanding tasks including content-based video retrieval and captioning do not involve temporal reasoning. Many of these tasks can be often solved by focusing on a single keyframe. For further improvement of cross-modal representations, it is insightful to explore novel tasks or applications that require temporal reasoning. There are several emerging applications, such as video question answering [88, 32, 60], and dense video captioning [41]. Video question answering requires to find some events in a long video and answer a question in natural language, and dense video captioning produces several descriptions aligned with parts of a long video. These are challenging tasks, and addressing these tasks will lead to a new framework to associate videos and languages.

Acknowledgements

The work in this dissertation was completed during my Ph.D. course at the Graduate Shool of Information Science, Nara Institute of Science and Technology.

First I would like to thank Professor Kiyoshi Kiyokawa, who is the chief of my thesis committee. He also supported me to start a research project. I must thank Dr. Naokazu Yokoya, who is my former supervisor and a member of the thesis committee. He also supervised me throughout my Ph.D. course. He offered insightful discussions and advice, which shaped my research. He also gave me a lot of support when I visited University of Oulu. When I wrote research proposals, he gave feedbacks and advice. I am very grateful to Dr. Yokoya for teaching and encouraging me.

I am also grateful to Professor Hirokazu Kato, who kindly undertook the committee member of this dissertation. I would like to thank Associate Professor Tomokazu Sato, who is currently Professor of Shiga University. He encouraged and guided me throughout my Ph.D. course. I am deeply grateful to Associate Professor Yuta Nakashima. This work would not be completed without his patient teaching for five years. I would like to thank Assistant Professor Norihiko Kawai who inspired me with many discussions. I am also like to thank members at Computer Vision and Media Computing Laboratory and Cybernetics Reality Engineering Laboratory with whom I had the pleasure of working and learning together.

I am also grateful to Professor Petri Pulli. Visiting his laboratory was a valuable experience for a student who just started her research. I would like to express my appreciation to Professor Janne Heikkilä and Professor Esa Rahtu, who kindly accepted my one-year visit to the University of Oulu and continuous collaboration after the stay through many remote meetings. The representation learning described in Chapters 4 and 5 in this dissertation is based on the joint work with them. It has been my great honor working with the two great researchers.

I also like to thank members of Institute of Datability Science at Osaka University. I have been very fortunate to work with energetic researchers there. Their work and discussions inspired me a lot.

Finally, I would like to thank my family. Their understanding and support have always been helpful during my Ph.D. course.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. YouTube-8M: A large-scale video classification benchmark. *arXiv preprint, arXiv:1609.08675*, 10 pages, 2016.

[2] Kiyoharu Aizawa, Kenichiro Ishijima, and Makoto Shiina. Summarizing wearable video. In *International Conference on Image Processing (ICIP)*, pages 398–401, 2001.

[3] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 73–80, 2010.

[4] Noboru Babaguchi, Yoshihiko Kawai, Takehiro Ogura, and Tadahiro Kitahashi. Personalized abstraction of broadcasted American football video by highlight selection. *IEEE Transactions on Multimedia*, 6(4):575–586, 2004.

[5] David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL)*, pages 190–200, 2011.

[6] X. Chen, H. Fang, TY Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 7 pages, 2015.

[7] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 539–546, 2005.

[8] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):5:1–5:60, 2008.

[9] Daniel DeMenthon, Vikrant Kobla, and David Doermann. Video summarization by curve simplification. In *ACM International Conference on Multimedia (MM)*, pages 211–218, 1998.

[10] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *IEEE International Conference on Machine Learning (ICML)*, pages 647–655, 2014.

[11] Naveed Ejaz, Irfan Mehmood, and Sung Wook Baik. Efficient visual attention based framework for extracting key frames from videos. *Signal Processing: Image Communication*, 28(1):34–44, 2013.

[12] Georgios Evangelopoulos, Athanasia Zlatintsi, Alexandros Potamianos, Petros Maragos, Konstantinos Rapantzikos, Georgios Skoumas, and Yannis Avrithis. Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention. *IEEE Transactions on Multimedia*, 15(7):1553–1568, 2013.

[13] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K. Srivastava, Li Deng, Piotr Dollar, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig. From captions to visual concepts and back. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1473–1482, 2015.

[14] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *European Conference on Computer Vision (ECCV)*, pages 15–29, 2010.

[15] Brendan J. Frey and Dueck Delbert. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[16] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc\textquotesingle Aurelio Ranzato, and Tomas Mikolov. DeViSE: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2121–2129. 2013.

[17] A. Gaidon, Z. Harchaoui, and C. Schmid. Temporal localization of actions with actoms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(11):2782–2795, 2013.

[18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[19] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2069–2077, 2014.

[20] Yihong Gong and Xin Liu. Video summarization using singular value decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 174–180, 2000.

[21] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5842–5850, 2017.

[22] Sergio Guadarrama, Subhashini Venugopalan, U T Austin, Niveda Krishnamoorthy, Raymond Mooney, Girish Malkarnenkar, Trevor Darrell, and U C Berkeley. YouTube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *International Conference on Computer Vision (ICCV)*, pages 2712–2719, 2013.

[23] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, Fabian Nater, and Luc van Gool. The interestingness of images. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1633–1640, 2013.

[24] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc van Gool. Creating summaries from user videos. In *European Conference on Computer Vision (ECCV)*, pages 505–520, 2014.

[25] Michael Gygli, Helmut Grabner, and Luc Van Gool. Video summarization by learning submodular mixtures of objectives. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3090–3098, 2015.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[27] Richang Hong, Jinhui Tang, Hung-Khoon Tan, Chong-Wah Ngo, Shuicheng Yan, and Tat-Seng Chua. Beyond search: Event-driven summarization for web videos. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 7(4):1–18, 2011.

[28] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Kazuhiro Sumi, John R. Hershey, and Tim K. Marks. Attention-based multimodal fusion for video description. In *International Conference on Computer Vision (ICCV)*, pages 4193–4202, 2017.

[29] Yongtao Hu, Jimmy SJ. Ren, Jingwen Dai, Chang Yuan, Li Xu, and Wenping Wang. Deep multimodal speaker naming. In *ACM International Conference on Multimedia (MM)*, pages 1107–1110, 2015.

[30] Chun Rong Huang, Huai Ping Lee, and Chu Song Chen. Shot change detection via local keypoint matching. *IEEE Transactions on Multimedia*, 10(6):1097–1108, 2008.

[31] Mihir Jain, Jan van Gemert, Herve Jegou, Patrick Bouthemy, and Cees G.M. Snoek. Action localization with tubelets from motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 740–747, 2014.

[32] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: Toward spatio-temporal reasoning in visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2758–2766, 2017.

[33] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(1):221–231, 2013.

[34] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. DenseCap: Fully convolutional localization networks for dense captioning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4565–4574, 2016.

[35] A Karpathy, G Toderici, S Shetty, T Leung, R Sukthankar, and L Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, 2014.

[36] Andrej Karpathy, Armand Joulin, and Li Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1889–1897. 2014.

[37] Aditya Khosla, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan. Large-scale video summarization using web-image priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2698–2705, 2013.

[38] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representation(ICLR)*, 11 pages, 2015.

[39] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3276–3284, 2015.

[40] Alexander Kläser, Marcin Marszałek, Cordelia Schmid, and Andrew Zisserman. Human focused action localization in video. In *European Conference on Trends and Topics in Computer Vision*, pages 219–233, 2012.

[41] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *IEEE International Conference on Computer Vision (ICCV)*, pages 706–715, 2017.

[42] A Krizhevsky, I Sutskever, and Ge Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[43] Robert Laganière, Raphael Bacco, Arnaud Hocevar, Patrick Lambert, Grégory Païs, and Bogdan E. Ionescu. Video summarization from spatio-temporal features. In *ACM TRECVid Video Summarization Workshop*, pages 144–148, 2008.

[44] Duy-Dinh Le, Sang Phan, Nguyen Vinh-Tiep, Renoust Benjamin, Tuan A. Nguyen, Van-Nam Hoang, Thanh Duc Ngo, Minh-Triet Tran, Yuki Watanabe, Martin Klinkigt, Atsushi Hiroike, Duc A. Duong, Yusuke Miyao, and Shin'ichi Satoh. NII-HITACHI-UIT at TRECVID 2016. In *TRECVID Workshops*, 25 pages, 2016.

[45] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning (ICML)*, pages 1188–1196, 2014.

[46] Joonseok Lee and Sami Abu-El-Haija. Large-scale content-only video recommendation. In *IEEE International Conference on Computer Vision (ICCV) Workshops*, pages 987 —- 995, 2017.

[47] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1346–1353, 2012.

[48] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 420–429, 2007.

[49] Shuang Li, Tong Xiao, Hongsheng Li, Bolei Zhou, Dayu Yue, and Xiaogang Wang. Person search with natural language description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1970–1979, 2017.

[50] Yingbo Li, Bernard Merialdo, and Sophia Antipolis. VERT: Automatic evaluation of video summaries. In *ACM International Conference on Multimedia (MM)*, pages 851–854, 2010.

[51] Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2657–2664, 2014.

[52] Tsung-Yi Lin, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5007–5015, 2015.

[53] Zheng Lu and Kristen Grauman. Story-driven summarization for egocentric video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2714–2721, 2013.

[54] Yf Ma, Lie Lu, Hj Zhang, and Mingjing Li. A user attention model for video summarization. In *ACM International Conference on Multimedia (MM)*, pages 533–542, 2002.

[55] Tegan Maharaj, Nicolas Ballas, Anna Rohrbach, Aaron Courville, and Christopher Pal. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. *arXiv preprint, arXiv:1611.07810*, 9 pages, 2016.

[56] Stephen Maybank. A survey on visual content-based video indexing and retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):797–819, 2011.

[57] Ryan Mcdonald. A study of global inference algorithms in multi-document summarization. In *European Conference on Information Retrieval (ECIR)*, pages 557–564, 2007.

[58] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119, 2013.

[59] Arthur G. Money and Harry Agius. Video summarisation: A conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation*, 19(2):121–143, 2008.

[60] Jonghwan Mun, Paul Hongsuck Seo, Ilchae Jung, and Bohyung Han. MarioQA: Answering Questions by Watching Gameplay Videos. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2867–2875, 2017.

[61] Seil Na, Sangho Lee, Jisung Kim, and Gunhee Kim. A read-write memory network for movie story understanding. In *IEEE International Conference on Computer Vision (ICCV)*, pages 677–685, 2017.

[62] Yuta Nakashima and Naokazu Yokoya. Inferring what the videographer wanted to capture. In *IEEE International Conference on Image Processing (ICIP)*, pages 191–195, 2013.

[63] Chong Wah Ngo, Yu Fei Ma, and Hong Jiang Zhang. Video summarization and scene detection by graph modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(2):296–304, 2005.

[64] Cuong Nguyen, Yuzhen Niu, Feng Liu, Arthur G. Money, and Harry Agius. Video summagator: An interface for video summarization and navigation. In *ACM Conference on Human Factors in Computing Systems (SIGCHI)*, volume 19, pages 3–6, 2012.

[65] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. Learning joint representations of videos and sentences with web image search. In *European Conference on Computer Vision (ECCV) Workshops*, pages 651–667, 2016.

[66] Federico Perazzi, Philipp Krahenbuhl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 733–740, 2012.

[67] Bryan Plummer, Matthew Brown, and Svetlana Lazebnik. Enhancing video summarization via vision-language embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5781–5789, 2017.

[68] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *European Conference on Computer Vision (ECCV)*, pages 540–555, 2014.

[69] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 139–147, 2010.

[70] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. YouTube-BoundingBoxes: A large high-precision human-annotated data set for object detection in video. *arXiv preprint, arXiv:1702.00824*, 11 pages, 2017.

[71] Joseph Redmon and Ali Farhadi. YOLO9000: Better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7263–7271, 2017.

[72] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3202–3212, 2015.

[73] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision*, 123(1):94–120, 2017.

[74] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. Translating video content to natural language descriptions. In *IEEE International Conference on Computer Vision (ICCV)*, pages 433–440, 2013.

[75] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[76] Jitao Sang and Changsheng Xu. Character-based movie summarization. In *ACM International Conference on Multimedia (MM)*, pages 855–858, 2010.

[77] Anna Senina, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. Coherent multi-sentence video description with variable level of detail. In *German Conference on Pattern Recognition (GCPR)*, pages 184–195, 2014.

[78] Aidean Sharghi, Boqing Gong, and Mubarak Shah. Query-focused extractive video summarization. In *European Conference on Computer Vision (ECCV)*, pages 3–19, 2016.

[79] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in Homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision (ECCV)*, pages 510–526, 2016.

[80] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recoginition. In *International Conference on Learning Representation(ICLR)*, 14 pages, 2015.

[81] Cees G. M. Snoek and Marcel Worring. Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 2(4):215–322, 2009.

[82] Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems (NIPS)*, pages 935–943, 2013.

[83] Yale Song. Real-time video highlights for yahoo esports. In *Neural Information Processing Systems (NIPS) Workshops*, 5 pages, 2016.

[84] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. TV-Sum: Summarizing web videos using titles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5179–5187, 2015.

[85] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[86] Hiroya Takamura and Manabu Okumura. Text summarization model based on maximum coverage problem and its variant. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 781–789, 2009.

[87] M Tapaswi. Book2Movie: Aligning video scenes with book chapters. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1827–1835, 2015.

[88] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. MovieQA: Understanding stories in movies through question-answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4631–4640, 2016.

[89] Cuneyt M. Taskiran, Zygmunt Pizlo, Arnon Amir, Dulce Ponceleon, and E.J. Edward J. Delp. Automated video program summarization using speech transcripts. *IEEE Transactions on Multimedia*, 8(4):775–790, 2006.

[90] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: A next-generation open source framework for deep learning. In *Advances in Neural Information Processing Systems (NIPS)*, 6 pages, 2015.

[91] Atousa Torabi, Niket Tandon, and Leonid Sigal. Learning language-visual embedding for movie understanding with natural-language. In *European Conference on Computer Vision (ECCV) Workshops*, 13 pages, 2016.

[92] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies (NAACL-HLT), pages 173–180, 2003.

[93] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015.

[94] Ba Tu Truong and Svetha Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 3(1), 19 pages, 2007.

[95] Sebastian Tschiatschek, Rishabh K. Iyer, Haochen Wei, and Jeff A. Bilmes. Learning mixtures of submodular functions for image collection summarization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1413–1421, 2014.

[96] Shingo Uchihashi, Jonathan Foote, Andreas Girgensohn, and John Boreczky. Video manga: Generating semantically meaningful video summaries. In *ACM International Conference on Multimedia (MM)*, pages 383–392, 1999.

[97] Kazuya Ueki, Kotaro Kikuchi, Susumu Saito, and Tetsunori Kobayashi. Waseda at TRECVID 2016: Ad-hoc video search. In *TRECVID Workshops*, 5 pages, 2016.

[98] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[99] Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence – video to text. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4534–4542, 2015.

[100] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language

using deep recurrent neural networks. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1494–1504, 2014.

[101] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015.

[102] Meng Wang, Richang Hong, Guangda Li, Zheng Jun Zha, Shuicheng Yan, and Tat Seng Chua. Event driven web video summarization by tag localization and key-shot identification. *IEEE Transactions on Multimedia*, 14(4):975–985, 2012.

[103] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *International Conference on Computer Vision (ICCV)*, pages 2794–2802, 2015.

[104] Zhe Wang, Limin Wang, Wenbin Du, and Yu Qiao. Exploring fisher vector and deep networks for action spotting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 10–14, 2015.

[105] Bo Xiong, Gunhee Kim, and Leonid Sigal. Storyline representation of egocentric videos with an applications to story-based search. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4525–4533, 2015.

[106] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, 2016.

[107] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML)*, pages 2048–2057, 2015.

[108] R Xu, C Xiong, W Chen, and JJ Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In

Association for the Advancement of Artificial Intelligence (AAAI), pages 2346–2352, 2015.

[109] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1155–1162, 2013.

[110] Huan Yang, Baoyuan Wang, Stephen Lin, David Wipf, Minyi Guo, and Baining Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4633–4641, 2015.

[111] Li Yao, Nicolas Ballas, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4507 – 4515, 2015.

[112] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[113] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3165–3173, 2017.

[114] Kuo-Hao Zeng, Tseng-Hung Chen, Juan Carlos Niebles, and Min Sun. Title generation for user generated videos. In *European Conference on Computer Vision (ECCV)*, pages 609–625, 2016.

[115] Bin Zhao and Eric P Xing. Quasi real-time summarization for consumer videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2513–2520, 2014.

[116] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems (NIPS)*, pages 487–495, 2014.

[117] Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, 2015.

List of Publications

Journal

1. M. Otani, Y. Nakashima, T. Sato, and N. Yokoya, "Video summarization using textual descriptions for authoring video blogs," *Multimedia Tools and Applications*, vol. 76, no. 9, pp. 12097–12115, 2017. (Chapter 3)

International Conference

1. M. Otani, H. Hioki, "Video colorization based on optical flow and edge-oriented color propagation," In *SPIE Computational Imaging XII*, vol. 9020, 9 pages, 2014.

2. M. Otani, Y. Nakashima, T. Sato, and N. Yokoya, "Textual description-based video summarization for video blogs," In *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2015. (Chapter 3)

3. M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya, "Learning joint representations of videos and sentences with web image search," In *European Conference on Computer Vision (ECCV) Workshops*, vol. 9913, pp. 651–667, 2016. (Chapter 4)

4. M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya, "Video summarization using deep semantic features," In *Asian Conference on Computer Vision (ACCV)*, vol. 10115, pp. 361–377, 2016. (Chapter 4)

5. M. Otani, Y. Nakashima, E. Rahtu, and J. Heikkilä, "Video question answering to find a desired video segment," In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) Workshops*, 3 pages, 2017. (Chapter 5)

6. M. Otani, Y. Nakashima, E. Rahtu, and J. Heikkilä, “Fine-grained video retrieval for multi-clip video,” In *International Conference on Computer Vision (ICCV) Workshops*, 3 pages, 2017. (Chapter 5)

Domestic Conference

1. 大谷 まゆ，日置尋久, 動きベクトル推定とエッジを考慮した色伝播に基づく動画カラリゼーション, マルチメディア情報ハイディング・エンリッチメント研究会，vol. 112, no. 467, pp. 13-18, 2013.

2. 大谷 まゆ，中島 悠太，佐藤 智和，横矢 直和, テキストと映像の類似度を用いた映像要約, 電子情報通信学会技術研究報告, PRMU2014-95, vol. 114, no. 409, pp.99-104, 2015. (Chapter 3)

3. 大谷 まゆ，中島 悠太，佐藤 智和，横矢 直和, テキスト記述を用いてユーザ意図を反映する映像要約, 電気関係学会関西連合大会講演論文集, pp. 384-385, 2014. (Chapter 3)

4. 大谷 まゆ，中島 悠太，佐藤 智和，横矢 直和, ユーザ意図反映のためのテキストに基づく映像要約手法, 電気関係学会関西連合大会講演論文集, pp. 388-389, 2015. (Chapter 3)

5. 大谷 まゆ, 中島 悠太, 佐藤 智和, 横矢 直和, テキストの内容に沿った重要箇所抽出による映像要約, 画像の認識・理解シンポジウム (MIRU) Extended Abstracts, PS2-39, 2016. (Chapter 3)

6. 橋岡佳輝, 大谷まゆ, 中島悠太, 佐藤智和, 横矢直和, DNN を用いたカメラの6自由度相対運動推定, 情報処理学会研究報告コンピュータビジョンとイメージメディア (CVIM), pp. 1-8, 2017.

7. M. Otani, Y. Nakashima, E. Rahtu, J. Heikkiä, N. Yokoya, “Unsupervised Video Summarization using Deep Video Features,” 画像の認識・理解シンポジウム (MIRU) Extended Abstracts, PS3-35, 2017. (Chapter 4)

8. M. Otani, Y. Nakashima, E. Rahtu, J, Heikkilä, “Finding Video Parts with Natural Language,” 情報処理学会研究報告コンピュータビジョンとイメージメディア (CVIM), 7 pages, 2018. (Chapter 5)

Awards

1. 平成26年度 奈良先端科学技術大学院大学情報科学研究科 「創造力と国際競争力を育む情報科学教育コアプロジェクト型研究」 特別賞, A. Tejero de Pabros, A. Tuchin, M. Otani, and H. Takehara, “Multimedia Abnormal Detection for Elders via Action and Pulse Recognition”.

2. 平成26年度 電気関連学会関西連合大会 奨励賞, 大谷 まゆ, 中島 悠太, 佐藤 智和, 横矢 直和, テキスト記述を用いてユーザ意図を反映する映像要約.

3. 平成27年度 奈良先端科学技術大学院大学優秀学生奨学生

4. 平成29年度　コンピュータビジョンとイメージメディア研究会 奨励賞，M. Otani, Y. Nakashima, E. Rahtu, J, Heikkilä, “Finding Video Parts with Natural Language”