GROUNDING NATURAL LANGUAGE PHRASES IN IMAGES AND VIDEO

BY

BRYAN A. PLUMMER

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Associate Professor Svetlana Lazebnik, Chair
Associate Professor Julia Hockenmaier
Associate Professor Derek Hoiem
Dr. Matthew Brown, Google Research

ABSTRACT

Grounding language in images has shown it can help improve performance on many image-language tasks. To spur research on this topic, this dissertation introduces a new dataset which provides the ground truth annotations of the location of noun phrase chunks in image captions. I begin by introducing a constituent task termed phrase localization, where the goal is to localize an entity known to exist in an image when provided with a natural language query. To address this task, I introduce a model which learns a set of models, each of which capture a different concept which is useful in our task. These concepts can be predefined, such as attributes gleamed from the adjectives, as well as those which are automatically learned in a single-end-to-end neural network. I also address the more challenging detection style task, where the goal is to localize a phrase *and* determine if it is associated with an image. Multiple applications of the models presented in this work demonstrate their value beyond the phrase localization task.

ACKNOWLEDGMENTS

Over the years I have been privileged to have received a lot of support from many sources for which I am grateful. My advisor, Svetlana Lazebnik, has been instrumental in my growth over the course of my PhD. Several other people have acted as mentors or helped my development including Julia Hockenmaier, Derek Hoiem, Matthew Brown, Stephen Kelley, Himanshu Arora, M. Hadi Kiapour, Shuai (Kyle) Zheng, Robinson Piramuthu, David Forsyth, and many others. I appreciate everyone's efforts in helping me become a better researcher and working with me over the years.

My friends, lab mates, and family have been essential to keeping me sane by offering me another view on a problem or a bit of respite from work: Aditya, Alice, Amanda, Arun, Cara, Charlotte, Chris, Chuhang, Daniel, Daphne, Erin, Fay, Felix, Jason, Jiajun, Joe, Kevin, Kristen, Liwei, Liz, Malia, Mariya, Marilyn, Nikoli, Paige, Pooya, Raj, Robert, Saraubh, Tanmay, Tom, Will, Zhizhong. I would also like to thank the administrative staff in Siebel over the years, who have always made me feel like they were happy to help me however they can.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

From robotics to human-computer interaction, there are numerous real-world tasks that would benefit from practical, large-scale systems that can identify objects in scenes based on language and understand language based on visual context. There has been a recent surge of work in this area, and in particular, on the task of sentence-based image description [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] and visual question answering [15, 16, 17, 18, 19, 20]. Unfortunately, due to a lack of datasets that provide not only paired sentences and images, but detailed *grounding* of specific phrases in image regions, most of these methods attempt to directly learn mappings from whole images to whole sentences. Not surprisingly, such models have a tendency to reproduce generic captions from the training data, and to perform poorly on *compositionally novel* images whose objects may have been seen individually at training time, but not in that combination [21]. Some works do try to find correspondences between image regions and parts of sentences [3, 22, 6, 23], but they treat such correspondences as latent and do not evaluate their quality directly. We argue that grounding of language to image regions is a problem that is hard and fundamental enough to require more extensive ground-truth annotations and standalone benchmarks.

After reviewing some related work in Chapter 2, we introduce the Flickr30K Entities dataset in Chapter 3. Our dataset augments the 158k captions from Flickr30k with 244k coreference chains, which links mentions of the same entity across different captions for the same image, and associates them with 276k manually annotated bounding boxes. These annotations enable us to define a new benchmark for localization of textual entity mentions in an image (see Figure 1.1 for an example of our task). We present a strong baseline for this task that combines an image-text embedding, detectors for common objects, a color classifier, and a bias towards selecting larger objects. Our experiments show that this approach rivals the accuracy of more complex models on the same task. We also provide a simple method to use our phrase localization approach to improve results on the task of bidirectional image-sentence retrieval.

In Chapter 4 we introduce a framework for localization or grounding of phrases in images which greatly extends the collection of linguistic and visual cues used previously. We model the appearance, size, and position of entity bounding boxes, adjectives that contain attribute information, and spatial relationships between pairs of entities connected by verbs or prepositions. Special attention is given to relationships between people and clothing or body part mentions, as they are useful for distinguishing individuals. We automatically learn weights for combining these cues and at test time, perform joint inference over all phrases in

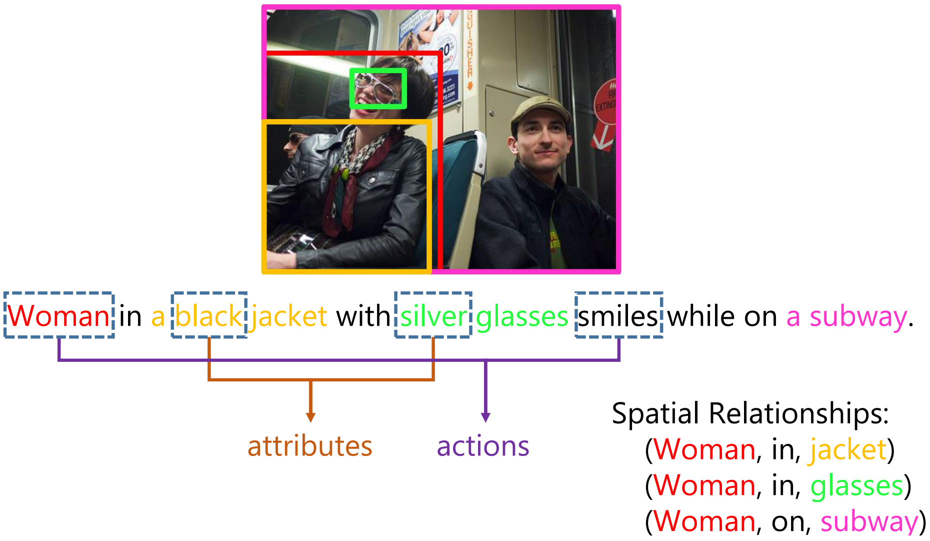


Figure 1.1: In the example above the image and sentence are provided as input into our phrase grounding model. Our task is to predict the location of each colored noun phrase within the sentence. We introduce a simple baseline which scores image regions based on a learned an embedding between vision and language features in Chapter 3. However, as shown above, the sentences themselves provide a wealth of cues such as attributes, actions, and spatial relationships which we use in a global inference approach to select the best image region for each phrase of the sentence in Chapter 4. We automatically learn a set of important concepts for phrase grounding rather than using predefined cues in Chapter 5. Then, we benchmark a more generalized version of this task in Chapter 6, where we have to decide if a phrase is associated with an image and then localize it.

a caption. The resulting system produces a 5% improvement in accuracy over other methods performing phrase localization on our dataset.

While the approach in Chapter 4 relies on a manually defined set of cues, Chapter 5 presents an approach for grounding phrases in images which jointly learns multiple text-conditioned embeddings in a single end-to-end model. This way our model learns what concepts may be important, which is especially important when cues from a predefined list are unavailable. For example, a short description of an object such as those that occur on the ReferIt Game dataset [24] is unlikely to have the verbs commonly associated between pair of objects. The data may also favor a different set of concepts than what has been defined (*e.g.* learning specialized models to identify the differences between "a person" and "people" may have a more significant impact on performance than identifying colors on some datasets). Our proposed solution simplifies the representation requirements for individual embeddings and allows the underrepresented concepts to take advantage of the shared representations before feeding them into concept-specific layers.

In Chapter 6 we propose to generalize our phrase localization models to not only determine where in an image a phrase is located, as done in Chapters 3 and 4, but to also indicate if it exists in an image or not. A significant challenge in undertaking this task is the high false positive rate for related phrases. For example, *a man* can be easily mistaken for *a woman* even though they refer to different things. To address this, the task is broken up into two specialized components: identification and localization. The phrase identification module determines if a query exists in an image while the localization module determines its location in the image. Using this approach we see significant gains, even when comparing to standard object detection methods for queries for which there is ample training data.

The models used in Chapter 6 are extended to tasks using videos in Chapter 7. First, we address the task of video summarization, or the problem of distilling a raw video into a shorter form while still capturing the original story. We show that visual representations supervised by freeform language make a good fit for this application by extending a recent submodular summarization approach [25] with representativeness and interestingness objectives computed on features from a joint vision-language embedding space. Then, we extend our phrase localization model to retrieve video segments from a given video which relate to a natural language input. Our work introduces new ways of addressing these problems and demonstrates how the models discussed in this work can generalize to new tasks.

The main contributions of this dissertation are summarized below:

- A new dataset which augments the Flickr30K dataset with region level annotations which are linked to noun phrases in image level captions (Chapter 3). This work appeared in ICCV 2015 [26] with an extended version in IJCV 2017 [27].

- An approach to phrase localization which combines an extensive set of predefined concepts to determine the location of an natural language query in an image (Chapter 4). This work appeared with experiments on Visual Relationship Detection [28], whose results are not discussed in this dissertation, in ICCV 2017 [29].

- A model for automatically learning important concepts for phrase localization in a single end-to-end model (Chapter 5).

- A benchmark on the more challenging detection style task where we must determine if a phrase is associated with an image as well as localize it (Chapter 6).

- Additional applications of the models presented in this work to tasks using videos rather than images as input (Chapter 7), some of which appeared in CVPR 2017 [30].

CHAPTER 2: RELATED WORK

There has been a long history of research on learning how to get computers to relate visual information with language. Early work on this topic included learning to relate words with pictures for automatic annotation [31, 32, 33] and using association between words and images to learn classifiers for visual concepts [34, 35, 36]. By leveraging relationships between words, works like Kulkarni *et al.* [9] and Farhadi *et al.* [4] proposed methods to generate complete sentences for an image. Sadeghi *et al.* [37] is an early precursor of the work addressed in this dissertation, where the authors localized a predefined objects and their relationships with associated textual input.

A typical approach to learning to relate images and text is to train an embedding model which projects the image and text featuers into a shared semantic space (*e.g.* [8, 38, 39, 40, 41]). Canonical correlation analysis (CCA) [42] is a popular method for learning these projections with extensions such as kernel CCA [39], deep CCA [40, 43], and normalized CCA [38], which we use in this work. Other approaches to learning an embedding between image and text features have included using autoencoders [44, 45], restricted Boltzmann machines [46], or to learn projections using shallow neural networks (*e.g.* [41, 47]).

2.1 OBJECT DETECTION

Phrase grounding can be viewed as simply performing object detection with a large set of classifiers. In recent years methods like the deformable part model [48] have been supplanted by neural network based models like RCNN [49]. This has been improved upon by focusing on making them faster [50, 51, 52] or improving candidate locations using a region proposal network [53], amongst many others. Some of these components, like the ROI pooling layer in Girshick *et al.* [50], is used by the models in this dissertation. Even though there are methods which can account for large numbers of object categories (*e.g.* [54, 55]), it is unfeasible to expect them to detect every object or attribute that may be important in discriminating between phrases. Since the distribution of words in language has a very long tail, generalizing to new objects is important for good grounding models.

In zero-shot object detection the goal is to train a model which can generalize to unseen object categories. In doing so, some of the approaches to this task are similar to those used by grounding models such as learning an embedding between the object's labels and the images (*e.g.* [56, 57, 58]) and using attributes to identify objects which generalize across categories (*e.g.* [59, 60]). A key difference between phrase grounding and zero-shot detection approaches is the need to performance instance level recognition. It is not enough to simply be able to identify an object (*e.g. a vase*), but may have to take into account relationships to other objects (*e.g. the vase on the table*) or other attributes which distinguish it from other objects of the same category (*e.g. a large blue vase*). Thus, a good grounding system needs to be able to perform instance recognition in addition to generalizing to unseen categories.

2.2 DATASETS WITH REGION-LEVEL DESCRIPTIONS

In this dissertation we would like to study phrase grounding, especially for semantically important entities. Thus, of particular import to this dissertation are datasets which not only contain global text descriptions and also include some kind of region-level annotations. An early example of this is the UIUC Sentences dataset [61], which consists of 1,000 images from PASCAL VOC 2008 [62] and five sentences per image. It inherits from PASCAL object annotations for 20 categories, but lacks explicit links between its captions and the object annotations. The most recent and large-scale dataset of this kind is Microsoft Common Objects in Context (MSCOCO) [63], containing over 300k images with five sentences per image and over 2.5m labeled object instances from 91 pre-defined categories. However, just as in UIUC Sentences, the MSCOCO region-level annotations are not linked to the captions in any way. Since none of the existing datasets provided the annotations we desired, we collected our own dataset (discussed in Chapter 3) which connects parts of the global text description to their location in the image.

Rather than pairing images with a caption that summarizes the entire image, some datasets pair specific objects in an image with short descriptions. The ReferIt dataset [24] focuses on *referring expressions* that are necessary to uniquely identify an object instance in an image. It augments the IAPR-TC dataset [64] of 20k photographs with 130k isolated entity descriptions for 97k objects from 238 categories. The Google Refexp dataset [65] is built on top of MSCOCO and contains a little under 27k images with 105k descriptions, and it uses a methodology that produces longer descriptions than ReferIt. Visual MadLibs [20] is a subset of 10,738 MSCOCO images with several types of focused fill-in-the-blank descriptions (360k in total), some referring to attributes and actions of specific people and object instances, and some referring to the image as a whole.

Johnson *et al.* [66] is another notable work concerned with grounding of semantic scene descriptions to image regions. Instead of natural language, it proposes a formal *scene graph* representation that encapsulates all entities, attributes and relations in an image, together with a dataset of scene graphs and their groundings for 5k images. The more recent Visual Genome dataset [17] follows the same methodology, but contains 108k images rather than

5k and a denser set of annotations. Each image in Visual Genome has an average of 21 objects, 18 attributes, and 18 pairwise relations. Due to the nature of the Visual Genome crowdsourcing protocol, its object annotations have a greater amount of redundancy than our dataset. For example, the phrases *a boy wearing jeans* and *this is a little boy* may be totally separate and come with separate bounding boxes despite referring to the same person in the image. In addition, for phrases referring to multiple objects like *three people*, Visual Genome would only have one box drawn around all three people, while we asked for individual boxes for each person, linking all three boxes to the phrase. While the Visual Genome is the largest source of unstructured localized textual expressions to date, our dataset is better suited for understanding the different ways people refer to the same visual entities within an image, and which entities are salient for the purpose of natural language description.

Finally, there exist a few specialized datasets with extensive annotations, but more limited domains of applicability than Flickr30k Entities or Visual Genome. Kong *et al.* [67] have taken the 1,449 RGB-D images of static indoor scenes from the NYUv2 dataset [68] and obtained detailed multi-sentence descriptions focusing mainly on spatial relationships between objects. Similar to Flickr30k Entities, this dataset contains links between different mentions of the same object, and between words in the description and the respective location in the image. [69] have introduced the Abstract Scene dataset, which contains 10,020 synthetic images created using clip art objects from 58 categories, together with captions and ground-truth information of how objects relate to the captions.

2.3 GROUNDED LANGUAGE UNDERSTANDING

A common image-language understanding task in the literature is automatic image captioning [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 70, 71]. Of most importance to us are the methods attempting to associate local regions in an image with words or phrases in the captions, as they would likely benefit the most from our annotations.

Many works leveraging region-phrase correspondences rely on weakly supervised learning due to a lack of ground-truth correspondences at training time. One popular approach is to use multiple instance learning to train detectors for words that commonly occur in captions, and then feed the outputs of these detectors into a language model to generate novel captions (*e.g.* [3, 72, 73]). Xu *et al.* [23] incorporates a soft form of attention into their recurrent model, which is trained to fixate on a sequence of latent image regions while generating words. [22, 6] propose an image-sentence ranking approach in which the score between an image and sentence is defined as the average over correspondence scores between each sentence fragment and the best corresponding image region; at training time, the correspondences are treated as latent and incorporated into a structured objective. Ma *et al.* [70] learns multiple networks capturing word, phrase, and sentence-level interactions with an image and combine the scores of these networks to obtain a whole image-sentence score. Since there is no explicit mapping between phrases and the image, all three networks use the whole image representation as input. However, Liu *et al.* [74] took advantage of the annotations in our dataset to provide a supervised version of attention to improve image captioning.

Visual Question Answering (VQA) [15, 16, 19, 20, 75] has also received a lot of attention in recent years. Since many questions are directed about particular components of the images, attention-based methods have been given a lot of consideration [75, 76, 77, 78]. Others have trained models to capture a specific desirable trait [79, 80, 81]. Indeed, since phrase localization and VQA require their models to recognize entities and their attributes there are approaches can be adapted to work on both tasks (*e.g.* [80, 82]).

2.3.1 Phrase Grounding

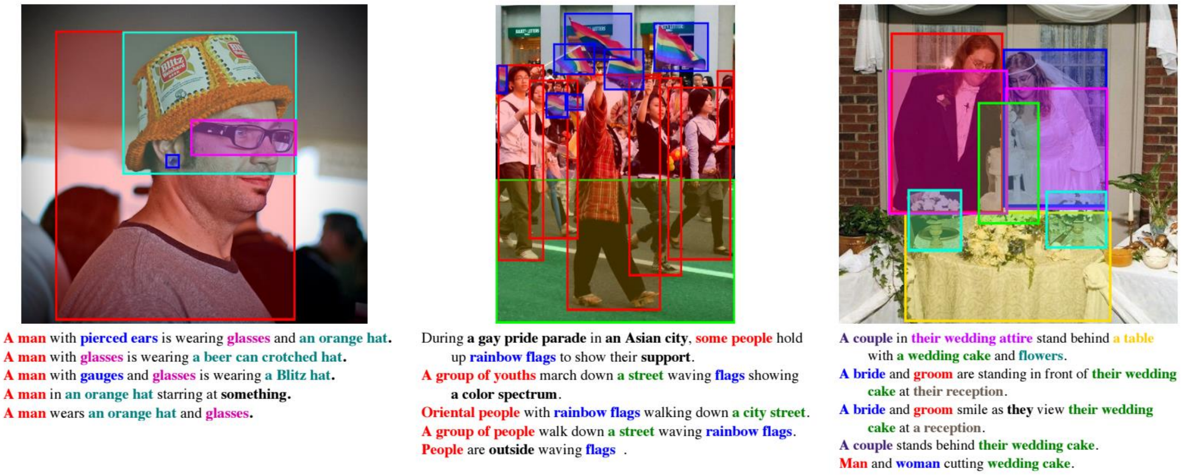
In this work we consider the task of grounding or localizing textual mentions of entities in an image. Until recently, it was rare to see direct evaluation on this task due to a lack of ground-truth annotations (with the notable exception of Kong *et al.* [67] and their dataset of RGB-D room descriptions). Rohrbach *et al.* [83] were among the first to use Flickr30K Entities for phrase localization by training an LSTM model to attend to the right image region in order to reproduce a given phrase. Their work shows that fully supervised training of this model with ground-truth region-phrase correspondences results in much better performance than weakly supervised training, thus confirming the usefulness of our annotations. Since then, a number of other works have adopted Flickr30K Entities as well. Zhang *et al.* [84] produces a linear classifier used to discriminate between image regions based on the textual input. Wang *et al.* [85] learns a nonlinear region-phrase embedding that can localize phrases more accurately than our simple linear embedding learned using CCA in Section 3.2. Fukui *et al.* [82] uses compact bilinear pooling to learn a detailed relationship between the image and text features. Zhang *et al.* [86] performs phrase localization with a tag prediction network and a top-down attention model. As a more open-ended alternative to phrase localization, Johnson *et al.* [87] introduce dense image captioning, or the task of predicting image regions and generating freeform descriptions for them.

Some researchers have focused on leveraging the structure of the data and other informative cues to improve their models. Hu *et al.* [88] leverage spatial information and global context to model where objects are likely to occur. Liu *et al.* [89] learned a set of attributes to help identify entities in an image. Wang *et al.* [90] formulate a linear program to localize all the phrases from a caption jointly, taking their semantic relationships into account. This idea of considering the predictions made by other phrases in a sentence was also used in a pair of works by Chen *et al*. [91, 92] with methods that were otherwise analogous to the Fast/Faster RCNN models with global inference models. Yu *et al*. [93] took into account the visual similarity of objects in a single image when providing context for their predictions. Yeh *et al*. [94] used a word prior in combination with segmentation masks, geometric features, and detection scores to select a region from all possible bounding boxes in an image.

CHAPTER 3: FLICKR30K ENTITIES – COLLECTING REGION-TO-PHRASE CORRESPONDENCES FOR RICHER IMAGE-TO-SENTENCE MODELS

This chapter discusses a major contribution of this work, a large-scale comprehensive dataset of region-to-phrase correspondences for image description. We build on the Flickr30k dataset [95], a popular benchmark for caption generation and retrieval that has been used, among others, by [1, 2, 3, 38, 22, 6, 7, 8, 10, 11, 13, 23]. Flickr30k contains 31,783 images focusing mainly on people and animals, and 158,915 English captions (five per image). Our new dataset, Flickr30k Entities, augments Flickr30k by identifying which mentions among the captions of the same image refer to the same set of entities, resulting in 244,035 *coreference chains*, and which image regions depict the mentioned entities, resulting in 275,775 bounding boxes. Figure 3.1 illustrates the structure of our annotations on three sample images and Table 3.1 offers a comparison against similar datasets which were discussed in Chapter 2. Section 3.1 describes our crowdsourcing protocol, which consists of two major stages – coreference resolution and bounding box drawing – and each stage in turn is split up into smaller atomic tasks to ensure both efficiency and quality.

Together with our annotations, we propose a new benchmark task of *phrase localization*, which we view as a fundamental building block and prerequisite for more advanced image-language understanding tasks. Given an image and a caption that accurately describes it, the goal of phrase localization is to predict a bounding box for a specific entity mention from that sentence. This task is akin to object detection and can in principle be evaluated in an analogous way, but it has its own unique challenges. Traditional object detection assumes a predefined list of semantically distinct classes with many training examples for each. By contrast, in phrase localization, the number of possible phrases is very large, and many of them have just a single example or are completely unseen at training time. Also, different phrases may be very semantically similar (e.g., *infant* and *baby*), which makes it difficult to train separate models for each. And of course, to deal with the full complexity of this task, we need to take into account the broader context of the whole image and sentence, for example, when disambiguating between multiple entities of the same type. In Section 3.2, we propose a strong baseline for this task based on a combination of image-text embeddings, pre-trained detectors, and size and color cues. While this baseline outperforms more complex recent methods (e.g., [83]), it is not yet strong enough to discriminate between multiple competing interpretations that roughly fit an image, which is necessary to achieve improvements over state-of-the-art global methods for image description.



A man with **pierced ears** is wearing **glasses** and **an orange hat**.
A man with **glasses** is wearing **a beer can crotched hat**.
A man with **gauges** and **glasses** is wearing **a Blitz hat**.
A man in **an orange hat** starring at **something**.
A man wears **an orange hat** and **glasses**.

During **a gay pride parade** in **an Asian city**, **some people** hold up **rainbow flags** to show their **support**.
A group of youths march down **a street** waving **flags** showing **a color spectrum**.
Oriental people with **rainbow flags** walking down **a city street**.
A group of people walk down **a street** waving **rainbow flags**.
People are **outside** waving **flags** .

A couple in **their wedding attire** stand behind **a table** with **a wedding cake** and **flowers**.
A bride and **groom** are standing in front of **their wedding cake** at **their reception**.
A bride and **groom** smile as **they** view **their wedding cake** at **a reception**.
A couple stands behind **their wedding cake**.
Man and **woman** cutting **wedding cake**.

Figure 3.1: Example annotations from our dataset. In each group of captions describing the same image, coreferent mentions (*coreference chains*) and their corresponding bounding boxes are marked with the same color. On the left, each chain points to a single entity (bounding box). Scenes and events like "outside" or "parade" have no box. In the middle example, the people (red) and flags (blue) chains point to multiple boxes each. On the right, blue phrases refer to the bride, and red phrases refer to the groom. The dark purple phrases ("a couple") refer to both of these entities, and their corresponding bounding boxes are identical to the red and blue ones.

3.1 ANNOTATION PROCESS

In this section, we describe the crowdsourcing protocol we adopted for collecting Flickr30k Entities. Our annotations, illustrated in Figure 3.1, consist of cross-caption coreference chains linking mentions of the same entities together with bounding boxes localizing those entities in the image. These annotations are highly structured and vary in complexity from image to image, since images vary in the numbers of clearly distinguishable entities they contain, and sentences vary in the extent of their detail. Further, there are ambiguities involved in identifying whether two mentions refer to the same entity or set of entities, how many boxes (if any) these entities require, and whether these boxes are of sufficiently high quality. Due to this intrinsic subtlety of our task, compounded by the unreliability of crowdsourced judgments, we developed a pipeline of simpler atomic tasks, screenshots of which are shown in Figure 3.2. These tasks can be grouped into two main stages: **coreference resolution**, or forming coreference chains that refer to the same entities (Section 3.1.1, whose annotations were collected by a co-author of the dataset, Juan C. Caicedo), and **bounding box annotation** for the resulting chains (Section 3.1.2, whose annotations I collected). This workflow provides two advantages: first, identifying coreferent mentions helps reduce redundancy and save box-drawing effort; and second, coreference annotation is intrinsically valuable, *e.g.*, for

	Dataset	Images	Objects Per Image	Object Categories	Objects Per Category	Sentences Per Image	Expressions Per Image
Image-Sentence Datasets	**Flickr30k Entities**	31,783	8.7	44,518	6.2	5	16.6
	MSCOCO [63]	328,000	7.7	91	27,473	5	–
Image-Phrase Datasets	ReferIt [24]	19,894	4.9	238	406.1	–	6.6
	Google Refexp [65]	26,711	2.1	80	685.3	–	3.9
	Scene Graph [66]	5,000	18.8	6,745	13.9	–	33.0
	Visual Genome* [17]	108,077	$\sim$56	110,689	$\sim$54	–	$\sim$40

*obtained via personal communication with the authors

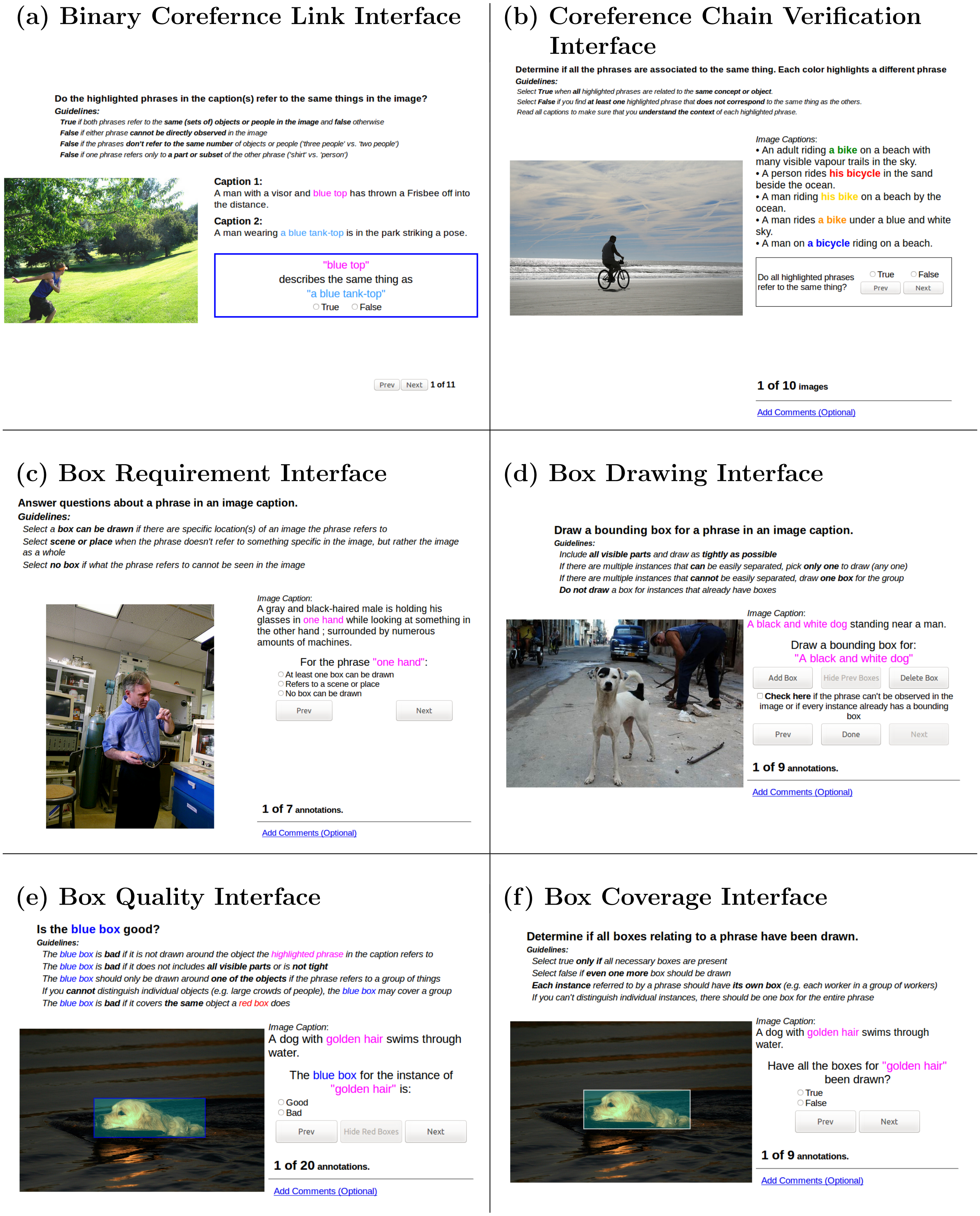
Table 3.1: Comparison of dataset statistics. For our dataset, we define Object Categories as the set of unique phrases after filtering out non-nouns in our annotated phrases (note that Scene Graph and Visual Genome also have very large numbers in this column because they correspond essentially to the total numbers of unique phrases). For Expressions Per Image, we list for our dataset the average number of entity mentions in all five sentences.

training cross-caption coreference models [96]. Section 3.1.3 will discuss issues connected to data quality, and Section 3.1.4 will give a brief analysis of dataset statistics.

3.1.1 Coreference Resolution

We rely on the chunking information given in the Flickr30k captions [95] to identify potential entity mentions. With the exception of personal pronouns (*he, she, they*) and a small list of frequent non-visual terms (*background*, *air*), we assume that any noun-phrase (NP) chunk is a potential entity mention. NP chunks are short (avg. 2.35 words), non-recursive phrases (e.g., the complex NP *[[a man] in [an orange hat]]* is split into two chunks). Mentions may refer to single entities (*a dog*); regions of "stuff" (*grass*); multiple distinct entities (*two men*, *flags*, *football players*); groups of entities that may not be easily identified as individuals (*a crowd*, *a pile of oranges*); or even the entire scene (*the park*). Finally, some NP chunks may not refer to any physical entities (*wedding reception*, *a trick*, *fun*).

Once we have our candidate mentions from the sentences corresponding to the same image, we need to identify which ones refer to the same set of entities. Since each caption is a single, relatively short sentence, pronouns (*he*, *she*, *they*) are relatively rare in this dataset. Therefore, unlike in standard coreference resolution in running text [97], which can be beneficial for identifying all mentions of people in movie scripts [98], we ignore anaphoric references between pronouns and their antecedents and focus on cross-caption coreference resolution [96]. Like standard coreference resolution, our task partitions the set of mentions M in a document (here, the five captions of one image), into subsets of equivalent mentions such that all mentions in the same subset $c \in C$ refer to the same set of entities. In keeping with standard terminology, we refer to each such set or cluster of mentions $c \subset M$ as a coreference chain.



(a) Binary Corefernce Link Interface

Do the highlighted phrases in the caption(s) refer to the same things in the image?
Guidelines:
True *if both phrases refer to the* ***same (sets of) objects or people in the image*** *and* ***false*** *otherwise*
False *if either phrase* ***cannot be directly observed*** *in the image*
False *if the phrases* ***don't refer to the same number*** *of objects or people ('three people' vs. 'two people')*
False *if one phrase refers only to* ***a part or subset*** *of the other phrase ('shirt' vs. 'person')*

Caption 1:
A man with a visor and blue top has thrown a Frisbee off into the distance.

Caption 2:
A man wearing a blue tank-top is in the park striking a pose.

"blue top"
describes the same thing as
"a blue tank-top"
○ True ○ False

(b) Coreference Chain Verification Interface

Determine if all the phrases are associated to the same thing. Each color highlights a different phrase
Guidelines:
Select ***True*** *when* ***all*** *highlighted phrases are related to the* ***same concept or object****.*
Select ***False*** *if you find* ***at least one*** *highlighted phrase that* ***does not correspond*** *to the same thing as the others.*
Read all captions to make sure that you ***understand the context*** *of each highlighted phrase.*

Image Captions:
- An adult riding **a bike** on a beach with many visible vapour trails in the sky.
- A person rides **his bicycle** in the sand beside the ocean.
- A man riding **his bike** on a beach by the ocean.
- A man rides **a bike** under a blue and white sky.
- A man on **a bicycle** riding on a beach.

Do all highlighted phrases refer to the same thing? ○ True ○ False

(c) Box Requirement Interface

Answer questions about a phrase in an image caption.
Guidelines:
Select a ***box can be drawn*** *if there are specific location(s) of an image the phrase refers to*
Select ***scene or place*** *when the phrase doesn't refer to something specific in the image, but rather the image as a whole*
Select ***no box*** *if what the phrase refers to cannot be seen in the image*

Image Caption:
A gray and black-haired male is holding his glasses in one hand while looking at something in the other hand ; surrounded by numerous amounts of machines.

For the phrase "one hand":
○ At least one box can be drawn
○ Refers to a scene or place
○ No box can be drawn

(d) Box Drawing Interface

Draw a bounding box for a phrase in an image caption.
Guidelines:
Include ***all visible parts*** *and draw as* ***tightly as possible***
If there are multiple instances that ***can*** *be easily separated, pick* ***only one*** *to draw (any one)*
If there are multiple instances that ***cannot*** *be easily separated, draw* ***one box*** *for the group*
Do not draw *a box for instances that already have boxes*

Image Caption:
A black and white dog standing near a man.

Draw a bounding box for:
"A black and white dog"

☐ **Check here** if the phrase can't be observed in the image or if every instance already has a bounding box

(e) Box Quality Interface

Is the blue box good?
Guidelines:
The blue box is ***bad*** *if it is not drawn around the object the highlighted phrase in the caption refers to*
The blue box is ***bad*** *if it does not includes* ***all visible parts*** *or is* ***not tight***
The blue box should only be drawn around ***one of the objects*** *if the phrase refers to a group of things*
If you ***cannot*** *distinguish individual objects (e.g. large crowds of people), the blue box may cover a group*
The blue box is ***bad*** *if it covers* ***the same*** *object a red box does*

Image Caption:
A dog with golden hair swims through water.

The blue box for the instance of "golden hair" is:
○ Good
○ Bad

(f) Box Coverage Interface

Determine if all boxes relating to a phrase have been drawn.
Guidelines:
Select true ***only if*** *all necessary boxes are present*
Select false if ***even one more*** *box should be drawn*
Each instance *referred to by a phrase should have* ***its own box*** *(e.g. each worker in a group of workers)*
If you can't distinguish individual instances, there should be one box for the entire phrase

Image Caption:
A dog with golden hair swims through water.

Have all the boxes for "golden hair" been drawn?
○ True
○ False

Figure 3.2: Examples of the interfaces used in our annotation pipeline described in Section 3.1.

Binary Coreference Link Annotation

Since the task of constructing an entire coreference chain from scratch is cognitively complex and error-prone, we broke it down into simpler tasks collecting binary coreference links between pairs of mentions. A coreference link between mentions m and m' indicates that m and m' refer to the same set of entities. In the manual annotation process, workers are shown an image and the two captions from which m and m' originate. The workers are asked whether these mentions refer to the same entity. See Figure 3.2(a) for a screenshot of the interface for this task. If a worker indicates that the mentions are coreferent, we add a link between m and m'. Given a set of mentions M for an images, manual annotation of all $O(|M|^2)$ pairwise links is very costly. But since M typically contains multiple mentions that refer to the same set of entities, the number of coreference chains is bounded by, and typically much smaller than, $|M|$. This allows us to reduce the number of links that need to be annotated to $O(|M||C|)$ by leveraging the transitivity of the coreference relation [99]. Given a set of identified coreference chains C and a new mention m that has not been annotated for coreference yet, we only have to ask for links between m and one mention from each element of C. If m is not coreferent with any of these mentions, it refers to a new entity whose coreference chain is initialized and added to C.

In the worst case, each entity has only one mention requiring annotation of all $|M|^2$ possible links. But in practice, most images have more mentions than coreference chains (in our final dataset, each image has 16.6 mentions and 7.8 coreference chains on average). We further reduce the number of required annotations with two simplifying assumptions. First, we assume that mentions from the same captions cannot be coreferent, as it would be unlikely for a caption to contain two non-pronominal mentions to the same set of entities. Second, we categorize each mention into eight coarse-grained types using manually constructed dictionaries (people, body parts, animals, clothing/color,[1] instruments, vehicles, scene, and other), and assume mentions belonging to different categories cannot be coreferent.

To ensure that our greedy strategy leveraging the transitivity relations would not have a significant impact on data quality, we conducted a small-scale experiment using 200 images. First, we asked workers to annotate each of the $O(|M|^2)$ pairwise links several times to obtain a set of gold (ground-truth) coreference chains. Then we collected the links again using both the exhaustive and greedy strategies and compared them to the gold links. In addition, after collecting the links, we looked for any violations of transitivity between phrases and asked additional workers to annotate the links involved until we got a consensus. We call the

[1] In Flickr30k, NP chunks that only consist of a color term are often used to refer to clothing, e.g. *man in blue*.

	Exhaustive	Greedy	Exhaustive Plus	Greedy Plus
Matched Gold Links	95.03%	94.46%	98.33%	96.40%
Matched Gold Coref. Chain	81.84%	81.55%	90.67%	86.07%
False Positive Rate	4.92%	5.74%	1.04%	3.14%
False Negative Rate	4.70%	4.99%	2.28%	4.05%
Links To Annotate	100.00%	57.00%	119.74%	66.94%

Table 3.2: Comparison of different annotation strategies for collecting binary coreference links on 200 images. We report the false positive/negative rates for the individual binary link judgments, as well as how many of the coreference chains created by the different strategies matched the gold coreference chains.

resulting strategies "exhaustive plus" and "greedy plus." As seen in the Table 3.2, the greedy and exhaustive strategies perform quite similarly, "greedy plus" actually performs better than exhaustive while requiring more than 30% fewer links, and "exhaustive plus" achieves the highest accuracy on this task but at prohibitive cost. Based on these considerations, we decided to use "greedy plus" for the entire dataset, and Figure 3.3 shows the source of the links we obtained using this strategy.

Coreference Chain Verification

To handle errors introduced by the coreference link annotation, we verify the accuracy of all chains that contain more than a single mention. In this task, workers are shown the mentions that belong to the same coreference chain and asked whether all the mentions refer to the same set of entities. If the worker answers True, the chain is kept as-is. If a worker answers False, that chain is broken into subsets of mentions that share the same head noun (the last word in a chunk). An example of the interface for this task is shown in Figure 3.2(b). There were 123,758 coreference chains with more than a single mention to verify in this stage. Of them, 111,628 (90.2%) were marked as good by workers, with the remaining 12,130 (9.8%) marked as bad and broken up before moving on the next step of the annotation pipleline.

It is important to note that our coreference chain verification is not designed to spot false negatives, or missing coreference links. Although false negatives lead to fragmented entities and redundant boxes (and consequently higher time and cost for box drawing), we can recover from many of these errors in a later stage by merging bounding boxes that have significant overlap (Section 3.1.3). On the other hand, false positives (spurious coreference links) are more harmful, since they are likely to result in mentions being associated with incorrect entities or image regions.

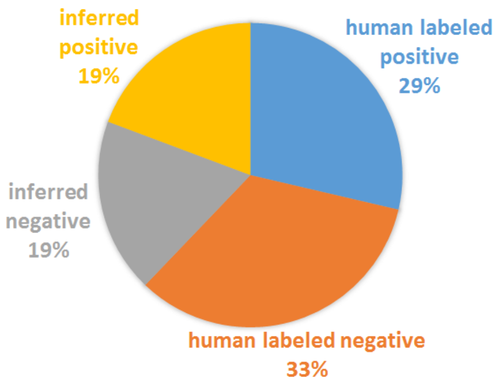


Figure 3.3: Distribution of the source of binary coreference link annotations on the entire dataset using the Greedy Plus strategy.

3.1.2 Bounding Box Annotations

The workflow to collect bounding box annotations is broken down similarly to [100], and consists of four separate AMT tasks, discussed below: (1) Box Requirement, (2) Box Drawing, (3) Box Quality, and (4) Box Coverage. In each task, workers are shown an image and a caption in which a representative mention for one coreference chain is highlighted. We use the longest mention in each chain, since we assume that it is the most specific.

Box Requirement

First, we determine if the entities a representative mention refers to require boxes to be drawn. A mention does not require boxes if it refers to the entire scene (*in [the park]*), to physical entities that are not in the image (*pose for [the camera]*), or to an action or abstract entity (*perform [a trick]*). As shown in the example interface in Figure 3.2(c), given an image and a caption with a highlighted mention, we ask workers whether (1) at least one box can be drawn (2) the mention refers to a scene or place or (3) no box can be drawn.

If the worker determines that at least one box can be drawn, the coreference chain proceeds to the Box Drawing task (below). Otherwise, we ask for a second and sometimes a third Box Requirement judgment to obtain agreement between two workers. If the majority agrees that no box needs to be drawn, the coreference chain is marked as “non-visual” and leaves the bounding box annotation workflow. After preliminary analysis, we determined that coreference chains with mentions from the people, clothing, and body parts categories so frequently required boxes that they immediately proceeded to the Box Drawing task, skipping the Box Requirement task altogether.

Box Drawing

In this task, we collect bounding boxes for a mention. The key source of difficulty here is due to mentions that refer to multiple entities. Our annotation instructions specify that we expect individual boxes around each entity if these can be clearly identified (e.g., *two people* would require two boxes). But if individual elements of a group cannot be distinguished (*a crowd of people*), a single box may be drawn around the group. We show workers all previously drawn boxes for the representative mention (if they exist), and ask them to draw one new box around one entity referred to by the mention, or to indicate that no further boxes are required (see Figure 3.2(d) for a screenshot).

If the worker adds a box, the mention-box pair proceeds to the Box Quality task. If the worker indicates that no boxes are required, the mention accrues a "no box needed" judgment. The mention is then returned to Box Requirement if it has no boxes associated with it. Otherwise, the mention is sent to Box Coverage.

Box Quality

For each newly drawn box, we ask a worker whether the box is good. Since we want to avoid redundant boxes, we also show all previously drawn boxes for the same mention. Good boxes are tightly drawn around the entire entity a mention refers to which no other box already covers. When mentions refer to multiple entities that can be clearly distinguished, these must be associated with individual boxes. If the worker marks the box as Bad, it is discarded and the mention is returned to the Box Drawing task. If the worker marks the box as Good, the mention proceeds to the Box Coverage task to determine whether additional boxes are necessary. See Figure 3.2(e) for an example interface for this task.

Box Coverage

In this step, workers are shown the boxes that have been drawn for a mention, and asked if all required boxes are present for that mention (Figure 3.2(f)). If the initial judgment says that more boxes are needed, the mention is immediately sent back to Box Drawing. Otherwise, we require a second worker to verify the decision that all boxes have been drawn. If the second worker disagrees, we collect a third judgment to break the tie, and either send the mention back to Box Drawing, or assume all boxes have been drawn.

3.1.3 Quality Control

Since worker quality on AMT is highly variable [101, 61], we take a combination of measures to ensure the integrity of annotations. First, we only allow workers who have completed at least 500 previous HITs with 95% accuracy, and have successfully completed a corresponding qualification test for each of our six tasks. After this basic filtering, it is still necessary to ensure that a worker continues to provide quality annotations. A common method for doing so is to insert verification questions (questions with known answers) in all the jobs. Initially, we included 20% verification questions in our jobs, which were evaluated on a per-worker basis in batches. While this process produced satisfactory results for the first three steps of the annotation pipeline (Binary Coreference Link Annotation, Coreference Chain Verification, and Box Requirement), we were not able to successfully apply this model to the last three steps having to do with box drawing. This appears to be due, in part, to the greater difficulty and attention to detail required in those steps. Not only does someone have to read and understand the sentence and how it relates to the image being annotated, but he or she must also be careful about the placement of the boxes being drawn. This increased difficulty led to a much smaller portion of workers successfully completing the tasks (see rejection rates in Table 3.3). Even our attempts to change the qualification task to be more stringent had little effect on worker performance. Sticking with a verification model for these challenging tasks would either lead to higher costs (if we were to pay workers for poorly completed tasks) or greatly reduced completion rates (due to workers not wanting to risk doing a task they may not get paid for).

Instead, we used a list of Trusted Workers to pre-filter who can do our tasks. To determine if a worker was to be placed on this list, those who passed our up-front screening were initially given jobs that only contained verification questions when they requested a job in our current batch. If they performed well on their first 30 items based on the thresholds in Table 3.3, they would qualify as a Trusted Worker and would be given our regular jobs with only 2% verification questions inserted. To remain on the Trusted Worker list, one simply had to maintain the same quality level in both overall and most recent set of responses to verification questions. The reduced number of verification questions limited the cost since poorly performing workers were identified quickly and more new items would be annotated for each job, which also increased the collection rate for our annotations.

	Anno/ Task	Time (s)	# Trusted Workers (TW)	TW Quality	Minimum Performance	% Rejected For Non-TW
Coreference Links	10	75	587	90.6%*	80%	2*
Coreference Verify	5	95	239	90.6%*	83%	2*
Box Requirement	10	81	684	88.4%	83%	< 1
Box Drawing	5	134	334	82.4%	70%	38.3
Box Quality	10	110	347	88.0%	78%	52.7
Box Coverage	10	91	624	89.2%	80%	35.4

*combined

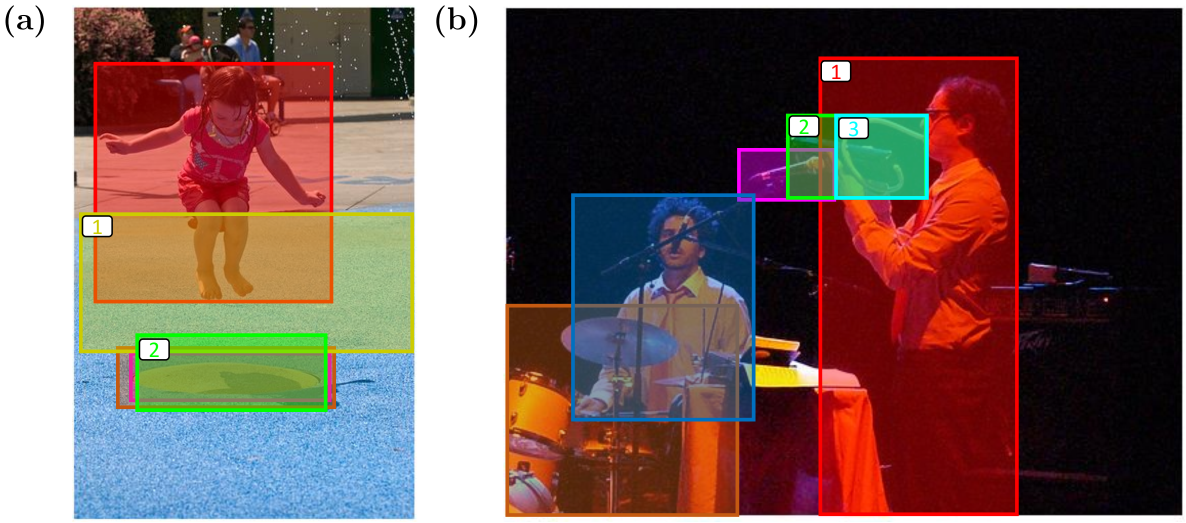
Table 3.3: Per-task crowdsourcing statistics for our annotation process. Trusted Worker Quality is the average accuracy of trusted workers on verification questions (or approved annotations in the Box Drawing task). Min Performance is the Worker Quality score a worker must maintain to remain approved to do our tasks. To give an idea of the general level of complexity of our different tasks, we also list % Rejected, which is the proportion of automatically rejected jobs (tasks) among *non-trusted* workers based on verification question performance. After we switched to a Trusted Worker model, we had virtually no rejected jobs.

Additional Review

At the end of the crowdsourcing process, we identified roughly 4k entities that required additional review. This included some chunking errors that came to our attention (e.g., through worker comments), as well as chains that cycled repeatedly through the Box Requirement or Box Coverage task, indicating disagreement among the workers. Images with the most serious errors were manually reviewed.

Box and Coreference Chain Merging

As discussed in Section 3.1.1, coreference chains may be fragmented due to missed links (false negative judgments). Additionally, if an image contains more than one entity of the same type, its coreference chains may overlap or intersect (e.g., *a bride* and *a couple* from Figure 3.1). Since Box Drawing operates over coreference chains, it results in redundant boxes for such cases. We remove this redundancy by merging boxes with IOU scores of at least 0.8 (or 0.9 for “other”). These thresholds were determined after an extensive manual review of the annotations. Some restrictions were placed on the types of phrases that were allowed to be combined (e.g. clothing and people boxes cannot be merged). Afterwards, we merge any coreference chains that point to the exact same set of boxes. This merging resulting in a reduction of the number of bounding boxes in the dataset by 19.8% and 5.9% fewer coreference chains.



A young girl playing is **a sprinkler fountain** jumps on a yellow concrete spot.

A young girl is jumping on a yellow dot in the middle of a blue play area.

Little girl jumping up to land on a yellow circle at a splash pad.

A young girl is jumping over a yellow circle on **the ground**.

A little girl jumps on a yellow circle in a field of blue.

a musician plays a strange pipe instrument whilst standing next to a drummer on **a stage**.

A man blows into a tube while standing in front of a man at the drumset on **stage**.

A man blows into an electrical instrument by a microphone.

A man plays an instrument next to a drummer.

Two men perform **a song** together on **stage**.

Figure 3.4: Examples of errors in Flickr30k Entities. In (a), the second caption contains an error due to complex constructions. Here, the proper chunking should be *[the middle] of [a blue play area]*, where the *blue play area* is the entire blue region, and *the middle* refers to the area containing the yellow dot. As it is, the coreference link *the middle of a blue play area* and *a field of blue* is not valid and there is an ambiguity as to whether the corresponding tan box (labeled 1) should cover just the yellow area or the entire blue area (either way, the box is incorrect). Furthermore, the entity mentions *a yellow dot*, *a yellow circle*, *a splash pad*, and *a yellow concrete spot* is fragmented into three chains with three distinct bounding boxes (labeled 2). In (b) the coreferent entity mentions *a strange pipe*, *a tube*, *an electrical instrument*, and *an instrument* are fragmented into three chains. The phrase *an instrument* in the fourth sentence is linked to both boxes 1 and 2, when it should be linked to box 2 alone. Box 3 for *a tube* is also too small, so it couldn't be merged with box 2.

Error Analysis

Errors present in our dataset mostly fall under two categories: chunking and coreference errors. Chunking errors occur when the automated tools made a mistake when identifying mentions in caption text. Coreference errors occur when AMT workers made a bad judgment when building coreference chains. An analysis using a combination of automated tools and manual methods identified chunking errors in less than 1% of the dataset's mentions and coreference errors in less than 1% of the datasets chains. Since, on average, there are over 16 mentions and 7 chains per image, there is an error of some kind in around 8% of our images. Figure 3.4 shows examples of some of the errors found in our dataset.

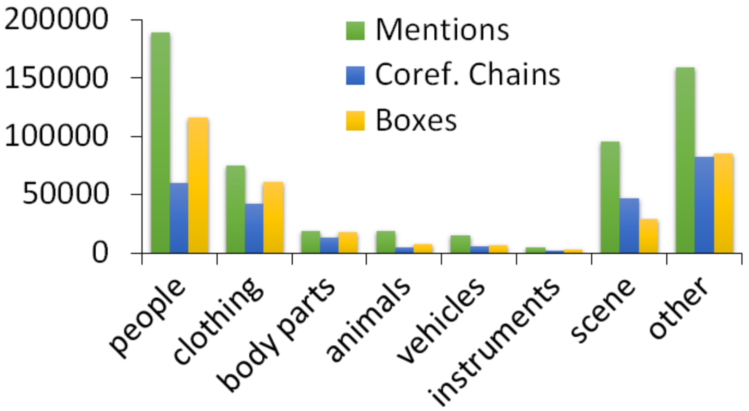


Figure 3.5: The total number of coreference chains, mentions, and bounding boxes per type.

Type	#Chains	Mentions/Chain	Boxes/Chain
people	59,766	3.17	1.95
clothing	42,380	1.76	1.44
body parts	12,809	1.50	1.42
animals	5,086	3.63	1.44
vehicles	5,561	2.77	1.21
instruments	1,827	2.85	1.61
scene	46,919	2.03	0.62
other	82,098	1.94	1.04
total	244,035	2.10	1.13

Table 3.4: Coreference chain statistics. The number of mentions per chain indicates how salient an entity is. The number of boxes per chain indicates how many distinct entities it refers to.

3.1.4 Dataset Statistics

Our annotation process has identified 513,644 entity or scene mentions in the 158,915 Flickr30k captions (3.2 per caption), and these have been linked into 244,035 coreference chains (7.7 per image). The box drawing process has yielded 275,775 bounding boxes in the 31,783 images (8.7 per image). Figure 3.5 shows the distribution of coreference chains, mentions, and bounding boxes across types, and Table 3.4 shows additional coreference chain statistics. 48.6% of the chains contain more than a single mention. The number of mentions per chain varies significantly across entity types, with salient entities such as people or animals being mentioned more frequently than clothing or body parts.

Aggregating across all five captions, people are mentioned in 94.2% of the images, animals in 12.0%, clothing and body parts in 69.9% and 28.0%, vehicles and instruments in 13.8% and 4.3%, while other objects are mentioned in 91.8% of the images. The scene is mentioned in 79.7% of images. 59.1% of the coreference chains are associated with a single bounding box,

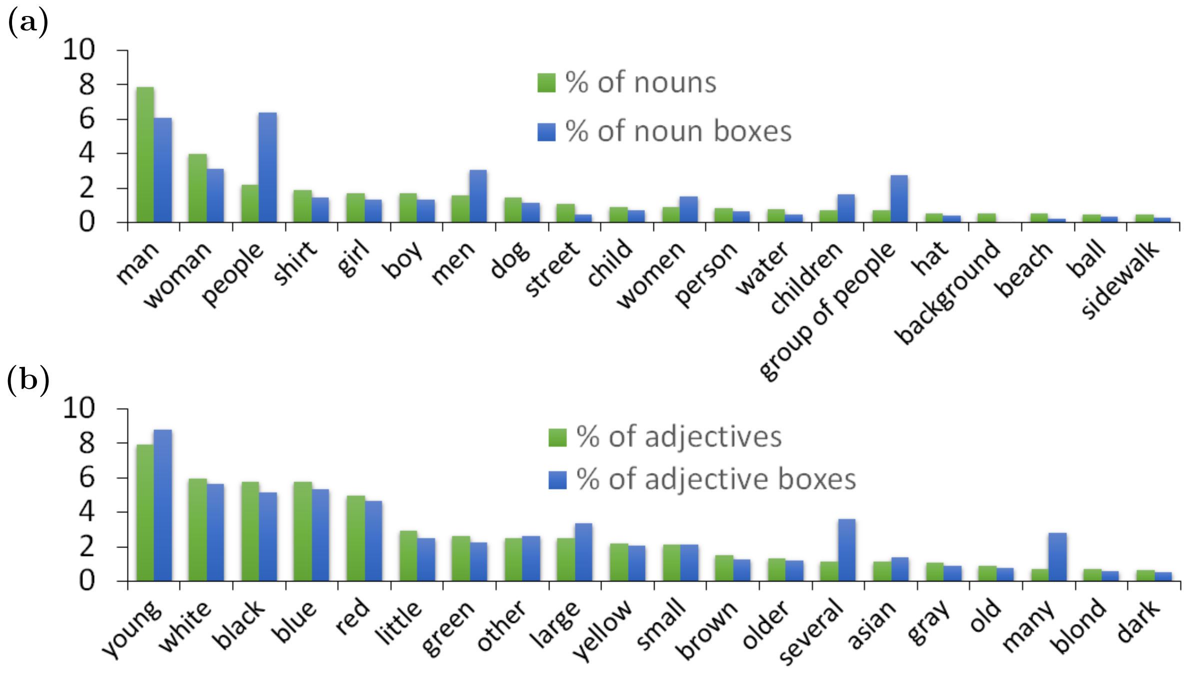


Figure 3.6: Proportion of bounding boxes and occurrences across Flickr30k Entities of the most common **(a)** nouns and **(b)** adjectives.

20.0% with multiple bounding boxes (with at least one such chain in 67.0% of images), and 20.9% with no bounding box, but there is again wide variety across entity types. The people category has significantly more boxes than chains (116k boxes for 60k chains) suggesting that many of these chains describe multiple individuals (*a family*, *a group of people*, etc.). On average, each bounding box in our dataset has IOU of 0.37 with one other ground truth box and 49.2% of boxes are completely enclosed by another ground truth box.

The 20 most common nouns and adjectives with their proportions of total boxes and occurrences are shown in Figure 3.6. Unsurprisingly, common nouns referring to people dominate, and adjectives referring to color appear quite often. Some phrases that could be referring to a scene or a specific image region are also quite common (e.g. *street, water*), providing a glimpse at the challenge faced when attempting to localize phrases since one would have to first identify the sense with which a phrase is being used.

3.2 EXPERIMENTAL EVALUATION

Our main motivation in collecting Flickr30k Entities is to further the development of methods that can reason about detailed correspondences between phrases in text and regions in an image. To evaluate this ability, we propose the following *phrase localization* benchmark:

given an image and a ground-truth sentence that describes it, predict a bounding box (or bounding boxes) for each of the entity mentions (NP chunks) from that sentence. In Section 3.2.1 we present a strong phrase localization baseline trained with our annotations, and in Section 3.2.2, we attempt to use it to improve performance on the standard task of bidirectional image-sentence retrieval.

3.2.1 Phrase Localization

Region-Phrase Model

We have developed a baseline approach for phrase localization that scores each region-phrase correspondence separately, without taking into account any context or performing any joint inference about the global correspondence between all regions in the image and all phrases in the sentence. This approach learns an embedding of region and phrase features to a shared latent space and uses distance in that space to retrieve image regions given a phrase. While there have been several neural network-based approaches for learning such embeddings [6, 7, 11], using state-of-the-art text and image features with Canonical Correlation Analysis (CCA) [42] continues to produce remarkable results [38, 8, 71], and is also much faster to train than a neural network. Given two sets of matching features from different views (in our case, image and text features), CCA finds linear projections of both views into a joint space of common dimensionality in which the correlation between the views is maximized.

Our implementation generally follows the details in [8]. Given a phrase, we represent each word with a 300-D word2vec feature [102] encoding only nouns, adjectives, and prepositions. Then we construct a Fisher Vector codebook [103] with 30 centers using a Hybrid Gaussian-Laplacian Mixture Model (HGLMM),[2] resulting in phrase features of dimensionality $300 \times 30 \times 2 = 18,000$. As in [8], we report results using the 4096-dimensional activations of the 19-layer VGG model [104], using a single crop of each ground truth region. We experiment with both classification and detection variants of the VGG network: the former is trained on the ImageNet dataset [105] and the latter is the Fast RCNN network [50] fine-tuned on a union of the PASCAL 2007 and 2012 trainval sets [106].

An important implementation issue for training the CCA model is how to sample region-phrase correspondences from the training dataset. If we train CCA using all region-phrase correspondences, we get poor performance because the distribution of region counts for

[2] Although in [8] their combined HGLMM+GMM Fisher Vectors performed the best on bidirectional retrieval, in our experiments the addition of the GMM features made no substantial impact on performance.

different NP chunks is very unbalanced: a few NP chunks, like *a man*, are extremely common, while others, like *tattooed, shirtless young man*, occur quite rarely. We found we can alleviate this problem by keeping at most N randomly selected exemplars for each phrase, and we get our best results by resampling the dataset with $N = 10$ regions per phrase. It is also important to note that in some images, a phrase can be associated with multiple regions (e.g., *two men*). In such cases, we merge the regions into a single bounding box for simplicity (although in follow-up work, it would be much more satisfying to detect the individual instances separately).

Consistent with [8], we set the CCA output dimensionality to 4096. To score region-phrase pairs using the learned CCA embedding, we use the normalized formulation of [107], where we scale the columns of the CCA projection matrices by the eigenvalues and normalize feature vectors projected by these matrices to unit length. In the resulting space, we use cosine distance to rank image regions given a phrase.

Evaluation Protocol

At test time we assume we are given an image and a set of NP chunks from all of its ground truth captions.[3] We use the EdgeBox region proposal method [108] to extract a set of candidate object regions from the test image. Experimentally, we found 200 proposals to give us the best performance. Then, for each phrase, we rank the proposal regions using the CCA model and perform non-maximum suppression using a 0.5 IOU threshold.

Following [38, 6, 8, 11], we split Flickr30K into 29,783 training, 1,000 validation, and 1,000 test images. Our split is the same as in [38]. We evaluate localization performance by treating the phrase as the query to retrieve the proposals from the input image and report Recall@K ($K = 1, 5, 10$), or the percentage of queries for which a correct match has rank of at most K (we deem a region to be a correct match if it has IOU ≥ 0.5 with the ground truth bounding box for that phrase).

Note that in the initial version of this work [26], we reported average precision (AP) numbers in addition to recall. However, our annotations are very sparse: there are many valid regions corresponding to some phrases, especially body parts and clothing, that lack ground truth bounding boxes because they are never mentioned in captions. This pervasive reporting bias for some phrase types, combined with the rarity of other phrase types, makes

[3] We use ground truth NP chunks and ignore the non-visual mentions (i.e., mentions not associated with a box). The alternative evaluation method is to extract the phrases automatically, which introduces chunking errors and lowers our recall by around 3%. To the best of our knowledge, the competing methods in Table 3.5(a) also evaluate using ground-truth NP chunks.

	Methods	R@1	R@5	R@10
(a)	c-MWP [86]	27.0	49.9	57.7
	SCRC [88] - VGG19	27.8	–	62.9
	GroundeR [83] - VGG19	41.56	–	–
	SMPL [90] - Fast RCNN	42.08	–	–
	NonlinearSP [85] - Fast RCNN	43.89	64.46	68.66
	GroundeR [83] - Fast RCNN	47.70	–	–
	MCB [82] - Fast RCNN	48.69	–	–
(b)	CCA - VGG19	30.83	58.01	67.15
	CCA - Fast RCNN	41.77	64.52	70.77
(c)	CCA or Detector	42.58	65.26	71.28
	CCA+Detector	43.84	65.83	71.75
(d)	CCA+Detector+Size	49.22	69.93	74.90
	CCA+Detector+Color	45.79	67.23	72.86
	CCA+Detector+Size+Color	**50.89**	**71.09**	**75.73**

Table 3.5: Overall phrase localization performance across the Flickr30k Entities test set. (a) Competing state-of-the-art methods. Note that these works use 100 Selective Search [109] or EdgeBox proposals while we use 200 EdgeBox proposals. (b-d) Variants of our CCA model with different features or additional score terms added (see text for details).

AP too unreliable. Thus, consistent with other works that perform evaluation on Flickr30K Entities [82, 88, 83, 90], we only report recall in this paper.

Phrase Localization Experiments

Table 3.5 summarizes the results of our phrase localization experiments. For reference, part (a) of the table lists recent results on this task which generally fall under two categories: LSTM-based methods [88, 83, 82] and those that use a shallow neural network to learn an embedding between text and image features [85, 90]. We also include the performance of the neural attention model of [86], but note that it is a weakly supervised method trained on outside data.

From Table 3.5(b), we can see that switching from the classification-based VGG19 network, which was used in the initial version of our work [26], to the detection-based Fast RCNN network improves accuracy significantly, which is consistent with the observations of [83]. However, the localization quality of CCA is fundamentally limited because it is trained only on positive examples (ground-truth regions and corresponding phrases). Ideally, we would prefer to use an actual detector that is also trained using negative examples, i.e., poorly localized and background regions. On the other hand, by using a continuous text

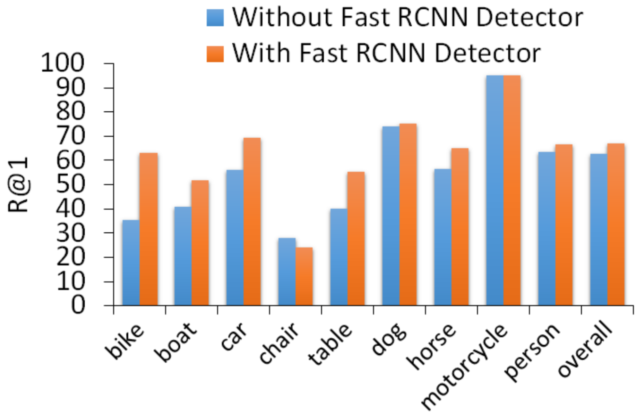


Figure 3.7: Comparison over PASCAL object categories that occur at least 20 times in the test set showing how averaging the CCA score with the output of the Fast RCNN detector affects phrase localization performance.

embedding, CCA can better cope with rare and unseen phrases, as well as phrases that are semantically related. To combine the advantages of both models, we put together the following hybrid scheme.

We manually created mappings from subsets of phrases in our dataset to the 20 PASCAL object categories. These mappings affect 25.32% of all our phrases, 83.4% of which are from the "person" type. When we encounter one of these phrases at test time, we score the proposal regions using the full detection machinery of [50], including bounding box regression. We then get a combined score for phrase ϕ and region r by averaging the detector and CCA scores:

$$D_{CCA+det}(\phi, r) = 0.5\, D_{CCA}(\phi, r) + 0.5(1 - \sigma_{det}(\phi, r))\,, \tag{3.1}$$

where D_{CCA} is the cosine CCA distance (which is between 0 and 1), and σ_{det} is the softmax detector score (which is also between 0 and 1). For phrases that do not correspond to a pre-trained detector, we use only the CCA score. As can be seen from Table 3.5(c), using the detector score alone for phrases that have it is better than using the CCA score alone, and using a combination of both works the best. Figure 3.7 compares the performance of CCA-only with the combined score over PASCAL categories that occur at least 20 times in our test set.

Next, we introduce two more additions to our CCA+Detector model to make it a very strong baseline indeed, rivaling the more complex method of [83]. First, we observe that we can get a big improvement by introducing a bias towards larger regions. In fact, simply selecting the largest proposal regardless of the phrase already gets R@1 of 24%. To trade

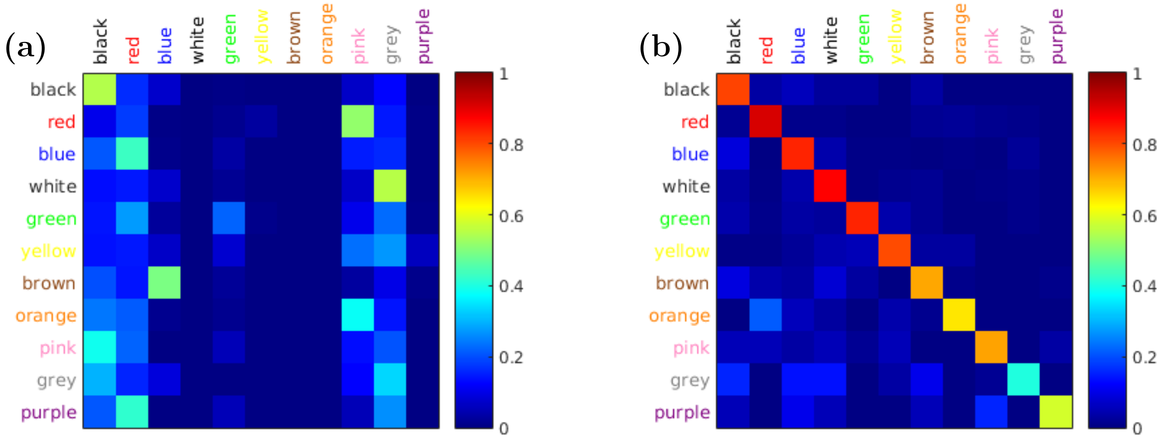


Figure 3.8: Confusion matrices for color classification on the test set using **(a)** linear SVM trained on fc7 features computed from a Fast RCNN network fine-tuned on PASCAL object classes or **(b)** a Fast RCNN network trained to predict colors. Colors are ordered from most to least prevalent in the dataset.

off the appearance-based score with the region size, we define the following combined score:

$$D_{CCA+det+size}(\phi, r) = \\ (1 - w_{size}) D_{CCA+det}(\phi, r) + w_{size}(1 - size(r)) \,, \tag{3.2}$$

where $size(r)$ is the proportion of the image area the region r covers. The weight w_{size} is separately determined for each of our eight phrase types based on the validation set (it is 0.2 for scene, vehicle, and instrument types, and 0.1 for everything else). The first line of Table 3.5(d) shows this simple method works remarkably well, increasing R@1 and mAP by about six points.

Color can also be a strong indicator of the location of a phrase in an image, especially for clothing. However, image features fine-tuned for object detection, where objects of different colors may fall under the same category, turn out to be relatively insensitive to color. Specifically, if we train an SVM classifier on top of Fast RCNN features to predict one of eleven colors that occur at least 1,000 times in the training dataset, we get only 16% accuracy (see Figure 3.8(a)). To obtain a better color predictor for bounding boxes, we fine-tuned the Fast RCNN network on these eleven colors. To avoid confusion with color terms that refer to race, we excluded people phrases from training and testing. We used a softmax loss (i.e., color classification is assumed to be one-vs-all) and fine-tuned the whole network with 0.001 learning rate, 0.0005 weight decay, and 0.9 momentum for 20K iterations. As can be seen in Figure 3.8(b), the resulting network has a much higher accuracy of 80.47%.

With our new color classifier, we add a color term to eq. (2) to obtain our full model:

$$\begin{aligned} D_{full}(\phi, r) = & \\ & (1 - w_{size} - w_{color}) D_{CCA+det}(\phi, r) + \\ & w_{size}(1 - size(r)) + w_{color}(1 - \sigma_{color}(\phi, r)) \, , \end{aligned} \tag{3.3}$$

where $\sigma_{color}(\phi, r)$ is the softmax output of the classifier for the color mentioned in phrase ϕ. We use this term for phrases that mention a color,[4] and eq. (2) otherwise. As can be seen from Table 3.6, the resulting CCA+Size+Color model mainly improves the accuracy for the clothing phrase type, but because this type is so common, this leads to an approximately 1.5% improvement on the entire test set (last two lines of Table 3.5(d)).

Phrase Localization Discussion

As can be seen in the last line of Table 3.5(d), our full model performs relatively well, accurately localizing a phrase more than 50% of the time in an image that contains that phrase. Table 3.6 shows a detailed breakdown that gives an idea of how our different cues contribute to the performance on different phrase types, and the relative difficulty of these phrase types. We can see that adding the size term gives the biggest improvement for vehicles and scenes. For phrases from the scene type, we also experimented with simply predicting the whole image, but that did not give better performance, possibly due to the ambiguity of some phrases (in some cases, *building* may refer to the whole image, and in some cases, it may refer to an object that occupies just a part of the image). As mentioned above, the color term gives the biggest improvement for clothing. It also helps with the body parts mainly due to improved ability to detect hair based on color (brown, black, gray, and even blue or pink).

In absolute terms, we get by far the lowest accuracy on body parts, followed by clothing and instruments (though the latter have just a few instances). This difficulty is due at least in part by the poor coverage that our region proposals give for these classes – as can be seen from the "Proposal upper bound" line of Table 3.6, only about 50% of body parts and 77% of clothing items have a box in our entire set of 200 region proposals with at least 50% IoU. We found that simply adding more region proposals decreased the precision for these phrase types, so their complex appearance adds to the challenge as well.

Figure 3.9(a) analyzes the sources of errors our model makes, showing that confusion between phrases is one of the biggest sources. Figure 3.9(b) shows a confusion matrix

[4] If a phrase includes more than one color, all the color mentions are ignored.

	people	clothing	body parts	animals	vehicles	instruments	scene	other
#Instances	5,656	2,306	523	518	400	162	1,619	3,374
CCA+Detector	61.24	36.90	15.30	62.74	59.75	31.48	31.93	25.34
CCA+Detector+Size	64.73	39.20	15.49	64.09	67.75	37.65	51.33	30.50
CCA+Detector+Size+Color	64.73	46.88	17.21	65.83	68.75	37.65	51.39	31.77
Proposal upper bound (R@200)	96.52	77.36	50.48	91.12	94.50	80.86	83.01	75.87

Table 3.6: Localization performance accuracy over phrase types to rank 200 object proposals per image.

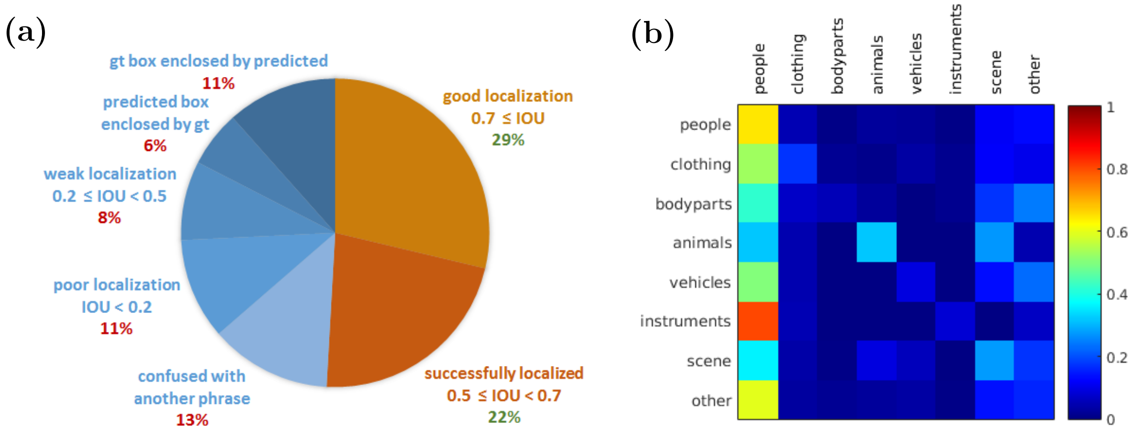


Figure 3.9: **(a)** A breakdown of the R@1 localization performance of our full model. **(b)** Confusion matrix for the 13% of phrases that get confused with another phrase. The entry in row i and column j shows how often a phrase of type i is localized to a box corresponding to phrase of type j. For example, how often does a poorly localized bounding box for a phrase of type "clothing" have $\geq$ 0.5 IOU with the ground truth box for a phrase of type "people"? The matrix calls attention to a pattern of predicting a bounding box for a person when the model is unsure about the location of a phrase.

between different phrase types, revealing a bias towards predicting bounding boxes for a person. Figure 3.10 shows the accuracies for the 25 most frequent phrases in our test set.

Figure 3.11(a) shows examples of relatively successful localization in three images. Our model can find small objects (e.g. *a tennis ball* in the left example and *a microphone* in the middle). In the middle example, it can correctly distinguish the man from the woman. Three typical failure modes are shown in Figure 3.11(b), reflecting our difficulties with localizing body parts and correctly disambiguating person instances. In the leftmost example, three different people phrases are localized to the same box. In the middle example, the bounding box for *arm* localizes the man's visible left arm, instead of the mentioned but mostly occluded *arm around a woman*. In the right example, there are several revealing errors. The bounding

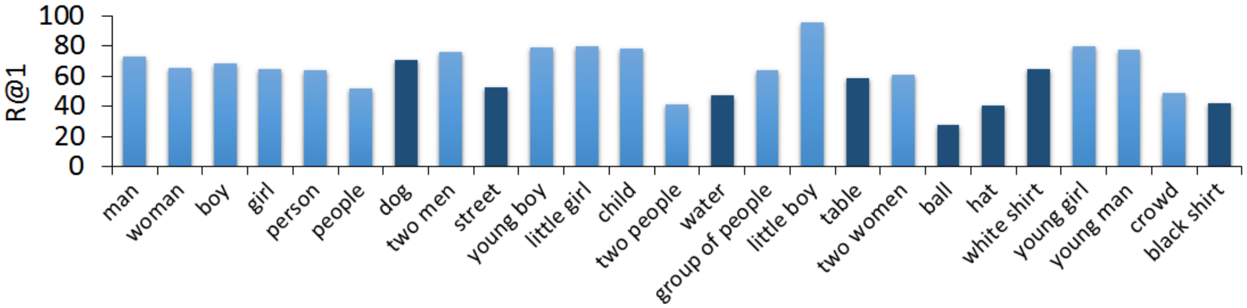


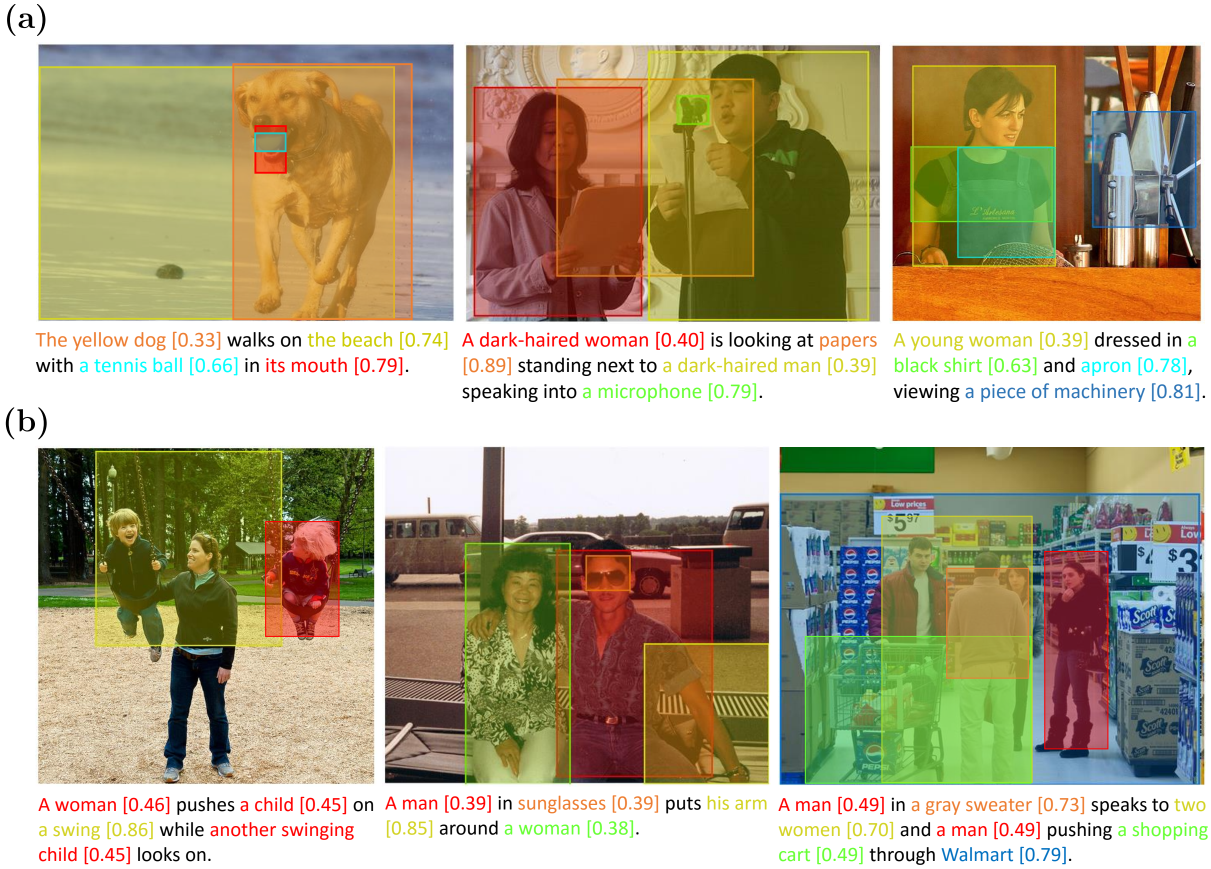
Figure 3.10: Localization performance of 25 of the most common phrases in the test set using our full model ranking 200 object proposals per image. Darker color indicates phrases that are not from the people type.

box for *two women*, while enclosing multiple people, is incorrect. Further, the boxes for two separate instances of *man* incorrectly land on the same woman even though the *gray sweater* belonging to one of the men is correctly localized. This is not surprising, since our model uses the phrase itself without any surrounding sentence context, so multiple instances whose mentions are identical must necessarily be localized to the same box; there is also no constraint in our model to enforce co-location of people and clothing or body parts.

In order to go beyond our baseline, it is necessary to develop methods that can decode the textual cues about cardinalities of entities and relationships between them, and translate these cues into constraints on the localized regions. In particular, since people are so important for our dataset and for image description in general, it is necessary to parse a sentence to determine how many distinct persons are in an image, which mentions of clothes and body parts belong to which person, and impose appropriate constraints on the respective bounding boxes. This is the subject of Chapter 4.

3.2.2 Image-Sentence Retrieval

Next, we would like to demonstrate the usefulness of phrase localization for the well-established benchmark of bidirectional image-sentence retrieval: given an image, retrieve the best-fitting sentence from a pre-existing database, and vice versa. For this, we will start with a state-of-the-art CCA model trained on whole images and sentences, which already does a very good job of capturing the global content of the two modalities, and attempt to refine it using the region-phrase model of Section 3.2.1. Here, the region-phrase model has to succeed at a more difficult task than in Section 3.2.1: instead of scoring regions in an image to localize a phrase that is assumed to be present, it has to compare scores for different region-phrase combinations in an attempt to determine which combination provides



(a)

The yellow dog [0.33] walks on the beach [0.74] with a tennis ball [0.66] in its mouth [0.79].

A dark-haired woman [0.40] is looking at papers [0.89] standing next to a dark-haired man [0.39] speaking into a microphone [0.79].

A young woman [0.39] dressed in a black shirt [0.63] and apron [0.78], viewing a piece of machinery [0.81].

(b)

A woman [0.46] pushes a child [0.45] on a swing [0.86] while another swinging child [0.45] looks on.

A man [0.39] in sunglasses [0.39] puts his arm [0.85] around a woman [0.38].

A man [0.49] in a gray sweater [0.73] speaks to two women [0.70] and a man [0.49] pushing a shopping cart [0.49] through Walmart [0.79].

Figure 3.11: Example phrase localization results. For each image and reference sentence, phrases and top matching regions are shown in the same color. The matching score is given in brackets after each phrase (low scores are better).

the best description of the image.

To get the best global image representation, we use the ImageNet-trained 19-layer VGG network and average the whole-image features over ten crops. Apart from this, we follow the implementation details of Section 3.2.1 to train an image-sentence CCA model that is essentially a reimplementation of [8]. Given the model, we compute the normalized projections of the image and sentence features into the CCA space and do image-to-sentence and sentence-to-image retrieval using the cosine distance.

For evaluation, we use the standard protocol for Flickr30k: given the 1,000 images and 5,000 corresponding sentences in the test set, we use the images to retrieve the sentences and vice versa, and report performance as Recall@K, or the percentage of queries for which at least one correct ground truth match was ranked among the top K matches. Table 3.7 shows the results. As can be seen by comparing Table 3.7(a) and (b), the global CCA has consistent performance with [8] and is competitive with the state of the art, which includes

	Methods on Flickr30k	Image Annotation			Image Search		
		R@1	R@5	R@10	R@1	R@5	R@10
(a) State of the art	HGLMM+GMM [8]	33.3%	62.0%	74.7%	25.6%	53.2%	66.8%
	m-RNN [11]	35.4%	63.8%	73.7%	22.8%	50.7%	63.1%
	mCNN [70]	33.6%	**64.1%**	74.9%	26.2%	**56.3%**	69.6%
	RNN FV [71]	35.6%	62.5%	74.2%	**27.4%**	55.9%	**70.0%**
(b) Whole image-sentence CCA	HGLMM+VGG19	36.5%	62.2%	73.3%	24.7%	53.4%	66.8%
(c) Image-sentence +region-phrase	WD VGG19	37.0%	62.9%	73.9%	25.7%	54.5%	67.6%
	WD RtP	**37.5%**	62.9%	**75.1%**	25.8%	54.7%	67.6%

Table 3.7: Bidirectional retrieval results. Image Annotation refers to using images to retrieve sentences, and Image Search refers to using sentences to retrieve images. The numbers in (a) come from published papers, and the numbers in (b) are from our own reproduction of the results of [8] using their code. See Section 3.2.2 for additional details.

complex CNN and RNN models.

Next, we want to add region-phrase correspondences to get a further improvement on image-sentence matching. Given an image I and a sentence S (which may or may not correctly describe the image), for each phrase ϕ_i, $i = 1, \ldots, L$, we find the best-matching candidate region r_j using the region-phrase CCA embedding.[5] Then, similarly to [6], we compute the overall image-sentence distance as the sum of the region-phrase distances:

$$D_{PR}(S, I) = \frac{1}{L^\gamma} \sum_{i}^{L} \min_j D_{full}(\phi_i, r_j)\,, \tag{3.4}$$

where D_{full} is our full region-phrase model (eq. 3) and the exponent $\gamma \geq 1$ is meant to lessen the penalty associated with matching images to sentences with a larger number of phrases, since such sentences tend to mention more details that are harder to localize. Experimentally, we have found $\gamma = 1.5$ to produce the best results. Finally, we define a combined image-sentence distance as

$$D_{SI} = \alpha\, D_{CCA}(S, I) + (1 - \alpha)\, D_{PR}(S, I)\,, \tag{3.5}$$

where $D_{CCA}(S, I)$ is the normalized CCA distance between the whole-image and whole-sentence feature vectors.

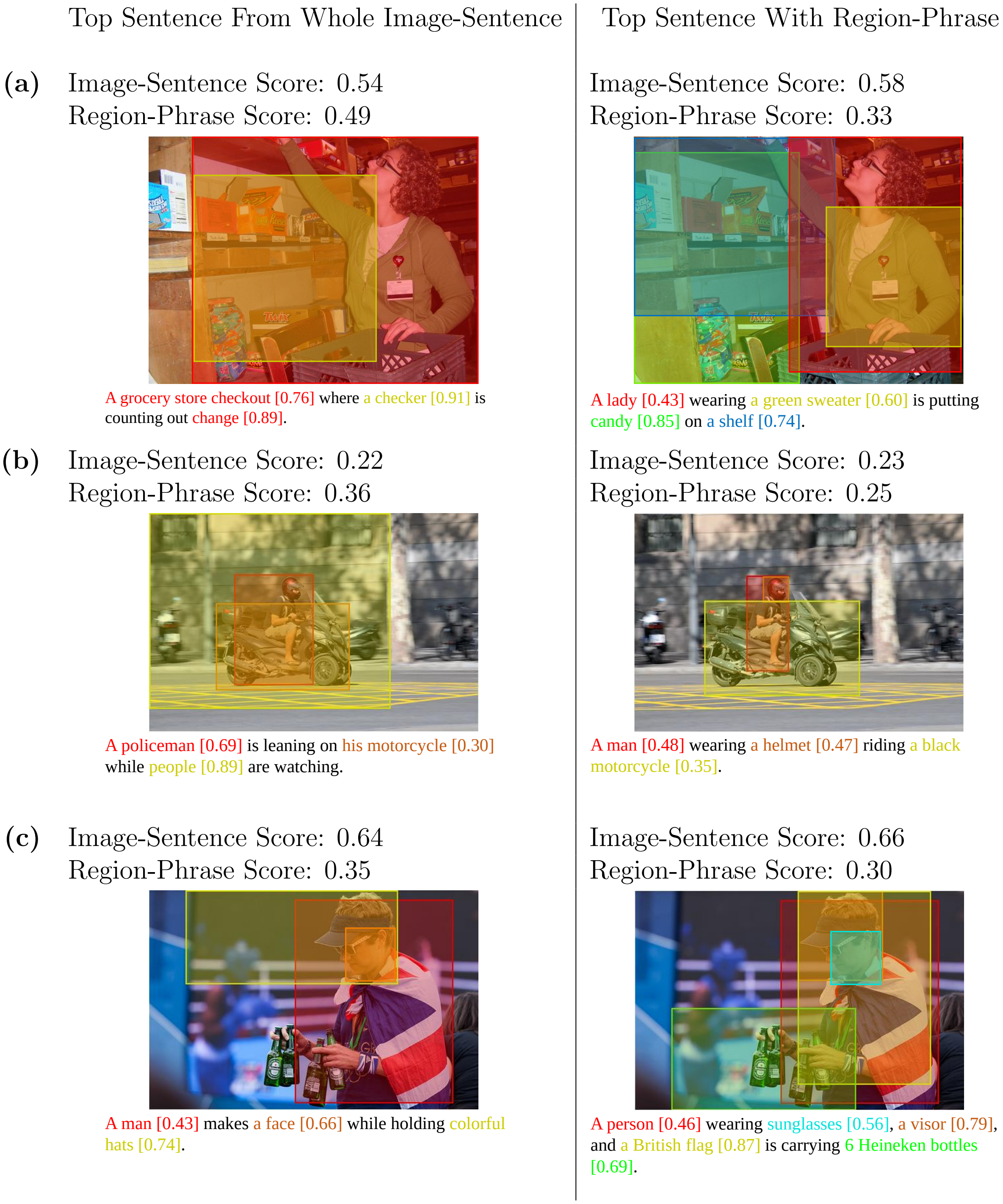
Table 3.7(c) shows results of this weighted distance with $\alpha = 0.7$. By itself, the performance of eq. (3.4) is very poor, but when combined with $D_{CCA}(I, S)$, it gives a small but consistent improvement of 1%-2%. For completeness, the two lines of the Table 3.7(c) com-

[5] Here, as in Section 3.2, our phrases are ground-truth NP chunks, but unlike in Section 3.2, we do not exclude NP chunks corresponding to non-visual concepts.

pare the performance of our full region-phrase model to just the basic VGG model. Despite big differences in R@1 for phrase localization (Table 3.5), the two models perform similarly for image-sentence retrieval. To understand why it is so difficult to get an improvement in image-sentence retrieval by incorporating increasingly accurate phrase localization models, it helps to examine retrieval results qualitatively.

First, Figure 3.12 illustrates cases in which our region-phrase model does improve image-sentence retrieval performance. In examples (a) and (b), the top retrieved sentences using the whole image-sentence model (left column) are incorrect but somewhat plausible. However, the region-phrase model is unable to locate some the phrases from those sentences with any degree of confidence (e.g., *a checker* in (a), *people* in (b)). However, the phrases of the correct sentences (right column) have much better region-phrase scores that compensate for the slightly worse whole image-sentence scores. The third example shows how our normalization term in eq. (3.4) helps longer sentences, which tend to have entities that are more difficult to localize.

Despite the encouraging examples above, why is the overall quantitative improvement afforded by region-phrase correspondences so small? As we can see from the left column of Figure 3.12, the global image-sentence CCA model usually succeeds in retrieving sentences that roughly fit the image. In order to provide an improvement, the region-phrase model must make fine distinctions, which is precisely where it tends to fail. Figure 3.13 shows two examples of this phenomenon. For the first example image, the top sentence retrieved by our model includes a man, a striped shirt, and glasses, all with correct localizations in the image. There is also an incorrect, but plausible, localization of a microphone. However, our model is not discerning enough to figure out that the found instances of shirt and glasses do not belong to the man and that *a man and a woman wearing costume glasses* is a more accurate interpretation of the image than *a man with a striped shirt and glasses.* For the second example, the top retrieved sentence mentions a woman who is not there (and who our phrase localization model co-locates with the man). In order to make all of the above distinctions, we need not only a much more precise local appearance model, but a global contextual inference algorithm. This is the topic of Chapter 4.



Top Sentence From Whole Image-Sentence

Top Sentence With Region-Phrase

(a) Image-Sentence Score: 0.54
Region-Phrase Score: 0.49

A grocery store checkout [0.76] where a checker [0.91] is counting out change [0.89].

Image-Sentence Score: 0.58
Region-Phrase Score: 0.33

A lady [0.43] wearing a green sweater [0.60] is putting candy [0.85] on a shelf [0.74].

(b) Image-Sentence Score: 0.22
Region-Phrase Score: 0.36

A policeman [0.69] is leaning on his motorcycle [0.30] while people [0.89] are watching.

Image-Sentence Score: 0.23
Region-Phrase Score: 0.25

A man [0.48] wearing a helmet [0.47] riding a black motorcycle [0.35].

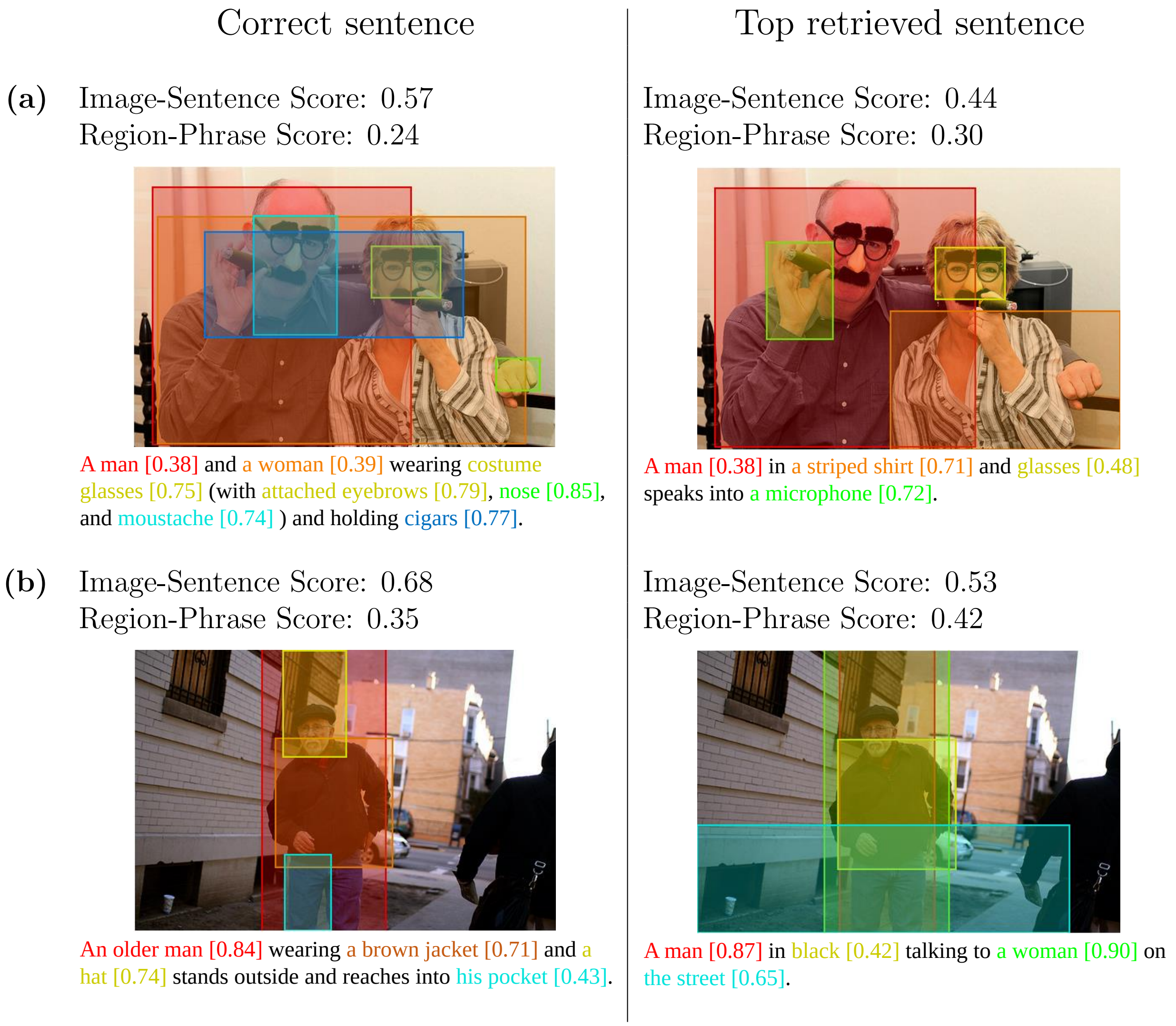
(c) Image-Sentence Score: 0.64
Region-Phrase Score: 0.35

A man [0.43] makes a face [0.66] while holding colorful hats [0.74].

Image-Sentence Score: 0.66
Region-Phrase Score: 0.30

A person [0.46] wearing sunglasses [0.56], a visor [0.79], and a British flag [0.87] is carrying 6 Heineken bottles [0.69].

Figure 3.12: Example image-sentence retrieval results where adding region-phrase correspondences helps to retrieve the correct sentence. For each test image, the left column shows the top retrieved sentence using the whole image-sentence model and the right column shows the top sentence retrieved by our full model. For each image and reference sentence, phrases and top matching regions are shown in the same color. The matching score is given in brackets after each phrase (low scores are better).



Correct sentence

Top retrieved sentence

(a) Image-Sentence Score: 0.57
Region-Phrase Score: 0.24

A man [0.38] and a woman [0.39] wearing costume glasses [0.75] (with attached eyebrows [0.79], nose [0.85], and moustache [0.74]) and holding cigars [0.77].

Image-Sentence Score: 0.44
Region-Phrase Score: 0.30

A man [0.38] in a striped shirt [0.71] and glasses [0.48] speaks into a microphone [0.72].

(b) Image-Sentence Score: 0.68
Region-Phrase Score: 0.35

An older man [0.84] wearing a brown jacket [0.71] and a hat [0.74] stands outside and reaches into his pocket [0.43].

Image-Sentence Score: 0.53
Region-Phrase Score: 0.42

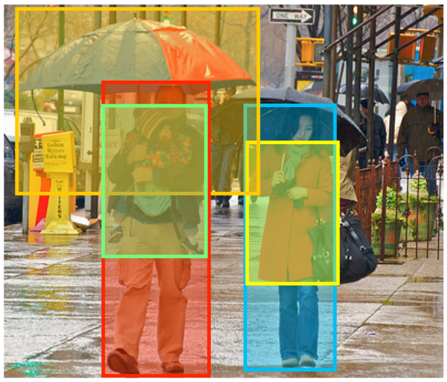
A man [0.87] in black [0.42] talking to a woman [0.90] on the street [0.65].

Figure 3.13: Example image-sentence retrieval results where region-phrase correspondences do not help to retrieve the correct sentence. For each test image, the left column shows a ground-truth sentence and the right column shows the top sentence retrieved by our method. For each image and reference sentence, phrases and top matching regions are shown in the same color. The matching score is given in brackets after each phrase (low scores are better).

CHAPTER 4: PHRASE LOCALIZATION WITH COMPREHENSIVE IMAGE-LANGUAGE CUES

Input Sentence and Image

A man carries a baby under a red and blue umbrella next to a woman in a red jacket



Cues	Examples
1) Entities	man, baby, umbrella, woman, jacket
2) Candidate Box Position	——
3) Candidate Box Size	——
4) Common Object Detectors	man → person baby → person woman → person
5) Adjectives	umbrella → red umbrella → blue jacket → red
6) Subject - Verb	(man, carries)
7) Verb – Object	(carries, baby)
8) Verbs	(man, carries, baby)
9) Prepositions	(baby, under, umbrella) (man, next to, woman)
10)Clothing & Body Parts	(woman, in, jacket)

Figure 4.1: Left: an image and caption, together with ground truth bounding boxes of entities (noun phrases). Right: a list of all the cues used by our system, with corresponding phrases from the sentence.

In Chapter 3 we introduced the task of phrase localization and proposed a method which combined appearance, size, and color cues to locate phrases in an image. However, this approach did not learn the combination weights and independently localized each phrase without taking their relationships into account. In this chapter we extend our baseline approach to use a larger set of cues, learned combination weights, and a global optimization method for simultaneously localizing all the phrases in a sentence. Figure 4.1 introduces the components of our system with an example image and caption.

Table 4.1 compares the cues used in our work to those in other recent papers on phrase localization and related tasks like image retrieval and referring expression understanding. To date, other methods applied to the Flickr30K Entities dataset [82, 88, 83, 110, 90] have used a limited set of single-phrase cues. Information from the rest of the caption, like verbs and prepositions indicating spatial relationships, has been ignored. One exception is Wang *et al.* [90], who tried to relate multiple phrases to each other, but limited their relationships only to those indicated by possessive pronouns, not personal ones. By contrast, we use pronoun cues to the full extent by performing pronominal coreference. Also, ours is the only work in this area incorporating verbs to perform action recognition.

	Methods applied to Flickr30K Entities						Models on related tasks		
	ours	NonlinearSP [110]	GroundeR [83]	MCB [82]	SCRC [88]	SMPL [90]	Scene Graph [66]	ReferIt [24]	Google RefExp [111]
(a) Region-Phrase Compatibility	✓	✓	✓	✓	✓	✓	–	–	✓
Candidate Position	✓	–	–	–	✓	–	–	✓	✓
Candidate Size	✓	–	–	–	–	–	–	✓	✓
Object Detectors	✓*	–	–	–	–	–	✓	✓	–
Adjectives	✓	–	–	–	–	–	✓	✓*	–
Verbs	✓	–	–	–	–	–	–	–	–
(b) Relative Position	✓	–	–	–	–	✓*	✓	✓	–
Clothes&Body Parts	✓	–	–	–	–	–	–	–	–
(c) Joint Localization	✓	–	–	–	–	✓	✓	–	–

Table 4.1: Comparison of cues for phrase-to-region grounding: **(a)** single phrase cues, **(b)** pair phrase cues, and **(c)** interference method. * indicates that the cue is used in a limited fashion, *i.e.* [24] restricted their adjective cues to colors, [90] only modeled possessive pronoun phrase-pair spatial cues ignoring verb and prepositional phrases, we limit the object detectors to 20 common categories.

4.1 PHRASE LOCALIZATION APPROACH

We follow the task definition used in [82, 88, 27, 83, 110, 90]: At test time, we are given an image and a caption with a set of entities (noun phrases), and we need to localize each entity with a bounding box. Section 4.1.1 describes our inference formulation, and Section 4.1.2 describes our procedure for learning the weights of different cues.

4.1.1 Joint phrase localization

For each image-language cue derived from a single phrase or a pair of phrases (Figure 4.1), we define a *cue-specific cost function* that measures its compatibility with an image region (small values indicate high compatibility). We will describe the cost functions in detail in Section 4.2; here, we give our test-time optimization framework for jointly localizing all phrases from a sentence.

Given a single phrase p from a test sentence, we score each region (bounding box) proposal b from the test image based on a linear combination of cue-specific cost functions $\phi_{\{1,\cdots,K_S\}}(p,b)$ with learned weights w^S:

$$S(p,b;w^S)=\sum_{s=1}^{K_S}\mathbb{1}_s(p)\phi_s(p,b)w_s^S, \tag{4.1}$$

where $\mathbb{1}_s(p)$ is an indicator function for the availability of cue s for phrase p (e.g., an adjective cue would be available for the phrase *blue socks*, but would be unavailable for *socks* by itself). As will be described in Section 4.2.2, we use 14 single-phrase cost functions: region-phrase compatibility score, phrase position, phrase size (one for each of the eight phrase types of [27]), object detector score, adjective, subject-verb, and verb-object scores.

For a pair of phrases with some relationship $r = (p, rel, p')$ and candidate regions b and b', an analogous scoring function is given by a weighted combination of pairwise costs $\psi_{\{1,\cdots,K_Q\}}(r, b, b')$:

$$Q(r,b,b';w^Q)=\sum_{q=1}^{K_Q}\mathbb{1}_q(r)\psi_q(r,b,b')w_q^Q. \tag{4.2}$$

We use three pairwise cost functions corresponding to spatial classifiers for verb, preposition, and clothing and body parts relationships (Section 4.2.3).

We train all cue-specific cost functions on the training set and the combination weights on the validation set. At test time, given an image and a list of phrases $\{p_1, \cdots, p_N\}$, we first retrieve top M candidate boxes for each phrase p_i using Eq. (4.1). Our goal is then to select one bounding box b_i out of the M candidates per each phrase p_i such that the following objective is minimized:

$$\min_{b_1,\cdots,b_N}\left\{\sum_{p_i}S(p_i,b_i) \quad + \sum_{r_{ij}=(p_i,rel_{ij},p_j)}Q(r_{ij},b_i,b_j)\right\} \tag{4.3}$$

where phrases p_i and p_j (and respective boxes b_i and b_j) are related by some relationship rel_{ij}. This is a binary quadratic programming formulation inspired by [112]; we relax and solve it using a sequential QP solver in MATLAB. The solution gives a single bounding box hypothesis for each phrase. Performance is evaluated using Recall@1, or proportion of phrases where the selected box has Intersection-over-Union (IOU) ≥ 0.5 with the ground truth.

4.1.2 Learning scoring function weights

We learn the weights w^S and w^Q in Eqs. (4.1) and (4.2) by directly optimizing recall on the validation set. We start by finding the unary weights w^S that maximize the number of correctly localized phrases:

$$w^S = \arg\max_w \sum_{i=1}^{N} \mathbb{1}_{IOU\geq 0.5}(b_i^*, \hat{b}(p_i; w)), \tag{4.4}$$

where N is the number of phrases in the training set, $\mathbb{1}_{IOU\geq 0.5}$ is an indicator function returning 1 if the two boxes have IOU ≥ 0.5, b_i^* is the ground truth bounding box for phrase p_i, $\hat{b}(p; w)$ returns the most likely box candidate for phrase p under the current weights, or, more formally, given a set of candidate boxes $\mathcal{B}$,

$$\hat{b}(p; w) = \min_{b \in \mathcal{B}} S(p, b; w). \tag{4.5}$$

We optimize Eq. (4.4) using a derivative-free direct search method [113] (MATLAB's fminsearch). We randomly initialize the weights, keep the best weights after 20 runs based on validation set performance (takes just a few minutes to learn weights for all single phrase cues in our experiments).

Next, we fix w^S and learn the weights w^Q over phrase-pair cues in the validation set. To this end, we formulate an objective analogous to Eq. (4.4) for maximizing the number of correctly localized region pairs. Similar to Eq. (4.5), we define the function $\hat{\rho}(r; w)$ to return the best pair of boxes for the relationship $r = (p, rel, p')$:

$$\hat{\rho}(r;w) = \min_{b,b' \in \mathcal{B}} S(p,b;w^S) + S(p',b';w^S) + Q(r,b,b';w). \tag{4.6}$$

Then our pairwise objective function is

$$w^Q = \arg\max_{w} \sum_{k=1}^{M} \mathbb{I}_{PairIOU\geq 0.5}(\rho_k^*, \hat{\rho}(r_k; w)), \tag{4.7}$$

where M is the number of phrase pairs with a relationship, $\mathbb{I}_{PairIOU\geq 0.5}$ returns the number of correctly localized boxes (0, 1, or 2), and ρ_k^* is the ground truth box pair for the relationship $r_k = (p_k, rel_k, p'_k)$.

Note that we also attempted to learn the weights w^S and w^Q using standard approaches such as rank-SVM [114], but found our proposed direct search formulation to work better. In phrase localization, due to its Recall@1 evaluation criterion, only the correctness of one best-scoring candidate region for each phrase matters, unlike in typical detection scenarios, where one would like all positive examples to have better scores than all negative examples.

4.2 CUES FOR PHRASE-REGION GROUNDING

Section 4.2.1 describes how we extract linguistic cues from sentences. Sections 4.2.2 and 4.2.3 give our definitions of the two types of cost functions used in Eqs. (4.1) and (4.2): single phrase cues (SPC) measure the compatibility of a given phrase with a candidate bounding box, and phrase pair cues (PPC) ensure that pairs of related phrases are localized in a spatially coherent manner.

4.2.1 Extracting linguistic cues from captions

The Flickr30k Entities dataset provides annotations for Noun Phrase (NP) chunks corresponding to entities, but linguistic cues corresponding to adjectives, verbs, and prepositions must be extracted from the captions using NLP tools. Once these cues are extracted, they will be translated into visually relevant constraints for grounding. In particular, we will learn specialized detectors for adjectives, subject-verb, and verb-object relationships (Section 4.2.2). Also, because pairs of entities connected by a verb or preposition have constrained layout, we will train classifiers to score pairs of boxes based on spatial information (Section 4.2.3).

Adjectives are part of NP chunks so identifying them is trivial. To extract other cues, such as verbs and prepositions that may indicate actions and spatial relationships, we obtain a constituent parse tree for each sentence using the Stanford parser [115]. Then, for possible relational phrases (prepositional and verb phrases), we use the method of Fidler *et al.* [116], where we start at the relational phrase and then traverse up the tree and to the left until we reach a noun phrase node, which will correspond to the first entity in an *(entity1, rel, entity2)* tuple. The second entity is given by the first noun phrase node on the right side of the relational phrase in the parse tree. For example, given the sentence *A boy running in a field with a dog*, the extracted NP chunks would be *a boy, a field, a dog.* The relational phrases would be *(a boy, running in, a field)* and *(a boy, with, a dog).*

Notice that a single relational phrase can give rise to multiple relationship cues. Thus, from *(a boy, running in, a field)*, we extract the verb relation *(boy, running, field)* and prepositional relation *(boy, in, field).* An exception to this is a relational phrase where the first entity is a person and the second one is of the clothing or body part type,[1] e.g., *(a boy, running in, a jacket).* For this case, we create a single special pairwise relation *(boy, jacket)* that assumes that the second entity is attached to the first one and the exact relationship words do not matter, i.e., *(a boy, running in, a jacket)* and *(a boy, wearing, a jacket)* are considered to be the same. The attachment assumption can fail for phrases like *(a boy, looking at, a jacket)*, but such cases are rare.

Finally, since pronouns in Flickr30k Entities are not annotated, we attempt to perform pronominal coreference (i.e., creating a link between a pronoun and the phrase it refers

[1] Each NP chunk from the Flickr30K dataset is classified into one of eight phrase types based on the dictionaries of [27].

to) in order to extract a more complete set of cues. As an example, given the sentence *Ducks feed themselves*, initially we can only extract the subject-verb cue ($ducks, feed$), but we don't know who or what they are feeding. Pronominal coreference resolution tells us that the ducks are themselves eating and not, say, feeding ducklings. We use a simple rule-based method similar to knowledge-poor methods [117, 118]. Given lists of pronouns by type,[2] our rules attach each pronoun with at most one non-pronominal mention that occurs earlier in the sentence (an antecedent). We assume that subject and object pronouns often refer to the main subject (e.g. *[A dog] laying on the ground looks up at the dog standing over [him]*), reflexive and reciprocal pronouns refer to the nearest antecedent (e.g. *[A tennis player] readies [herself].*), and indefinite pronouns do not refer to a previously described entity. It must be noted that compared with verb and prepositional relationships, relatively few additional cues are extracted using this procedure (432 pronoun relationships in the test set and 13,163 in the train set, while the counts for the other relationships are on the order of 10K and 300K).

4.2.2 Single Phrase Cues (SPCs)

Region-phrase compatibility: This is the most basic cue relating phrases to image regions based on appearance. It is applied to every test phrase (i.e., its indicator function in Eq. (4.1) is always 1). Given phrase p and region b, the cost $\phi_{CCA}(p, b)$ is given by the cosine distance between p and b in a joint embedding space learned using normalized Canonical Correlation Analysis (CCA) [119]. We use the same procedure as [27]. Regions are represented by the fc7 activations of a Fast-RCNN model [50] fine-tuned using the union of the PASCAL 2007 and 2012 trainval sets [106]. After removing stopwords, phrases are represented by the HGLMM fisher vector encoding [8] of word2vec [120].

Candidate position: The location of a bounding box in an image has been shown to be predictive of the kinds of phrases it may refer to [121, 88, 24, 122]. We learn location models for each of the eight broad phrase types specified in [27]: people, clothing, body parts, vehicles, animals, scenes, and a catch-all "other." We represent a bounding box by its centroid normalized by the image size, the percentage of the image covered by the box, and its aspect ratio, resulting in a 4-dim. feature vector. We then train a support vector machine (SVM) with a radial basis function (RBF) kernel using LIBSVM [123]. We randomly sample EdgeBox [108] proposals with IOU < 0.5 with the ground truth boxes for negative examples.

[2] Relevant pronoun types are subject, object, reflexive, reciprocal, relative, and indefinite.

Our scoring function is

$$\phi_{pos}(p, b) = -\log(\mathrm{SVM}_{type(p)}(b)),$$

where $\mathrm{SVM}_{type(p)}$ returns the probability that box b is of the phrase type $type(p)$ (we use Platt scaling [124] to convert the SVM output to a probability).

Candidate size: People have a bias towards describing larger, more salient objects, leading prior work to consider the size of a candidate box in their models [116, 24, 27]. We follow the procedure of [27], so that given a box b with dimensions normalized by the image size, we have

$$\phi_{size_{type(p)}}(p, b) = 1 - b_{width} \times b_{height}.$$

Unlike phrase position, this cost function does not use a trained SVM per phrase type. Instead, each phrase type is its own feature and the corresponding indicator function returns 1 if that phrase belongs to the associated type.

Detectors: CCA embeddings are limited in their ability to localize objects because they must account for a wide range of phrases and because they do not use negative examples during training. To compensate for this, we use Fast R-CNN [50] to learn three networks for common object categories, attributes, and actions. Once a detector is trained, its score for a region proposal b is

$$\phi_{det}(p, b) = -\log(\mathrm{softmax}_{det}(p, b)),$$

where $\mathrm{softmax}_{det}(p, b)$ returns the output of the softmax layer for the object class corresponding to p. We manually create dictionaries to map phrases to detector categories (e.g., man, woman, *etc.* map to 'person'), and the indicator function for each detector returns 1 only if one of the words in the phrase exists in its dictionary. If multiple detectors for a single cue type are appropriate for a phrase (e.g., *a black and white shirt* would have two adjective detectors fire, one for each color), the scores are averaged. Below, we describe the three detector networks used in our model. Complete dictionaries can be found in the supplementary material of the published paper [29].

Objects: We use the dictionary of [27] to map nouns to the 20 PASCAL object categories [106] and fine-tune the network on the union of the PASCAL VOC 2007 and 2012 trainval sets. At test time, when we run a detector for a phrase that maps to one of these object categories, we also use bounding box regression to refine the original region proposals. Regression is not used for the other networks below.

Adjectives: Adjectives found in phrases, especially color, provide valuable attribute information for localization [116, 66, 24, 27]. The Flickr30K Entities baseline approach [27] used a network trained for 11 colors. As a generalization of that, we create a list of adjectives that occur at least 100 times in the training set of Flickr30k. After grouping together similar words and filtering out non-visual terms (e.g., *adventurous*), we are left with a dictionary of 83 adjectives. As in [27], we consider color terms describing people (*black man*, *white girl*) to be separate categories.

Subject-Verb and Verb-Object: Verbs can modify the appearance of both the subject and the object in a relation. For example, knowing that a person is riding a horse can give us better appearance models for finding both the person and the horse [125, 37]. As we did with adjectives, we collect verbs that occur at least 100 times in the training set, group together similar words, and filter out those that don't have a clear visual aspect, resulting in a dictionary of 58 verbs. Since a person running looks different than a dog running, we subdivide our verb categories by phrase type of the subject (resp. object) if that phrase type occurs with the verb at least 30 times in the train set. For example, if there are enough animal-running occurrences, we create a new category with instances of all animals running. For the remaining phrases, we train a catch-all detector over all the phrases related to that verb. Following [125], we train separate detectors for subject-verb and verb-object relationships, resulting in dictionary sizes of 191 (resp. 225). We also attempted to learn subject-verb-object detectors as in [125, 37], but did not see a further improvement.

4.2.3 Phrase-Pair Cues (PPCs)

So far, we have discussed cues pertaining to a single phrase, but relationships between pairs of phrases can also provide cues about their relative position. We denote such relationships as tuples $(p_{left}, rel, p_{right})$ with $left, right$ indicating on which side of the relationship the phrases occur. As discussed in Section 4.2.1, we consider three distinct types of relationships: verbs (*man, riding, horse*), prepositions (*man, on, horse*), and clothing and body parts (*man, wearing, hat*). For each of the three relationship types, we group phrases referring to people but treat all other phrases as distinct, and then gather all relationships that occur at least 30 times in the training set. Then we learn a spatial relationship model as follows. Given a pair of boxes with coordinates $b = (x, y, w, h)$ and $b' = (x', y', w', h')$, we compute a four-dim. feature

$$[(x - x')/w,\ (y - y')/h,\ w'/w,\ h'/h]\,, \tag{4.8}$$

and concatenate it with combined SPC scores $S(p_{left}, b)$, $S(p_{right}, b')$ from Eq. (4.1). To obtain negative examples, we randomly sample from other box pairings with IOU < 0.5

with the ground truth regions from that image. We train an RBF SVM classifier with Platt scaling [124] to obtain a probability output. This is similar to the method of [66], but rather than learning a Gaussian Mixture Model using only positive data, we learn a more discriminative model. Below are details on the three types of relationship classifiers.

Verbs: Starting with our dictionary of 58 verb detectors and following the above procedure of identifying all relationships that occur at least 30 times in the training set, we end up with 260 $(p_{left}, rel_{verb}, p_{right})$ SVM classifiers.

Prepositions: We first gather a list of prepositions that occur at least 100 times in the training set, combine similar words, and filter out words that do not indicate a clear spatial relationship. This yields eight prepositions (*in, on, under, behind, across, between, onto, and near*) and 216 $(p_{left}, rel_{prep}, p_{right})$ relationships.

Clothing and body part attachment: We collect $(p_{left}, rel_{c\&bp}, p_{right})$ relationships where the left phrase is always a person and the right phrase is from the clothing or body part type and learn 207 such classifiers. As discussed in Section 4.2.1, this relationship type takes precedence over any verb or preposition relationships that may also hold between the same phrases.

4.3 EXPERIMENTS

4.3.1 Implementation details

We perform experiments on the Flickr30K Entities dataset using the same splits and region proposals as in Chapter 3. At test time, given a sentence and an image, we first use Eq. (4.1) to find the top 30 candidate regions for each phrase after performing non-maximum suppression using a 0.8 IOU threshold. Restricted to these candidates, we optimize Eq. (4.2) to find a globally consistent mapping of phrases to regions.

Consistent with Chapter 3, we only evaluate localization for phrases with a ground truth bounding box. If multiple bounding boxes are associated with a phrase (e.g., four individual boxes for *four men*), we represent the phrase as the union of its boxes. For each image and phrase in the test set, the predicted box must have at least 0.5 IOU with its ground truth box to be deemed successfully localized. As only a single candidate is selected for each phrase, we report the proportion of correctly localized phrases (i.e. Recall@1).

	Method	Accuracy
(a)	**Single-phrase cues**	
	CCA	43.09
	CCA+Det	45.29
	CCA+Det+Size	51.45
	CCA+Det+Size+Adj	52.63
	CCA+Det+Size+Adj+Verbs	54.51
	CCA+Det+Size+Adj+Verbs+Pos (SPC)	**55.49**
(b)	**Phrase pair cues**	
	SPC+Verbs	55.53
	SPC+Verbs+Preps	55.62
	SPC+Verbs+Preps+C&BP (SPC+PPC)	**55.85**
(c)	**State of the art**	
	SMPL [90]	42.08
	NonlinearSP [110]	43.89
	GroundeR [83]	47.81
	MCB [82]	48.69
	RtP [27]	50.89

Table 4.2: Phrase-region grounding performance on the Flickr30k Entities dataset. **(a)** Performance of our single-phrase cues (Sec. 4.2.2). **(b)** Further improvements by adding our pairwise cues (Sec. 4.2.3). **(c)** Accuracies of competing state-of-the-art methods. This comparison excludes concurrent work that was published after our initial submission [91].

4.3.2 Results

Table 4.2 reports our overall localization accuracy for combinations of cues and compares our performance to the state of the art. Object detectors, reported on the second line of Table 4.2(a), show a 2% overall gain over the CCA baseline. This includes the gain from the detector score as well as the bounding box regressor trained with the detector in the Fast R-CNN framework [50]. Adding adjective, verb, and size cues improves accuracy by a further 9%. Our last cue in Table 4.2(a), position, provides an additional 1% improvement.

We can see from Table 4.2(b) that the spatial cues give only a small overall boost in accuracy on the test set, but that is due to the relatively small number of phrases to which they apply. In Table 4.4 we will show that the localization improvement on the affected phrases is much larger.

Table 4.2(c) compares our performance to the state of the art. We shall refer to the phrase localization approach used in Chapter 3 RtP here. RtP relies on a subset of our single-phrase cues (region-phrase CCA, size, object detectors, and color adjectives), and localizes each phrase separately. The closest version of our current model to RtP is CCA+Det+Size+Adj, which replaces the 11 colors used in Chapter 3 with our more general model for 83 adjectives,

	People	Clothing	Body Parts	Animals	Vehicles	Instruments	Scene	Other
#Test	5,656	2,306	523	518	400	162	1,619	3,374
SMPL [90]	57.89	34.61	15.87	55.98	52.25	23.46	34.22	26.23
GroundeR [83]	61.00	38.12	10.33	62.55	**68.75**	36.42	**58.18**	29.08
RtP [27]	64.73	46.88	17.21	65.83	**68.75**	**37.65**	51.39	31.77
SPC+PPC (ours)	**71.69**	**50.95**	**25.24**	**76.25**	66.50	35.80	51.51	**35.98**
Upper Bound	97.72	83.13	61.57	91.89	94.00	82.10	84.37	81.06

Table 4.3: Comparison of phrase localization performance over phrase types. Upper Bound refers to the proportion of phrases of each type for which there exists a region proposal having at least 0.5 IOU with the ground truth.

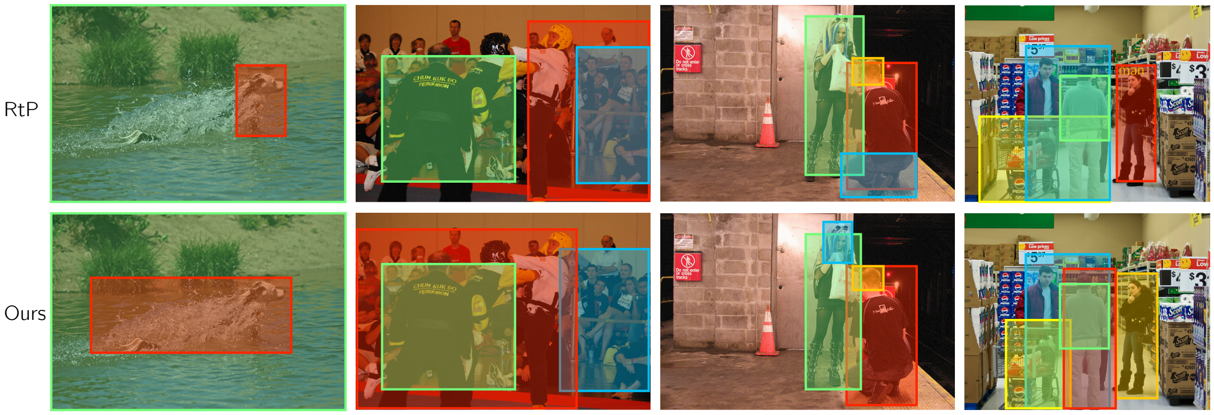
Method	Single Phrase Cues (SPC)				Phrase-Pair Cues (PPC)					
	Object Detectors	Adjectives	Subject-Verb	Verb-Object	Verbs		Prepositions		Clothing & Body Parts	
					Left	Right	Left	Right	Left	Right
Baseline	74.25	57.71	69.68	40.70	78.32	51.05	68.97	55.01	81.01	50.72
+Cue	75.78	64.35	75.53	47.62	78.94	51.33	69.74	56.14	82.86	52.23
#Test	4,059	3,809	3,094	2,398	867	858	780	778	1,464	1,591
#Train	114,748	110,415	94,353	71,336	26,254	25,898	23,973	23,903	42,084	45,496

Table 4.4: Breakdown of performance for individual cues restricted only to test phrases to which they apply. For SPC, Baseline is given by CCA+Position+Size. For PPC, Baseline is the full SPC model. For all comparisons, we use the improved boxes from bounding box regression on top of object detector output. PPC evaluation is split by which side of the relationship the phrases occur on. The bottom two rows show the numbers of affected phrases in the test and training sets. For reference, there are 14.5k visual phrases in the test set and 427k visual phrases in the train set.

and obtains almost 2% better performance. Our full model is 5% better than RtP. It is also worth noting that a rank-SVM model [114] for learning cue combination weights gave us 8% worse performance than the direct search scheme of Section 4.1.2.

Table 4.3 breaks down the comparison by phrase type. Our model has the highest accuracy on most phrase types, with scenes being the most notable exception, for which GroundeR [83] does better. However, GroundeR uses Selective Search proposals [109], which have an upper bound performance that is 7% higher on scene phrases despite using half as many proposals. Although body parts have the lowest localization accuracy at 25.24%, this represents an 8% improvement in accuracy over prior methods. However, only around 62% of body part phrases have a box with high enough IOU with the ground truth, showing a major area of weakness of category-independent proposal methods. Indeed, if we were to augment our EdgeBox region proposals with ground truth boxes, we would get an overall improvement in accuracy of about 9% for the full system.

Since many of the cues apply to a small subset of the phrases, Table 4.4 details the



This dog is jumping through **the water**.

Two people are hitting each other in a karate match, while an **audience** and **referee** watch.

A young man kneeling in front of **a young girl** who has **blond hair** and **blue stripes**.

A man in **a gray sweater** speaks to **two women** and **a man** pushing **a shopping cart** through Walmart.

Figure 4.2: Example results on Flickr30k Entities comparing our SPC+PPC model's output with the RtP model [27]. See text for discussion.

performance of cues over only the phrases they affect. As a baseline, we compare against the combination of cues available for all phrases: region-phrase CCA, position, and size. To have a consistent set of regions, the baseline also uses improved boxes from bounding box regressors trained along with the object detectors. As a result, the object detectors provide less than 2% gain over the baseline for the phrases on which they are used, suggesting that the regression provides the majority of the gain from CCA to CCA+Det in Table 4.2. This also confirms that there is significant room for improvement in selecting candidate regions. By contrast, adjective, subject-verb, and verb-object detectors show significant gains, improving over the baseline by 6-7%.

The right side of Table 4.4 shows the improvement on phrases due to phrase pair cues. Here, we separate the phrases that occur on the left side of the relationship, which corresponds to the subject, from the phrases on the right side. Our results show that the subject, is generally easier to localize. On the other hand, clothing and body parts show up mainly on the right side of relationships and they tend to be small. It is also less likely that such phrases will have good candidate boxes – recall from Table 5.2 that body parts have a performance upper bound of only 62%. Although they affect relatively few test phrases, all three of our relationship classifiers show consistent gains over the SPC model. This is encouraging given that many of the relationships that are used on the validation set to learn our model parameters do not occur in the test set (and vice versa).

Figure 4.2 provides a qualitative comparison of our output with the RtP model from Chapter 3. In the first example, the prediction for the dog is improved due to the subject-verb classifier for *dog jumping*. For the second example, pronominal coreference resolution (Section 4.2.1) links *each other* to *two men*, telling us that not only is a man hitting something, but also that another man is being hit. In the third example, the RtP model is not able to locate the woman's blue stripes in her hair despite having a model for *blue*. Our adjective detectors take into account *stripes* as well as *blue*, allowing us to correctly localize the phrase, even though we still fail to localize the hair. Since the blue stripes and hair should co-locate, a method for obtaining co-referent entities would further improve performance on such cases. In the last example, the RtP model makes the same incorrect prediction for the two men. However, our spatial relationship between the first man and his gray sweater helps us correctly localize him. We also improve our prediction for the shopping cart.

This chapter has combined a collection of cues and demonstrated their effectiveness at the phrase localization task, achieving a 5% gain over our approach which was the previous state-of-the-art in Chapter 3. One drawback of our approach is that many of the cues are hand-crafted, which requires some annotation effort to produce the dictionaries used to identify the categories used for our detectors. In the next chapter, we introduce a new model which decides what concepts to learn along with how to associate the image and text features in a single network.

CHAPTER 5: CONDITIONAL IMAGE-TEXT EMBEDDING NETWORKS

In this chapter, we propose a Conditional Image-Text Embedding (CITE) network that jointly learns different embeddings for subsets of phrases (Figure 5.1). This enables our model to train separate embeddings for phrases that share a concept. Each conditional embedding can learn a representation specific to a subset of phrases while also taking advantage of weights that are shared across phrases. This is especially important for smaller groups of phrases that would be prone to overfitting if we were to train separate embeddings for them. In contrast to our approach in Chapter 4 as well as similar approaches that manually determine how to group concepts [89, 126], we use a concept weight branch, trained jointly with the rest of the network, to do a soft assignment of phrases to learned embeddings automatically. The concept weight branch can be thought of producing a unique embedding for each region-phrase pair based on a phrase-specific linear combination of individual conditional embeddings. By training multiple embeddings our model also reduces variance akin to an ensemble of networks, but with far fewer parameters and lower computational cost.

Our idea of conditional embeddings was directly inspired by the conditional similarity networks of Veit *et al*. [126], although that work does not deal with cross-modal data and does not attempt to automatically assign different input items to different similarity subspaces. An earlier precursor of the idea of conditional similarity metrics can be found in [127] which assigned unseen categories to an embedding used to train similar categories in the training data. In contrast, our approach determines which concepts to learn and produces assignments automatically in a single end-to-end model.

We begin Section 5.1 by describing the image-text Similarity Network [41] that we use as our baseline model. Section 5.2 describes our text-conditioned embedding model. Section 5.2.1 discusses three methods of assigning phrases to the trained embeddings. Lastly, Section 5.3 contains detailed experimental results and analysis of our proposed approach.

5.1 IMAGE-TEXT SIMILARITY NETWORK

Given an image and a phrase, our goal is to select the most likely location of the phrase from a set of region proposals. To accomplish this, we build upon the image-text similarity network introduced in Wang *et al*. [41]. At a high level, this approach consists of learning a nonlinear embedding and a metric to compare them with while as the CCA model used in previous chapters creates a linear embedding trained solely on positive image-text pairs. The image and text branches of this network each have two fully connected layers with

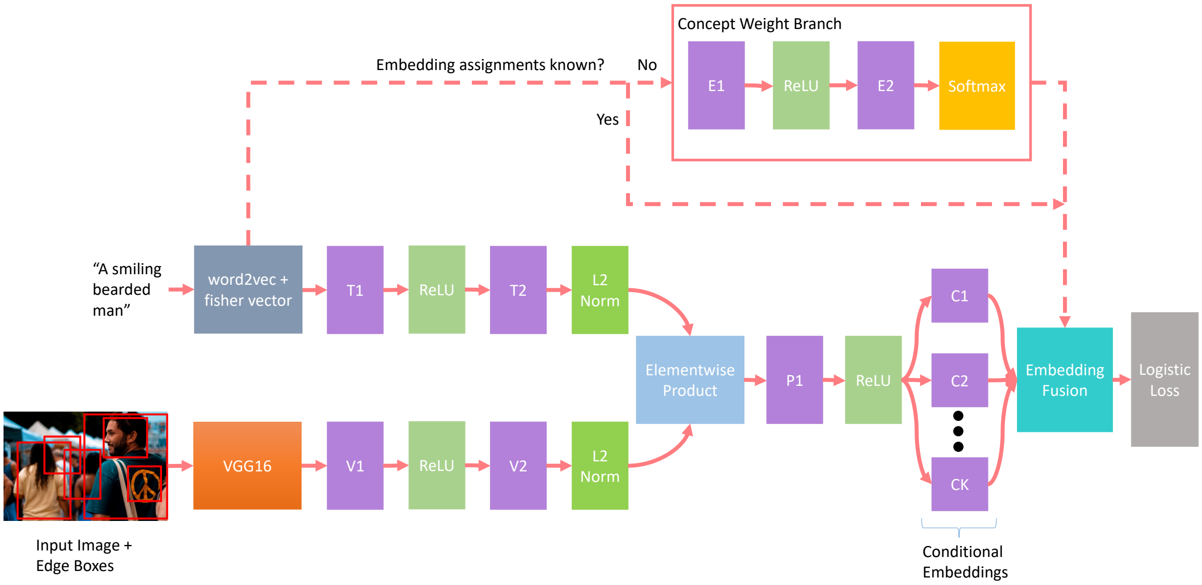


Figure 5.1: Our CITE model separates phrases into different groups and learns conditional embeddings for these groups in a single end-to-end model. Assignments of phrases to embeddings can either be pre-defined (*e.g.* by separating phrases into distinct concepts like *people* or *clothing*), or can be jointly learned with the embeddings using the concept weight branch. Similarly colored blocks refer to layers of the same type, with purple blocks representing fully connected layers. Best viewed in color.

batch normalization [128] and ReLUs. The final outputs of these branches are L2 normalized before performing an element-wise product between the image and text representations. This representation is then fed into a triplet of fully connected layers using batch normalization and ReLUs. This is analogous to using the CITE model in Figure 5.1 with a single conditional embedding.

The training objective for this network is a logistic regression loss computed over phrases P, the image regions R, and labels Y. The label y_{ij} for the ith input phrase and jth region is $+1$ where they match and -1 otherwise. Since this is a supervised learning approach, matching pairs of phrases and regions need to be provided in the annotations of each dataset. After producing some score x_{ij} measuring the affinity between the image region and text features using our network, the loss is given by

$$L_{sim}(P, R, Y) = \sum_{ij} \log(1 + \exp(-y_{ij} x_{ij})). \tag{5.1}$$

In this formulation, it is easy to consider multiple regions for a given phrase as positive examples and to use a variable number of region proposals per image. This is in contrast to competing methods which score regions with softmax with a cross entropy loss over a set number of region proposals per image (*e.g.* [82, 83, 91]).

Sampling phrase-region training pairs. Following Wang *et al.* [41], we consider any regions with at least 0.6 intersection over union (IOU) with the ground truth box for a given phrase as a positive example. Negative examples are randomly sampled from regions of the same image with less than 0.3 IOU with the ground truth box. We select twice the number of negative regions as we have positive regions for a phrase. If too few negative regions occur for an image-phrase pair, then the negative example threshold is raised to 0.4 IOU.

Input Features. We represent phrases using the HGLMM fisher vector encoding [8] of word2vec [120] PCA reduced down to 6,000 dimensions. We generate region proposals using Edge Boxes [108]. Similarly to most state-of-the-art methods on our target datasets, we represent image regions using a Fast RCNN network [50] fine-tuned on the union of PASCAL 2007 and 2012 trainval sets [106]. The only exception is the experiment reported in Table 5.1(d), where we fine-tune the Fast RCNN parameters (corresponding to the VGG16 box in Figure 5.1) on the Flickr30K Entities dataset.

Spatial location. Following [83, 91, 92, 93], we experiment with concatenating bounding box location features to our region representation. This way our model can learn to bias predictions for phrases based on their location (*e.g.* that *sky* typically occurs in the top part of an image). For Flickr30K Entities we encode this spatial information as defined in [91, 92] for this dataset. For an image of height H and width W and a box with height h and width w is encoded as $[x_{min}/W, y_{min}/H, x_{max}/W, y_{max}/H, wh/WH]$. For a fair comparison to prior work [83, 91, 92], experiments on the ReferIt Game dataset encode the spatial information as an 8-dimensional feature vector $[x_{min}, y_{min}, x_{max}, y_{max}, x_{center},\ y_{center}, w, h]$. For Visual Genome we adopt the same method of encoding spatial location as used for the ReferIt Game dataset.

5.2 CONDITIONAL IMAGE-TEXT NETWORK

Inspired by Veit *et al.* [126], we modify the image-text similarity model of the previous section to learn a set of conditional or concept embedding layers denoted $C_1, \ldots C_K$ in Figure 5.1. These are K parallel fully connected layers each with output dimensionality M. The outputs of these layers, in the form of a matrix of size $M \times K$, are fed into the embedding fusion layer, together with a K-dimensional concept weight vector U, which can be produced by several methods, as discussed in Section 5.2.1. The fusion layer simply performs a matrix-vector product, *i.e.*, $F = CU$. This is followed by another fully connected layer representing the final classifier (*i.e.*, the layer's output dimension is 1).

5.2.1 Embedding Assignment

This section describes three possible methods for producing the concept weight vector U for combining the conditional embeddings as introduced in Section 5.2.

Coarse categories. The Flickr30K Entities dataset comes with hand-constructed dictionaries that group phrases into eight coarse categories: *people, clothing, body parts, animals, vehicles, instruments, scene, other.* We use these dictionaries to map phrases to binary concept vectors representing their group membership. This is analogous to the approach of Veit *et al.* [126], which defines the concepts based on meta-data labels. Both the remaining approaches base their assignments on the training data rather than a hand-defined category label.

Nearest cluster center. A simple method of creating concept weights is to perform K-means clustering on the text features of the queries in the test set. Each cluster center becomes its own concept to learn. The concept weights U are then encoded as one-hot cluster membership vectors which we found to work better than alternatives such as similarity of a sample to each cluster center.

Concept weight branch. Creating a predefined set of concepts to learn, either using dictionaries or K-means clustering, produces concepts that don't necessarily have anything to do with the difficulty or ease in localizing the phrases within them. An alternative is to let the model decide which concepts to learn. With this in mind, we feed the raw text features into a separate branch of the network consisting of two fully connected layers with batch normalization and a ReLU between them, followed by a softmax layer to ensure the output sums to 1 (denoted as the concept weight branch in Figure 5.1). The output of the softmax is then used as the concept weights U. This can be seen as analogous to using soft attention [23] on the text features to select concepts for the final representation of a phrase. We use L1 regularization on the output of the last fully connected layer before being fed into the softmax to promote sparsity in our assignments. The training objective for our full CITE model then becomes

$$L_{CITE} = L_{sim}(P, R, Y) + \lambda \|\phi\|_1, \tag{5.2}$$

where ϕ are the inputs to the softmax layer and λ is a parameter controlling the importance of the regularization term. Note that we do not enforce diversity of assignments between different phrases, so it is possible that all phrases attend to a single embedding. However, we do not see this actually occur in practice. We also tried to use entropy minimization rather then L1 regularization for our concept weight branch as well as hard attention instead of soft attention, but found all worked similarly in our experiments.

5.3 EXPERIMENTS

5.3.1 Datasets and Protocols

We evaluate the performance of our phrase-region grounding model on three datasets: Flickr30K Entities [27], ReferIt Game [24], and Visual Genome [17]. The metric we report is the proportion of correctly localized phrases in the test set. Consistent with prior work, a 0.5 IOU between the best-predicted box for a phrase and its ground truth is required for a phrase to be considered successfully localized. Similarly to [41, 29, 92], for phrases associated with multiple bounding boxes, the phrase is represented as the union of its boxes.

Training procedure. We begin training our models using Adam [129]. After every epoch, we evaluate performance on the validation set. If a model hasn't increased performance in 5 epochs, we fine-tune our model using stochastic gradient descent at 1/10th the learning rate using the same stopping criteria. We report test set performance for the model that performed best on the validation set.

Comparative evaluation. In addition to comparing to previously published numbers of state-of-the-art approaches on each dataset, we systematically evaluate the following baselines and variants of our model:

- **Similarity Network.** Our first baseline is given by our own implementation of the model from Wang *et al.* [41], trained using the procedure described above. Phrases are pre-processed using stop word removal rather than part-of-speech filtering as done in the original paper. This change, together with a more careful tuning of the training settings, leads to a 2.5% improvement in performance over the reported results in [41]. The model is further enhanced by using the spatial location features (Section 5.1), resulting in a total improvement of 3.5%.

- **Individual Coarse Category Similarity Networks.** We train multiple Similarity Networks on different subsets of the data created according to the coarse category assignments as described in Section 5.2.1.

- **Individual K-means Similarity Networks.** We train multiple Similarity Networks on different subsets of the data created according to the nearest cluster center assignments as described in Section 5.2.1.

- **CITE, Coarse Categories.** No concept weight branch. Phrases are assigned according to their coarse category.
- **CITE, Random.** No concept weight branch. Phrases are randomly assigned to an embedding. At test time, phrases that were assigned to an embedding during training use the same assignments, while new phrases are randomly assigned.
- **CITE, K-means.** No concept weight branch. Phrases are matched to embeddings using nearest cluster center assignments.
- **CITE, Learned.** Our full model with the concept weight branch used to automatically produce concept weights as described in Section 5.2.1.

5.3.2 Flickr30K Entities

We use the same splits as in previous chapters, consisting of 29,783 images for training and 1,000 images each for testing and validation. Models are trained with a batch size of 200 (128 if necessary to fit into GPU memory) and learning rate of 5e-5. We set λ = 5e-5 in Eq. (5.2). We use the top 200 Edge Box proposals per image and embedding dimension $M = 256$ unless stated otherwise.

Grounding Results. Table 5.1 compares overall localization accuracies for a number of methods which make predictions based on a single phrase[1]. The numbers for our Similarity Network baseline are reported in Table 5.1(b), and as stated above, they are better than the published numbers from [41]. Table 5.1(c) reports results for variants of conditional embedding models. From the first two lines, we can see that learning embeddings from subsets of the data without any shared weights leads to only a small improvement ($\leq 1\%$) over the Similarity Network baseline. The third line of Table 5.1(c) reports that separating phrases by manually defined high-level concepts only leads to a 1% improvement even when weights are shared across embeddings. This is likely due, in part, to the significant imbalance between different coarse categories, as a uniform random assignment shown in the fourth line of Table 5.1(c) lead to a 3% improvement. The fifth line of Table 5.1(c) demonstrates that grouping phrases based on their text features better reflects the needs of the data, resulting in just over 3% improvement over the baseline, only slightly better than random assignments. An additional improvement is reported in the eighth line of Table 5.1(c) by incorporating our

[1] Performance on this task can be further improved by taking into account the predictions made for other phrases in the same sentence [29, 90, 91, 92], with the best result using Pascal-tuned features of 57.53% achieved by Chen *et al.* [91] and 65.14% using Flickr30K-tuned features [92].

	Method	Accuracy
(a)	**Single Phrase Methods (PASCAL-tuned Features)**	
	NonlinearSP [110]	43.89
	GroundeR [83]	47.81
	MCB [82]	48.69
	RtP [27]	50.89
	Similarity Network [41]	51.05
	IGOP [94]	53.97
	SPC [29]	55.49
	MCB + Reg + Spatial [91]	51.01
	MNN + Reg + Spatial [91]	55.99
(b)	**Our Implementation**	
	Similarity Network	53.45
	Similarity Network + Spatial	54.52
(c)	**Conditional Models + Spatial**	
	Individual Coarse Category Similarity Networks, $K = 8$	55.32
	Individual K-means Similarity Networks, $K = 8$	54.95
	CITE, Coarse Categories, $K = 8$	55.42
	CITE, Random, $K = 16$	57.58
	CITE, K-means, $K = 16$	57.89
	CITE, Learned, $K = 4$	58.69
	CITE, Learned, $K = 4$, 500 Edge Boxes	59.27
(d)	**Flickr30K-tuned Features + Spatial**	
	PGN + QRN [92]	60.21
	CITE, Learned, $K = 4$, 500 Edge Boxes	**61.89**

Table 5.1: Phrase localization performance on the Flickr30k Entities test set. (a) State-of-the-art results when predicting a single phrase at a time taken from published works. (b,c) Our baselines and variants using PASCAL-tuned features. (d) Results using Flickr30k-tuned features.

concept weight branch, enabling our model to both determine what concepts are important to learn and how to assign phrases to them. We see in the last line of Table 5.1(c) that going from 200 to 500 bounding box proposals provides a small boost in localization accuracy. This results in our best performance using PASCAL-tuned features which is 3% better than the prior work reported in Table 5.1(a) and 4.5% better than the Similarity Network. We also note that the time to test an image-phrase pair is almost unaffected using our approach (the CITE, Learned, K=4 model performs inference on 200 Edge Boxes at 0.182 seconds per pair using a NVIDIA Titan X GPU with our implementation) compared with the baseline Similarity Network (0.171 seconds per pair). Finally, Table 5.1(d) gives results for models whose visual features were fine-tuned for localization on the Flickr30K Entities dataset. Our model still obtains a 1.5% improvement over the approach of Chen *et al.* [92], which used

	People	Clothing	Body Parts	Animals	Vehicles	Instruments	Scene	Other
PASCAL-tuned Features								
GroundeR [83]	61.00	38.12	10.33	62.55	68.75	36.42	58.18	29.08
RtP [27]	64.73	46.88	17.21	65.83	68.75	37.65	51.39	31.77
IGOP [94]	68.71	**56.83**	19.50	70.07	73.75	39.50	60.38	32.45
MCB + Reg + Spatial [91]	62.75	43.67	14.91	65.44	65.25	24.74	**64.10**	34.62
MNN + Reg + Spatial [91]	67.38	47.57	20.11	73.75	72.44	29.34	63.68	37.88
CITE, Learned, $K = 4$ + Spatial	**73.20**	52.34	**30.59**	**76.25**	**75.75**	**48.15**	55.64	**42.83**
Flickr30K-tuned Features								
PGN + QRN + Spatial [92]	75.05	55.90	20.27	73.36	68.95	45.68	**65.27**	38.80
CITE, Learned, $K = 4$ + Spatial	**75.95**	**58.50**	**30.78**	**77.03**	**79.25**	**48.15**	58.78	**43.24**

Table 5.2: Comparison of phrase grounding performance over coarse categories on the Flickr30K Entities dataset. Our models were tested with 500 Edge Box proposals.

bounding box regression as well as a region proposal network. In principle, we could also incorporate these techniques to further improve the model.

Table 5.2 breaks down localization accuracy by coarse category. Of particular note are our results on the challenging *body part* category, which are typically small and represent only 3.5% of the phrases in the test set, improving over the next best model as well as the Similarity Network trained on just body part phrases by 10% when using Flickr30K-tuned features. We also see a substantial improvement in the *vehicles* and *other* categories, seeing a 5-9% improvement over the previous state-of-the-art. The only category where we perform worse are phrases referring to scenes, which commonly cover the majority (or entire) image. Here, incorporating a bias towards selecting larger proposals, as in [27, 29], can lead to significant improvements.

Parameter Selection. In addition to reporting the localization performance, we also provide some insight into the effect of different parameter choices and what information our model is capturing. In Figure 5.2 we show how the number K of learned embeddings affects performance. Using our concept weight branch consistently outperforms K-means cluster assignments. Table 5.3 shows how the embedding dimensionality M affects performance. Here we see that reducing the output dimension from 256 to 64 (*i.e.*, by 1/4th) leads to a minor (1%) decrease in performance. This result is particularly noteworthy as the CITE network with $K = 4, M = 64$ has 4 million parameters compared the 14 million the baseline Similarity Network has with $M = 256$ while still maintaining a 3% improvement in performance. We also explore the effect the number of Edge Boxes has on performance in Table 5.4. In contrast to some prior work which performed best using 200 candidates (*e.g.* [27, 29]), our

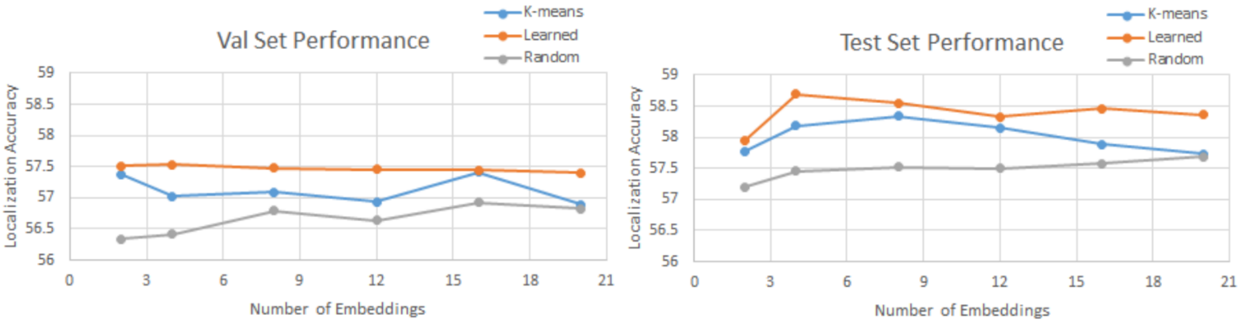


Figure 5.2: Effect of the number of learned embeddings (K) on Flickr30K Entities localization accuracy using PASCAL-tuned features.

Embedding Size (M)	64	128	256	512
Validation Set Accuracy	56.32	57.51	**57.53**	57.42
Test Set Accuracy	57.77	58.48	**58.69**	58.64

Table 5.3: Localization accuracy with different embedding sizes using the CITE, Learned, $K = 4$ model on Flickr30K Entities with PASCAL-tuned features. Embedding size refers to M, the output dimensionality of layers P1 and the conditional embeddings in Figure 5.1. The remaining fully connected layers' output dimensions (excluding those that are part of the VGG16 network) are four times the embedding size.

model's increased discriminate power enables us to still be able to obtain a benefit from using up to 500 proposals.

Concept Weight Branch Examination. To analyze what our model is learning, Figure 5.3 shows the means and standard deviations of the weights over the different embeddings broken down by coarse categories. Interestingly, *people* are assigned to one of two embeddings, split by whether the phrase refers to a single person or multiple people. Table 5.5 lists the ten phrases with the highest weight for each embedding to provide insight into what is important to each embedding. While most phrases give the first embedding little weight, it provides the most benefit for finding very specific references to people rather than generic terms (*e.g. little curly hair girl* instead of *girl* itself). These patterns generally hold through multiple runs of the model, indicating they are important concepts to learn for the task.

Qualitative Results. Figure 5.4 gives a look into areas where our model could be improved. Of the phrases that occur at least 100 times in the test set, the lowest performing phrases are *street* and *people* at (resp.) 60% and 64% accuracy. The highest performing of these common phrases is *man* at 81% accuracy, which also happens to be the most common phrase with 1065 instances in the test set. In the top-left example of Figure 5.4, the word *people*, which is not correctly localized, refers to partially visible background pedestrians. Analyzing the saliency of a phrase in the context of the whole caption may lead to treating these phrases

#Edge Box Proposals	100	200	500	1000
Validation Set Accuracy	49.61	57.53	58.48	57.87
Test Set Accuracy	51.32	58.69	59.27	58.63

Table 5.4: Localization accuracy with different numbers of proposals using the CITE, Learned, $K = 4$ model on Flickr30K Entities with PASCAL-tuned features.

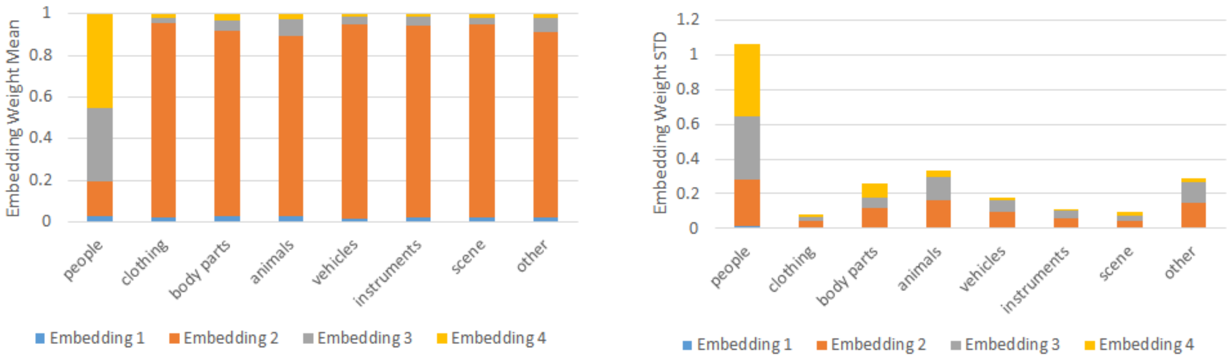


Figure 5.3: The mean weight for each embedding (left) along with the standard deviation of those weights (right) broken down by coarse category for the Flickr30K Entities dataset using Flickr30K-tuned features.

differently. Global inference constraints, for example, a requirement that predictions for *a man* and *a woman* must be different, would be useful for the top-center example. Performing pronoun resolution, as attempted in [29], would help in the top-right example. In the test set, the pronoun *one* is correctly localized around 36% of the time, whereas *the blond woman* is correctly localized 81% of the time. Having an understanding of relationships between entities may help in cases such as the bottom-left example of Figure 5.4, where the extent of the table could be refined by knowing that the groceries are "on" it. Our model also performs relatively poorly on phrases referring to classic "stuff" categories, as shown in the bottom-center and bottom-right examples. The *water* and *street* phrases in these examples are only partly localized. Using pixel-level predictions may help to recover the full extent of these types of phrases since the parts of the images they refer to are relatively homogeneous.

5.3.3 ReferIt Game

We use the same splits as Hu *et al.* [88], which consist of 10,000 images combined for training and validation with the remaining 10,000 images for testing. Models are trained with a batch size of 128, learning rate of 5e-4, and λ = 5e-4 in Eq. (5.2). We generate 500 Edge Box proposals per image.

Results. Table 5.6 reports the localization accuracy across the ReferIt Game test set. The

Embedding 1	soldiers (0.08), male nun (0.07), rather angry looking woman (0.07), skinny dark complected boy (0.07), little curly hair girl (0.07), middle eastern woman (0.07), first man's leg (0.07), statue athletic man (0.07), referee (0.07), woman drink wine (0.07)
Embedding 2	red scooter (0.97), blue clothes (0.97), yellow bike (0.97), red bike (0.97), red buckets (0.97), yellow backpack (0.97), street window shops (0.97), red blue buckets (0.97), red backpack (0.97), purple red backpack (0.97)
Embedding 3	two people (0.94), two men (0.93), two young kids (0.93), two kids (0.93), two white-haired women (0.93), two women (0.93), group three boys (0.93), two young people (0.93), three people (0.92), crowd people (0.92)
Embedding 4	blond-haired woman (0.91), dark-skinned woman (0.91), gray-haired man (0.91), one-armed man (0.91), dark-haired man (0.91), red-haired man (0.91), boy young man (0.91), man (0.91), well-dressed man (0.91), dark-skinned man (0.91)

Table 5.5: The ten phrases with the highest weight per embedding on the Flickr30K Entities dataset using Flickr30K-tuned features.

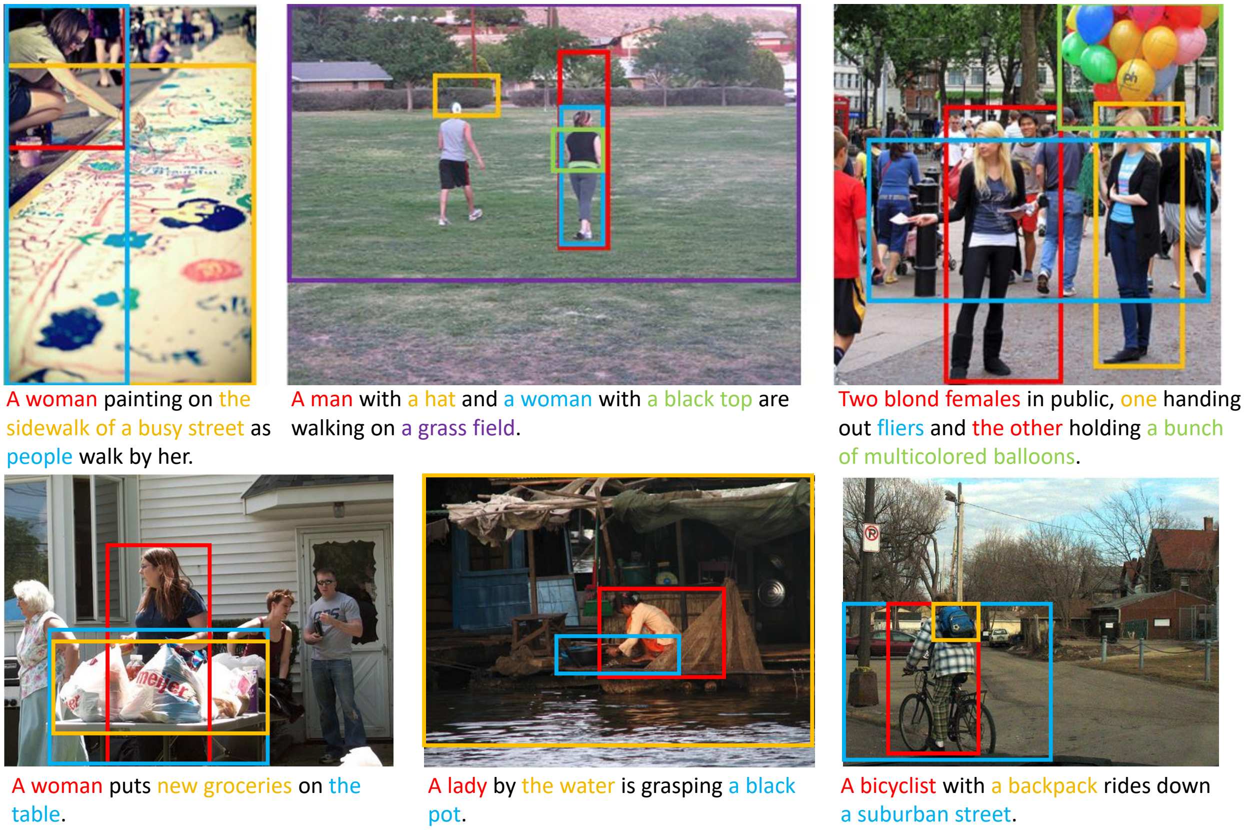
first line of Table 5.6(b) shows that our model using the nearest cluster center assignments results in a 2.5% improvement over the baseline Similarity Network. Using our concept weight branch in order to learn assignments yields an additional small improvement.

We note that we do not outperform the approach of Yeh *et al.* [94] on this dataset. This can likely be attributed to the failures of Edge Boxes to produce adequate proposals on the ReferIt Game dataset. Oracle performance using the top 500 proposals is 93% on Flickr30K Entities, while it is only 86% on this dataset. As a result, the specialized bounding box methods used by Yeh *et al.* as well as Chen *et al.* [91] may play a larger role here. Our model would also likely benefit from these improved bounding boxes.

As with the Flickr30K Entities dataset, we show the effect of the number K of embeddings on localization performance in Figure 5.5. While the concept weight branch provides a small performance improvement across many different choices of K, when $K = 2$ the clustering assignments actually perform a little better. However, this behavior is atypical in our experiments across all three datasets, and may simply be due to its size since ReferIt Game has far fewer ground truth phrase-region pairs in the training set.

5.3.4 Visual Genome

We use the same splits as Zhang *et al.* [84], consisting of 77,398 images for training and 5,000 each for testing and validation. Models are trained with a learning rate of 5e-5, and $\lambda =$ 5e-4 in Eq. (5.2). We generate 500 Edge Box proposals per image, and use a batch size



A woman painting on the sidewalk of a busy street as people walk by her.

A man with a hat and a woman with a black top are walking on a grass field.

Two blond females in public, one handing out fliers and the other holding a bunch of multicolored balloons.

A woman puts new groceries on the table.

A lady by the water is grasping a black pot.

A bicyclist with a backpack rides down a suburban street.

Figure 5.4: Examples demonstrating some common failure cases on the Flickr30K Entities dataset. See Section 5.3.2 for discussion.

of 128.

Results. Table 5.7 reports the localization accuracy across the Visual Genome dataset. Table 5.7(a) lists published numbers from several recent methods. The current state of the art performance belongs to Zhang *et al.* [84], who fine-tuned visual features on this dataset and created a cleaner set during training by pruning ambiguous phrases. We did not perform either fine-tuning or phrase pruning, so the most comparable reference number for our methods is their 17.5% accuracy without these steps.

The baseline accuracies for our Similarity Network with and without spatial features are given in the last two lines of Table 5.7(a). We can see that including the spatial features gives only a small improvement. This is likely due to the denser annotations in this dataset as compared to Flickr30K Entities. For example, a phrase like *a man* in Flickr30K Entities would typically refer to a relatively large region towards the center since background instances are commonly not mentioned in an image-level caption. However, entities in Visual Genome include both foreground and background instances.

In the first line of Table 5.7(b), we see that our K-means model is 3.5% better than the

	Method	Accuracy
(a)	**State-of-the-art**	
	SCRC [88]	17.93
	GroundeR + Spatial [83]	26.93
	MCB + Reg + Spatial [91]	26.54
	CGRE [130]	31.85
	MNN + Reg + Spatial [91]	32.21
	IGOP [94]	34.70
	Similarity Network + Spatial	31.26
(b)	**Conditional Models + Spatial**	
	CITE, K-Means, $K = 2$	34.01
	CITE, Learned, $K = 12$	34.13

Table 5.6: Localization performance on the ReferIt Game test set. (a) Published results and our Similarity Network baseline. (b) Our best-performing conditional models.

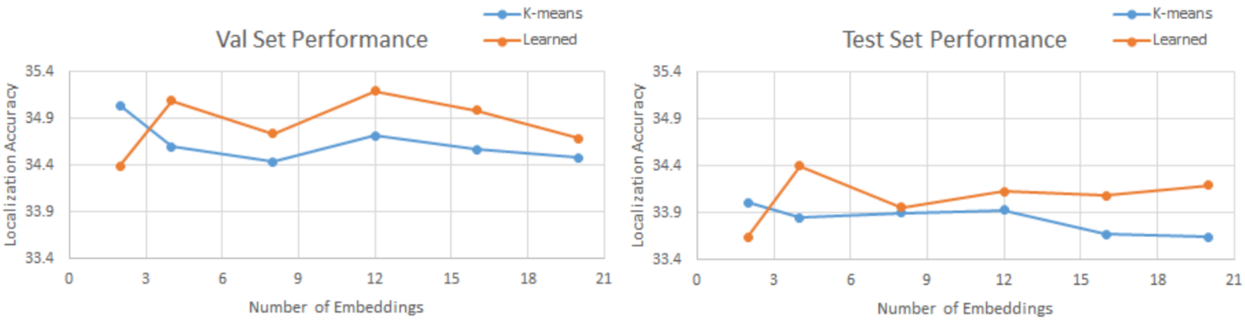


Figure 5.5: Effect of the number K of embeddings on localization accuracy on the ReferIt Game dataset.

Similarity Network baseline, and over 6% better than the 17.5% accuracy of [84]. According to the second line of Table 5.7(b), using the concept weight branch obtains a further improvement. In fact, our full model with pre-trained PASCAL features has better performance than Zhang *et al*. [84] with fine-tuned features.

As with the other two datasets, Figure 5.6 reports performance as a function of the number of learned embeddings. Echoing most of the earlier results, we see a consistent improvement for the learned embeddings over the K-means ones. The large size of this dataset (> 250,000 instances in the test set) helps to reinforce the significance of our results.

5.4 DISCUSSION AND FUTURE DIRECTIONS

This chapter introduced a method of learning a set of conditional embeddings and phrase-to-embedding assignments in a single end-to-end network. The effectiveness of our approach

	Method	Accuracy
(a)	**State-of-the-art**	
	Densecap [66]	10.1
	SCRC [88]	11.0
	DBNet [84]	17.5
	DBNet (with APP) [84]	21.2
	DBNet (with APP, V. Genome-tuned Features) [84]	23.7
	Similarity Network	19.76
	Similarity Network + Spatial	20.08
(b)	**Conditional Models + Spatial**	
	CITE, K-Means, $K = 12$	23.67
	CITE, Learned, $K = 12$	24.43

Table 5.7: Phrase localization performance on Visual Genome. (a) Published results and our Similarity Network baselines. APP refers to ambiguous phrase pruning (see [84] for details). (b) Our best-performing conditional models.

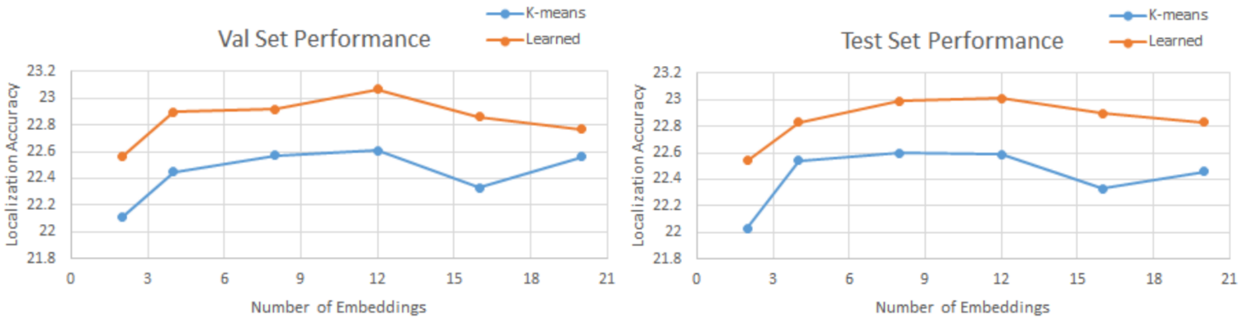


Figure 5.6: Effect of the number of learned embeddings on performance on the Visual Genome with models trained on 1/3 of the available training data.

was demonstrated on three popular and challenging phrase-to-region grounding datasets. In future work, our model could be further improved by including a term to enforce that distinct concepts are being learned by each embedding. Whereas we only considered predefining concepts using k-means and the coarse categories which group by the type of object, our work in Chapter 4 used part of speech tags to identify concepts to learn. A direct application of the approach of Chapter 4 is shown in Figure 5.7, where a Similarity Network is trained for adjectives (Adj), subject-verb (SV), and verb-object (VO) cues. Unlike Chapter 4, however, there is no restriction on the adjectives or verbs used, and each additional cue is trained in sequence (*i.e.* first the phrase model is trained, then those weights are frozen and the adjective model is trained). The HGLMM representation for the entire phrase is fed into the adjective cues (*i.e.* nouns and adjectives), while the phrase and the verb are used to compute the text features for the verb-based cues. Since each network is trained by adding

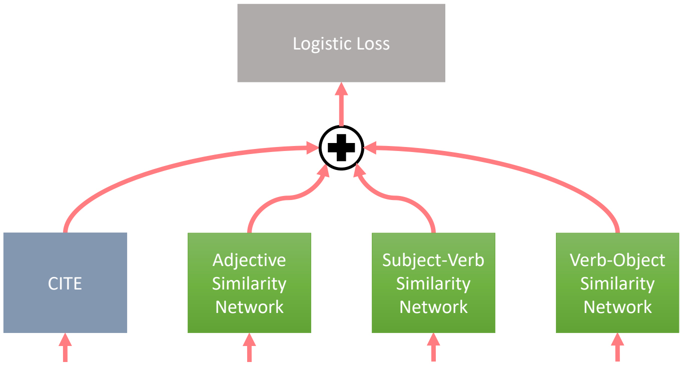

Logistic Loss
CITE
Adjective Similarity Network
Subject-Verb Similarity Network
Verb-Object Similarity Network

Figure 5.7: Approach which combines our CITE model with cue-specific Similarity Networks. Each network has its own VGG16 CNN providing visual features which we fine-tune.

Method	Accuracy
PGN + QRN [92]	60.21
CITE, Learned, $K = 4$, 500 Edge Boxes	61.89
CITE, Learned, $K = 4$, 500 Edge Boxes + Adj	62.12
CITE, Learned, $K = 4$, 500 Edge Boxes + Adj + SV	62.75
CITE, Learned, $K = 4$, 500 Edge Boxes + Adj + SV + VO	**63.25**

Table 5.8: Phrase localization performance reporting the effect of performance Similarity Networks trained for part-of-speech cues like those used in Chapter 4 have on the Flickr30k Entities test set using Flickr30k-tuned features.

its score to the output of the previously trained models, no weights are learned to combine the cue-specific scores. As reported in Table 5.8, this produces a 1.5% improvement to localization accuracy. However, since each additional cue also comes with its own fine-tuned convolutional neural network (CNN) image encoder, it adds a considerable amount of model complexity. A promising direction may be to create a hybrid model between our CITE network and the part-of-speech cues, where some of the conditional embeddings in Figure 5.1 are assigned to the part-of-speech cues and others are learned with our concept weight branch.

Our approach could be further improved by incorporating a number of orthogonal techniques used in competing work. By jointly predicting multiple phrases in an image as also done in Chapter 4 our model could take advantage of relationships between multiple entities (additional examples include [90, 91, 92]). Including bounding box regression and a region proposal network as done in [91, 92] would also likely lead to a better model. In fact, tying the regression parameters to a specific concept embedding may further improve performance since it would simplify our prediction task as a result of needing to learn parameters for just the phrases assigned to that embedding. However, these directions are focused specifically on the localization task, which is evaluated only on its ability to find a ground truth phrase in an image. In Chapter 6, we address a more general version of this task which requires a model to both localize a phrase and identify if it exists in an image.

CHAPTER 6: PHRASE DETECTION

Object detection has made remarkable progress in recent years with models like Faster RCNN [53], the Single Shot Detector [52], and YOLO [51, 131] improving the speed and accuracy of this task. However, these approaches limit their recognition ability to a list of predefined objects and are unable to distinguish between instances of the same object. Phrase grounding solves this by learning to localize any object described by a natural language query. In this task entities like *boy* and *girl* are considered distinct even though they are both *people*. The fact that both phrases can be referred to by another word also points to another issue: annotation sparsity. Annotating every way in which an entity may be referenced as a phrase is considered too costly, and therefore only a short list of variations is provided. For phrase localization this is mitigated since we are given ground truth image-text pairs. However, in phrase detection we localize every phrase in every image. The ramification of this is a very high false positive and false negative rates for most entities. As a result, researchers attempting a more detection style task have only evaluated their approach on a very limited number of queries [132] or only considered a limited number of randomly selected images as negatives for each phrase [84].

In this chapter, we are trying to localize entities which may or may not be in the image, without any restrictions on queries or negative regions. To solve this task we split the problem into two parts: one module which identifies if a query is present in an image and another module which localizes it (see Figure 6.1 for an overview). In this way we can take advantage of prior work which accomplishes these two subtasks separately and combine them to solve our problem. We explore several different alternatives for both components. For our localization module we compare Conditional Image-Text Embedding (CITE) model introduced in Chapter 5 with the CCA baseline we used in Chapter 3 and a Fast RCNN detector [50] trained to detect common words. Our phrase identification module relates images to sentences and then localizes the phrases within them. By linking our phrases to sentences we can take advantage of the structured knowledge present in this representation to help reduce our false positives. For example, if we were to look for a hand then we should also likely be looking for a person, or if we see a beach then we may also look for surfboards or the ocean. We evaluate performance using both retrieval-based approaches (*i.e.* for an image at test time, sentences are selected from the training set) as well as caption generation.

In addition to investigating the effect different components may have on the detection task, we also introduce small modifications which can help the Embedding Network identify related sentences more accurately. First, we explore multi-scale representations of images

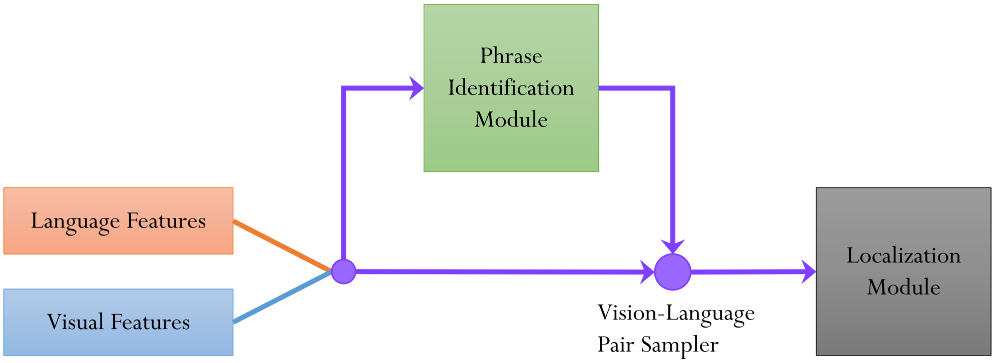


Figure 6.1: Given the feature encodings of the set of phrases we want to detect and their corresponding images and their regions, we first identify which phrases are likely to exist in each image using our phrase identification module. Unlikely phrases for each image are filtered out, and the remaining image-phrase pairs are provided as input to the localization module which makes the final predictions.

for our phrase identification module that has proven to be effective for object detectors (like YOLO and SSD) and have had little study in the image-sentence retrieval setting using convolutional neural networks. After all, identifying smaller objects can prove difficult when they cover only a few pixels in the input image. Also, in prior work two nearly identical sentences from different images would be considered a negative pair when training an embedding using a triplet loss (as used in [41]). By taking into account the similarity between sentences measured by the number of words they share our model implicitly learns an ordering to its embedding, leading to better retrieval performance. We demonstrate the usefulness of our approach not only for phrase detection on the Flickr30K Entities dataset, but on the underlying task of bidirectional image-sentence retrieval as well.

6.1 APPROACH

Given a set of phrases and a database of images, our task is to identify which images contain the query phrases and localize them. We split this task into its constituent parts, phrase identification and localization, and train separate models to perform both tasks. Our phrase identification module takes the image as input and produces the K most likely captions. We compare the following methods for this module:

- Embedding Network [41]: We modify the two branch embedding network of Wang *et al*. which we use to retrieve the K most likely sentences for a given image from the training set at test time. We review this model and discuss our modifications in Section 6.1.1

- Show and Tell [133]: We generate the K best captions for a given image using the standard image encoder-LSTM decoder paradigm.

After we retrieve our set of likely captions, we filter out every phrase that does not have any of its words present in the set of likely captions and feed the remainder into our localization module (*i.e.* we keep a phrase even if only one of its words was within the captions). We compare the following alternatives for our localization module:

- Fast RCNN phrase detectors [50]: We train a detector for every word which occurs at least 100 times in the training set, resulting in 953 classes. At test time, we average the scores of each word which exists in the input phrase. If no detector is trained for any word in a phrase, we use the most similar detector to the phrase's head noun. Similarity is measured by the cosine distance of the HGLMM encoding [8] of the phrases.
- CCA: We train a CCA model between the image and HGLMM text features as we did in Chapter 3, except that we use the visual features of the Fast RCNN phrase detectors instead of the PASCAL-tuned features.
- CITE: We localize phrases using the Flickr30K-tuned model from Chapter 5.

For most of the localization and identification modules we use the same approach as in prior work or previous chapters in this dissertation. The exception is the Embedding Network of Wang *et al.* [41], where we found performance could be significantly improved by using a larger input image resolution as well as introduce a variable margin to the standard triplet loss function. We shall now describe how we define our margin and how it fits into the rest of the Embedding Network.

6.1.1 Variable Margin Embedding Network

We modify the two branch embedding network of Wang *et al.* [41] which uses a pair of fully connected layers followed by an L2 normalization to project the input visual and language features into a shared semantic space. Following our approach in Chapter 3, we encode sentences using the HGLMM fisher encodings and average 10 crops of an image to produce our visual representation. However, rather than use VGG features as we did in Chapter 3, we compute features using a 152 layer Deep Residual Network (ResNet) [134] which was pretrained on ImageNet [105]. Not only does the ResNet provide a more powerful feature representation in a more compact space (2048 dimensions vs. VGG's 4096), but since it is fully-convolutional we can explore the effect of the image input resolution without retraining the model. The only adjustment that is required is to alter the last average pooling layer so that the dimension of the representation for each image remains consistent. Thus, we provide results using both the standard 224 crop inputs as well as 448 crop inputs as well. We shall now discuss the loss function we use to train our embedding model.

Word-based Similarity

A standard way of training embedding models is to encourage a positive image-language pair to be separated by a negative pair by some margin using triplet loss function. While some works have included additional structural constraints when training their embedding (*e.g.* [41, 135]), they still tend to treat two nearly identical language inputs from different images as a negative pair despite their similarities. We use the similarity between language queries, based on the number of shared words between them, to define a margin which is larger for very dissimilar queries and small for similar ones. More formally, let q be the set of words in a query after stopword removal and w be the set of words across all queries associated with a visual input. Our similarity between the pair of queries and the ground truth is defined as:

$$s(q, w) = \frac{q \cap w}{K}, \tag{6.1}$$

where K is the maximum number of shared words any query in the training set has with w. Then, for a triplet of queries (w, q_p, q_n) where q_p is the positive query for the visual input associated with w and q_n the negative, then we define our word-based margin n as:

$$n(w, q_p, q_n) = \max[0, h * (s(w, q_p) - s(w, q_n))], \tag{6.2}$$

where h is a scalar parameter. This is used to adjust our margin in our triplet loss between some vision input v and its positive and negative queries (q_p, q_n) using some distance function d:

$$L_{WordSim}(v, w, q_p, q_n) = \max[0, m + n(w, q_p, q_n) + d(v, q_p) - d(v, q_n)], \tag{6.3}$$

where m is a scalar parameter representing a minimum margin between positive and negative samples. In our experiments we set m in Eq. (6.3) to the same value as h in Eq. (6.2) and used euclidean distance to measure similarity between projected features.

Structure-Preserving Constraints

In addition to encouraging the vision-language inputs to embed into a similar space, Wang *et al.* [41] also included additional constraints which also considered the performance when comparing the vision-vision and language-language inputs. This helped produce a more structured representation which generalized better to new samples. Thus, we take advantage of these same constraints in our work. Each constraint is modeled after a triplet loss function $L_T(anchor, positive, negative)$ pairs of positive and negative samples in each modality. The total loss function for our classification module using an analogous naming convention as Eq. (6.3) is:

$$L_{cls} = L_w(v, q_p, q_n) + \lambda_1 L_T(q, v_p, v_n) + \lambda_2 L_T(q, q_p, q_n) + \lambda_3 L_T(v, v_p, v_n), \tag{6.4}$$

where λ_{1-3} are scalar parameters. We set all our parameters (*e.g.* embedding size, learning rate, etc) following the values dictated by Wang *et al.* [41] as we found they still provided best performance in our experiments.

6.2 EXPERIMENTS

We perform our experiments using the Flickr30K Entities dataset using the same splits as in Chapter 3. First we evaluate the performance of our individual modules at their specialized tasks in Section 6.2.1. Then we report our results on the salient phrase detection task in Section 6.2.3. Finally, we show our the improvements to the Embedding Network used for the phrase identification module affects performance on the task of bidirectional image-sentence retrieval in Section 6.2.2.

6.2.1 Module Evaluation

We begin by evaluating our two modules in isolation on their subtasks. For our localization module we use the same experimental setup and splits for the Flickr30K Entities dataset in Chapter 5. Here we assume we are given ground truth image-phrase pairs and performance is evaluated on how accurately we localize the phrase within the image. Table 6.1 reports the performance of the three types of localization modules we use in our experiments. We see in the first line of Table 6.1 that using the improved visual features provides a significant boost in localization accuracy, improving by 8% over the PASCAL-tuned features used in Chapter 4, but still performs significantly worse on this task than the CITE approach from Chapter 5. In the second line of Table 6.1 we see that directly using

Method	Accuracy
CCA	49.22
Fast RCNN [50]	21.95
CITE	61.89

Table 6.1: Phrase localization performance of the three localization module variants. This is the same localization task and experimental setup used in previous chapters, with CITE being taken from our best model in Table 5.1(d). CCA performance reported here uses Flickr30K-tuned features resulting in a 6% improvement over the CCA model trained with PASCAL-tuned features from Table 4.2.

object detection methods does not translate well to this task, with our Fast RCNN word detectors obtaining less than half the performance of the CCA baseline.

Phrase Identification Experiments

Since we decide to detect a phrase if even one of its words is present in a predicted sentence, our goal is to provide a set of sentences which have the highest number of shared words with the ground truth phrases, while producing as few false positives as possible, *i.e.* we would like to balance recall and precision. Thus, we measure performance using its F1-score within the top K retrieved sentences treating the predicted sentences and ground truth phrases for a single image as sets of words, removing stopwords and any words that don't exist in any ground truth phrases from consideration. We decide on the value of K based on validation performance.

Results. We report performance on the phrase identification task in Table 6.2. Comparing the first three lines of the table we can see that our word similarity (WordSim) based margin as well as increasing the input crop size can provide a significant performance boost on this task. The fourth line of Table 6.2 reports performance using the CNN encoder- LSTM decoder approach to caption generation. This approach performs relatively poorly compared with our retrieval-based module, reporting a F1-score 18 points lower than the retrieval approach. This is likely due, in part, to the limited variation in the generated captions. We expect more recent methods which take into account the diversity in the caption generation results would perform better (*e.g.* [136]).

6.2.2 Bidirectional Retrieval Experiments

In addition to our experiments on phrase identification, we also demonstrate how our modifications to the Embedding Network of Wang *et al.* [41] is useful to the task of bidi-

Method	F1-Score	#Sentences
Embedding Network-224	22.6	5
Embedding Network-488	23.7	5
Embedding Network-488+WordSim	24.1	5
Show and Tell [133]	6.3	100

Table 6.2: Phrase identification experiments comparing retrieving sentences from the training set using our modified Embedding Network with an CNN encoder-LSTM decoder caption generation approach. The number of sentences are based on the method's best F1-score on the validation set.

	Method	Image Annotation			Image Search			mR
		R@1	R@5	R@10	R@1	R@5	R@10	
(a)	RRF [137]	47.6	77.4	89.3	35.4	68.3	79.9	66.0
	DAN [138]	55.0	81.8	89.0	39.4	69.2	79.1	68.9
	VSE++ [139]	52.9	79.1	87.2	39.6	69.6	79.5	68.0
	SCOISM [140]	55.5	82.0	89.3	41.1	70.5	80.1	69.7
	Embedding Network-224 (VGG) [41]	43.2	71.6	79.8	31.7	61.3	72.4	60.0
(b)	Embedding Network-224	52.1	80.2	88.2	39.8	69.8	79.3	68.2
	Embedding Network-448	57.3	85.1	91.7	43.4	75.4	84.1	72.8
	Embedding Network-448+WordSim	59.8	85.7	92.7	43.7	75.1	84.9	73.7

Table 6.3: Bidirectional image-sentence retrieval results on the Flickr30K test set. (a) contains the current state-of-the-art on this task using ResNet features, except where noted, reported in prior work while (b) shows how the effect the crop size and our variable word similarity margin has on the performance of our implementation of the Embedding Network of Wang *et al.* [41].

rectional image-sentence retrieval. Given an image, the task is to retrieve a sentence and given a sentence the model produces the most likely image. Success is measured by whether the ground truth image or sentence is within the top N results (*i.e.* Recall@N). We use the same train/test/val splits on the Flickr30K dataset as Wang *et al.* [41], which consists of 1K images for validation and testing with the rest left for training.

Results. Our results on the bidirectional retrieval task are reported in Table 6.3. It is noteworthy that a tuned version of our model with our ResNet features perform comparably to the current state-of-the-art. This suggests that as we obtain better feature representations a relatively simple model becomes more attractive. Clearly, the resolution of the input image is a significant factor for this dataset, as we a 4.5% improvement in the average recall in the first line of Table 6.3(b) from using a larger input crop size. Our word similarity defined margin further enhances our performance, increasing the average recall by another 1%.

	#Test Occurrences Per Category	$1-9$	$10-29$	≥ 30	mean/ total
(a)	**Localization-Only**				
	CCA	8.2	15.7	17.5	13.8
	Fast RCNN [53]	1.7	4.0	6.9	4.2
	CITE	3.8	9.8	17.0	10.2
	CITE+CCA	9.6	18.1	23.2	17.0
(b)	**+ Embedding Network-488-WordSim**				
	CCA	8.1	16.4	17.6	14.0
	Fast RCNN [50]	2.1	4.9	7.3	4.8
	CITE	6.1	15.0	19.6	13.6
	CITE+CCA	10.0	19.6	23.5	17.7
(c)	**+ Show and Tell**				
	CCA	6.6	11.6	16.6	11.6
	Fast RCNN [50]	1.6	3.5	6.8	4.0
	CITE	3.7	8.7	16.3	9.6
	CITE+CCA	7.6	13.2	20.7	13.8
	#Categories	4,837	134	53	5,024
	#Total Test Occurrences	7,609	2,168	4,704	14,481

Table 6.4: mAP for phrase detection on the Flickr30K test set.

6.2.3 Phrase Detection Experiments

We evaluate the performance on this task using mean average precision (mAP) as is standard with detection tasks. Since this metric may be unstable for phrases with few instances (*i.e.* getting a mAP of 0 or 100 is more likely to arise for phrases with a single occurrence), we separate the phrases into three main groups based on the number of test instances: $\leq 9, 10-29, \geq 30$. A single prediction per image is made for every query evaluated for it. This helps alleviate the issues with false negatives arising from annotation sparsity, since a full labeling for vision-language datasets is not typically done as it would require every applicable phrase to be annotated for every entity.

Results. We report performance on our detection task in Table 6.4. Notably, we see that our CCA module works the best out of the three individual methods both with and without the phrase identification module, even though it performed worse at the localization task as seen in Table 6.1. Thus, we experimented with an approach where we use the CITE module to propose regions, but score them using CCA. This hybrid between the CITE and CCA modules reporting best performance. Including the retrieval-based Embedding Network produces a small, but consistent gain across all approaches as shown in Table 6.4(b). Following our results on the phrase identification task, using generating a caption performs worse than our retrieval method, and also only hinders performance compared to the localization module as reported in Table 6.4(c). Additional discussion on these results as well as directions for future work can be found in Section 6.3.

6.3 DISCUSSION AND FUTURE DIRECTIONS

One of the noteworthy results is that our CCA module was still able to outperform the CITE network, despite the fact that it did worse on the localization task. This suggests that the CITE module has overfit somewhat to the task of localization and its scores for a given phrase are not comparable across images. This observation is in line with prior work, where models which perform better on the localization task do not necessarily generalize to other, even very related tasks (*e.g.* [27, 141]). This may be due to the nature of the task. Consider the bicyclist in the second row of Figure 5.4. Simply having a person detector would be sufficient for finding him without any other knowledge of why he is considered a bicyclist. Even for images with multiple people, the ones mentioned in a caption are typically large and in the center, making them easier to detect than others in the image. As a result, the CITE module as it is trained may find it beneficial to essentially learn category detectors and fire anytime they see that category, even if the phrase itself isn't applicable. Thus, using another method such as our CCA module to score the regions selected by the CITE module seems a reasonable approach, and additional investigation in this direction seems promising.

In related work, Wang *et al.* [41] provided experiments which showed the Embedding Network (*i.e.* what we used in our phrase identification module) is more accurate on some tasks than the Similarity Network which is used as the basis of the CITE module. However, simply replacing the CITE module with an Embedding Network did not produce any significant performance advantages. This may simply be due to not sampling suitable negatives, as embedding style networks have shown to be quite sensitive to these choices [41, 142]. In our experiments we found that simply sampling negatives from other images to be insufficient to produce a more generalizable embedding, and we may need to more carefully consider what constitutes a negative (*e.g.* using the negative phrase augmentation approach of Hinami *et al.* [132] and/or the ambiguous phrase pruning of Zhang *et al.* [84]).

While our phrase identification module only produced small improvements in performance, they were also the only method which we found to help performance in our experiments. Some of the alternatives we tried which all produced worse results were:

- Training an Embedding Model to identify a single phrase based on the whole image representation.

- Training a CCA Model to identify a single phrase based on the whole image representation.
- Using multiple instance learning methods to learn classifiers for individual words that have been used on related tasks (*e.g.* [3, 72, 73]).
- Constructing a phrase detector by combining scores for individual phrases as done in Li *et al.* [143].
- A post-detection method which took the scores for common words or phrases for an image and trained a classifier to rescore them. Concatenating these scores the whole image representation and keeping the top K phrases per image also did not lead to performance improvements in our experiments.
- Concatenating a whole image representation along with our region scores, essentially combining the identification and localization tasks in a single module.

In the end, only by jointly predicting the likelihood of multiple phrases together as done with our retrieval-based module resulted in better performance in our experiments. This suggests that investigation into additional structured prediction approaches may be promising. Since the recall after filtering phrases based on the top 100 sentences on both the caption generation method and retrieval based methods was low (resp. 49% and 63%), methods which diversity these results, especially for rarer phrases, may help improve performance. We can also consider what phrases are typically mutually exclusive to help filter out false positives for phrases which they are commonly confused with (*i.e.* similar to the sampling approach in Hinami *et al.* [132]).

CHAPTER 7: APPLICATIONS TO VIDEO

In this chapter we explore applications of the models used in prior chapters to video tasks. First, we use the two branch network of Wang *et al.* [41], which we also used for our classification module in Chapter 6, to provide a good visual representation for the task of video summarization in Section 7.1. Then, we modify the CITE network introduced in Chapter 5 to localize segments within a video which relate to a natural language query in Section 7.2.

7.1 ENHANCING VIDEO SUMMARIZATION VIA VISION-LANGUAGE EMBEDDING

People today are producing and uploading video content at ever increasing rates. To appeal to potential viewers, videos should be well edited, containing only significant highlights while still conveying the overall story. This is especially important for video from wearable cameras, which can consist of hours of monotonous raw footage. Automatic video summarization techniques [144] can facilitate more rapid video search [145, 146] and ease the burden of editing a long video by hand [147]. Consequently, many methods for computing video summaries have been proposed by researchers [148, 149, 150, 151, 152, 153, 154, 155, 146, 156, 157].

Summarizing video typically involves a tradeoff between including segments that are interesting in their own right and those that are representative for the story as a whole. Some events may be interesting in isolation, but if they are repeated too frequently the summary may become redundant or unrepresentative. Gygli *et al.* [25], whose work we build upon, proposed an optimization approach for balancing the criteria of interestingness and representativeness. Prior work has defined these criteria in abstract mathematical terms (e.g., using notions of sparsity, graph connectedness, or statistical significance) [158, 159, 160] or tried to learn them using implicit or explicit supervision [155, 147, 161, 157]. Generally, it is agreed that bringing in explicit semantic understanding, or the ability to associate video shots with high-level categories or concepts, is helpful for enabling meaningful summaries. A number of approaches have focused on learning limited vocabularies of concepts (often in a weakly supervised manner) from large databases of images and/or video collected from the web [148, 162, 152, 163]. When rich supervision in the form of freeform language (titles, on-screen text, or closed captioning) is available, it becomes possible to use more sophisticated joint vision-language models to capture a wider range of concepts and to extract a more meaningful video summary [164]. Joint modeling of visual content and text is becoming increasingly common for video summarization and retrieval, typically to help identify whether a given shot is relevant to the overall story of a video or a particular user query [165, 166, 167, 146].

Recently, we have seen a proliferation of powerful vision-language models based on state-of-the-art feedforward and recurrent neural networks. Such models have been used for cross-modal retrieval [8, 71, 70, 27, 11, 168, 110], image caption generation [6, 22, 11, 13, 23], and visual storytelling [169, 170]. Motivated by these successful applications, we experiment with a joint image-text embedding as a representation for video summarization. Such an embedding is given by functions trained to project image and text features, which may initially have different dimensionalities, into a common latent space in which proximity between samples reflects their semantic similarity. We use the two-branch neural network of Wang *et al.* [110] to learn a nonlinear embedding using paired images and text (or video and specially produced annotations). Then, at test time, we use the embedding to compute the similarity between two video segments without requiring any language inputs. As we can see from Figure 7.1, even an embedding trained on a different domain, i.e., the Flickr30k dataset of still images and captions [95], can retrieve semantically consistent results for a query video frame (e.g. images of an outdoor market are returned for the second query, or a woman sitting at a table for the third).

An overview of our system is presented in Figure 7.2. We start with the approach of Gygli *et al.* [25], which creates a video summary based on a mixture of submodular objectives on top of vision-only features. We augment this method, which we will refer to as Submod in the following, with a set of vision-language objectives computed in the cross-modal embedding space. The effectiveness of this approach is experimentally demonstrated on the UT Egocentric [171] and TV Episodes [172] datasets, which have different statistics and visual content. Our experiments show that the embedding can be learned on traditional vision-language datasets like Flickr30k [95] while still providing a good representation for the target video datasets. We are able to leverage this improved representation to create more compelling video summaries and, using the same underlying model, allow a user to create custom summaries guided by text input.

7.1.1 Semantically-aware video summarization

A common way of summarizing video is by selecting a sequence of segments that best represent the content found in the input clip. Following the Submod method of Gygli *et al.* [25], we formulate this selection process as optimization of a linear combination of

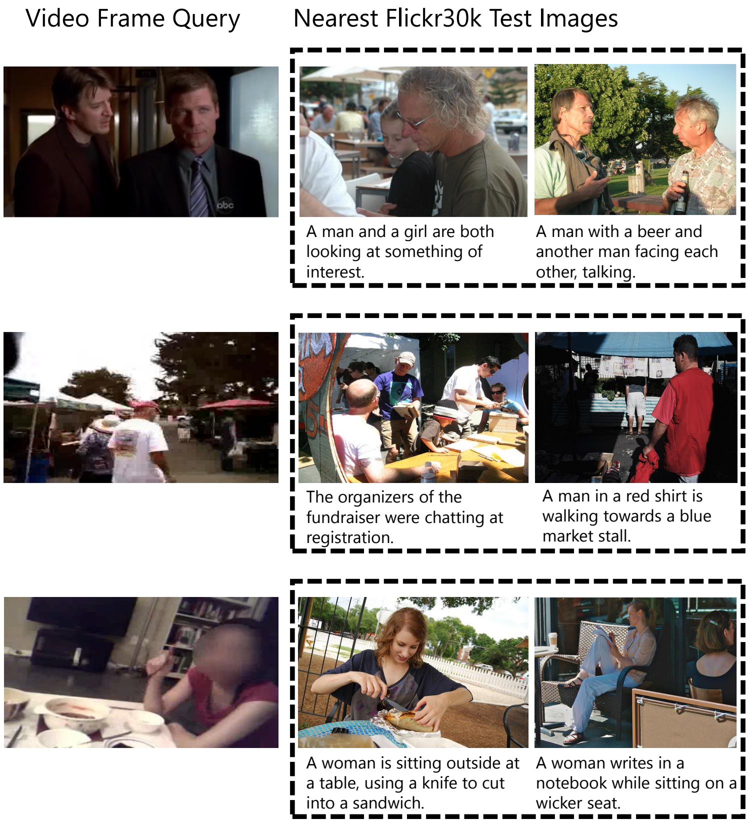

Video Frame Query
Nearest Flickr30k Test Images
A man and a girl are both looking at something of interest.
A man with a beer and another man facing each other, talking.
The organizers of the fundraiser were chatting at registration.
A man in a red shirt is walking towards a blue market stall.
A woman is sitting outside at a table, using a knife to cut into a sandwich.
A woman writes in a notebook while sitting on a wicker seat.

Figure 7.1: Example query video frames (left column) and their best-matching still images with their captions from the Flickr30k dataset [95] (right two columns). The similarity is computed by mapping the visual features describing both the video frames and the still images into a learned vision-language space, which provides a semantically consistent representation for video summarization.

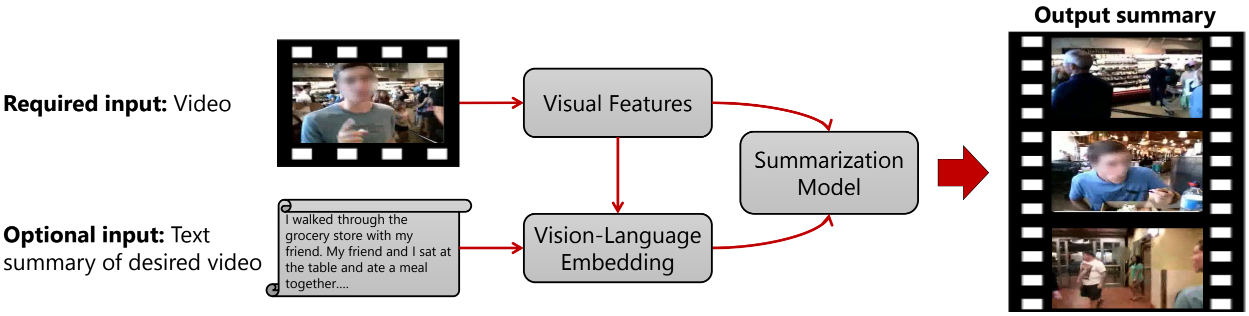

Required input: Video
Optional input: Text summary of desired video
I walked through the grocery store with my friend. My friend and I sat at the table and ate a meal together....
Visual Features
Vision-Language Embedding
Summarization Model
Output summary

Figure 7.2: **Method Overview.** At test time, we assume we are given a video and, optionally, a written description of the desired summary. Our approach projects visual features into a learned vision-language embedding space where similarity reflects semantic closeness. By using this representation, we can produce more diverse and representative summaries than those created with visual features alone. The cross-modal embedding space further enables us to use text input to directly modify a summary.

objectives that capture different traits desired in the output summary. We chose to build on the Submod framework due to its two attractive properties. First, it is generic and easily adaptable to different summarization tasks that may have different requirements. Second, by constraining the weights in the combination to be nonnegative and the objectives to be submodular, a near-optimal solution can be found efficiently [173].

Given a video V consisting of n segments, our goal is to select the best summary $Y \subset V$ (typically subject to a budget or cardinality constraint) based on a weighted combination of visual-only objectives $\phi_o(V, Y)$ and vision-language objectives $\phi_{o'}(V, Y)$:

$$\underset{Y \subset V}{\arg\max} \underbrace{\sum_o w_o \phi_o(V, Y)}_{\text{Visual-Only Objectives}} + \underbrace{\sum_{o'} w_{o'} \phi_{o'}(V, Y)}_{\text{Vision-Language Objectives}} \quad . \tag{7.1}$$

The weights are learned from pairs of videos and output summaries as in [25]. The objectives are restricted to being submodular and the weights to being non-negative, which makes it possible to use a greedy algorithm to obtain approximate solutions Eq. (7.1) with guarantees on the approximation quality.

We start with the same visual-only objectives as in the original Submod method [25], which will be reviewed in Section 7.1.1. The contribution of our work is in proposing new vision-language objectives, which will be introduced in Sections 7.1.1 and 7.1.1.

Visual Objectives

Submod [25] splits the subshot selection task into a mixture of three objectives enforcing representativeness, uniformity, and interestingness, as explained below.

Representativeness. A good summary needs to include all the major events of a video. To measure how well the current summary represents the original video's content, visual features are extracted from each segment and a k-medoids loss function is employed. We can think of the summary as a set of codebook centers, and for each segment from the original video represented by some feature vector f_i, we can map it onto the closest codebook center f_s, and compute the total squared reconstruction error:

$$L(V, Y) = \sum_{i=1}^{n} \min_{s \in Y} ||f_i - f_s||_2^2 \,. \tag{7.2}$$

This is reformulated into a submodular objective:

$$\phi_{rep}(V, Y) = L(V, \{p'\}) - L(V, Y \cup \{p'\}) \,, \tag{7.3}$$

where p' represents a phantom exemplar [150], which ensures we don't take the minimum over an empty set.

As in [25], we represent a segment's visual content f_s by the average of the image features over all its frames. However, we replace the DeCAF features [174] used in [25] with more up-to-date Deep Residual Network features [134] (we use the 2048-dimensional activations before the last fully connected layer of the 152-layer ResNet trained on ImageNet [105]).

Uniformity. The second objective is designed to enforce temporal coherence, as excessively large temporal gaps between segments can interrupt the flow of the story, while segments that are too close to each other can be redundant. The uniformity objective $\phi_{unif}(V, Y)$ is completely analogous to Eq. (7.3), except that the feature representing each frame is simply its mean frame index (i.e., it is a scalar in this case).

Interestingness. Some segments might be preferred over others in a summary, even if they all represent the same event. For example, a segment where a child is smiling and waving at the camera might be preferred to one where they have their back to the camera. The notion of what is "interesting" is typically highly particular to the exact nature of the desired summary and/or application domain, although some generic definitions of "interestingness" have been proposed as well (e.g. [151, 171]). We use the same method as in [25] to produce a per-frame interestingness score for all the frames in a video segment. Since in principle it is possible for different segments to overlap, we sum over the interestingness scores $I(y)$ of all the unique frames y in the current summary Y:

$$\phi_{int}(V, Y) = \sum_{y \in \hat{Y}} I(y)\,, \tag{7.4}$$

where $\hat{Y}$ denotes the union of all the frames in Y. In our experiments, we use this term on only one dataset, UT Egocentric [171], which has per-frame interestingness annotations that can be used for training a classifier for producing the scores $I(y)$. More details about this will be given in Section 7.1.2.

Vision-Language Objectives

We would like to project video features into a learned joint vision-language embedding space, in which we expect similarity to be more reflective of semantic closeness between different video segments. Due to its state-of-the-art performance on vision-language retrieval tasks, we chose to learn our embedding model using the Embedding Network of Wang *et al.* [110], which we also modified for our phrase identification module in Chapter 6. One of the branches of this network takes in original visual features A and the other one takes in text features B. Each branch consists of two fully connected layers with ReLU nonlinearities between them, followed by L2 normalization. The network is trained with a margin-based triplet loss combining bi-directional ranking terms (for each image feature, matching text features should be closer than non-matching ones, and vice versa), and a neighborhood-preserving term (e.g. text features that correspond to the same image should be closer to each other than non-matching text features).

In this dissertation, we experiment with two different embeddings. The first one is trained using the dense text annotations that come with both our video datasets. However, due to the small size and vocabulary of these datasets, as well as the domain-specific nature of their descriptions, this embedding may not generalize well. Thus, we train a second embedding on the Flickr30k dataset [95], which contains 31,783 still images with five sentences each. By using Flickr30k, we can evaluate how well its representation can be transferred to video, which has quite different properties.

We train both embeddings using the code provided by the authors of [110] using the same HGLMM text features and ResNet visual features as in Chapter 6. The output dimensionality of the embedding space is 512.

After learning an embedding, we map our visual features to the shared semantic space and use them to compute two additional objectives we refer to as **semantic representativeness** and **semantic interestingness**. These share the forms of the visual-only versions, i.e., Eqs. (7.3) and (7.4), respectively. While one might assume that these semantic objectives should supersede their visual-only counterparts, our experiments will show that both are needed for best results. Just as semantic representativeness provides a notion of how semantically similar two video segments are, visual representativeness provides a notion of more low-level visual similarity. Ideally, a good summary will be both semantically and visually diverse so as to provide the maximum amount of information under the current budget.

Text-Guided Summarization

Including a vision-language embedding into our summarization model not only allows us to select segments that are more semantically representative and interesting, but also gives us a direct way to incorporate human input when creating a summary, as shown in Figure 7.2. A user can supply a freeform description of the desired summary, and the objective function can be augmented with a term that encourages the result to be consistent with this description. This is similar to the query-focused summarization framework of

Sharghi et al. [166], but rather than consisting of keywords that can apply across many videos, our descriptions can be freeform sentences that are specific to the input video. We consider two scenarios corresponding to different assumptions about the form of the optional language input.

Constrained text guidance. In this version of text guidance, we assume that we are given a written description in which each sentence maps onto a single desired segment. That is, the first selected segment from the video should be consistent with the first sentence in the input description, the second segment should be consistent with the second sentence, and so on. We introduce an additional vision-language objective for Eq. (7.1) based on the sum of inter-modal scores between each summary segment and its corresponding sentence. More precisely, let g_s denote the feature representation of the segment s (i.e., the mean of per-frame feature vectors in the vision-language embedding space), t_s be the representation of the corresponding sentence from the description D, and $\mathrm{sim}(g_s, t_s)$ be the cosine similarity between them. Then our new text guidance objective is given by

$$\phi_{text}(V, Y, D) = \sum_{s \in Y} \mathrm{sim}(g_s, t_s) \, . \tag{7.5}$$

This is similar to what one would do for sentence-to-video retrieval, except the sentences are provided as a set and there are global costs for the summary as a whole (e.g., the uniformity and representativeness objectives). Since we assume that the sentences are given in correct temporal order, when a segment is chosen for a sentence it greatly restricts the available segments for the remaining sentences. Since our target videos have continuous shots with a lot of redundant segments, a standard retrieval approach would likely return a lot of very similar nearby segments in the top few results. The global summary-level costs are necessary to provide diversity.

Unconstrained text guidance. For videos that contain hours of footage, or in cases when a description of the desired summary cannot be written immediately after a video is shot, it may be difficult to remember the correct ordering of events or provide a temporally aligned description. In a related scenario, a user may want to summarize a video they did not shoot and maybe have not even seen – e.g., someone may want to summarize a soccer match and is particularly interested in corner kicks. For these reasons, we also implement an unconstrained version of text guidance, in which the input sentences and the associated video segments do not have to appear in the same order. This results in a bipartite matching problem between a set of candidate segments and the list of sentences which we solve using the Hungarian algorithm. After obtaining the assignments, we compute the text guidance objective using Eq. (7.5).

7.1.2 Experiments

Protocol and Implementation Details

Datasets. We evaluate our approach on two datasets for which detailed segment-level text annotations are available: the UT Egocentric (UTE) dataset [171] and the TV Episodes dataset [172]. The UTE dataset consists of four wearable camera videos capturing a person's daily activities. Each video is three to five hours long, for a total of over 17 hours. The TV Episodes dataset [172] consists of four videos of three different TV shows that are each 45 minutes long.

For both UTE and TV Episodes datasets, Yeung *et al*. [172] provided dense text annotations for each 5- and 10-second video segment, respectively. While the UTE dataset videos are first-person videos taken in an uncontrolled environment, the TV episodes are well edited, third-person videos. As a result of these variations, the text annotations also have some obvious differences in statistics (e.g. the UTE annotations typically begin with a self reference to the camera wearer in the first person, while the TV Episodes typically refer to people by their name in the episode).

Note that there exist other popular benchmarks for video summarization, including SumMe [151] and TVSUM [167] datasets. However, we did not include them in our evaluation as they do not have text annotations on which a vision-language embedding model could be trained.

Training. For each video in the UTE and TV Episodes datasets, Yeung *et al*. [172] have supplied three human-composed reference text summaries. To train the weights for different objectives in the Submod method, these summaries need to be mapped to suitable subsets of segments in the videos by matching sentences from the summaries to the original per-segment video annotations. We follow the same greedy n-gram matching and ordered subshot selection procedures as previous work [25, 172] to obtain 15 training summaries for each video.

For each dataset, we use a four-fold cross-validation setup, training on each each subset of three videos and testing on the fourth one. This involves training the vision-language embedding (for models that do not use the Flickr30k-trained embedding), the interestingness function (only on the UTE dataset, as detailed in Section 7.1.2) and the weights in Eq. (7.1). For the latter step, the training data consists of 45 video-summary pairs.

Testing and evaluation. For both datasets, we set our budget (i.e., the maximum number of segments that can be selected) at 24, producing 2-minute summaries on the UTE dataset and 4-minute summaries on the TV Episodes dataset. Following [25, 166, 172], we evaluate video summarization in the text domain. At test time, given a video summary generated by our method, we create the corresponding text summary by concatenating the original text annotations of the segments that make up the summary. We use non-overlapping segments for each dataset so as to have a non-ambiguous mapping to the text annotations, though the Submod approach is still applicable to video segmentations that produce overlapping segments [25]. The automatically produced summary is compared against the three human-provided reference summaries using the recall-based ROUGE metric [175]. Note that this evaluation is content-based: multiple segments may score the same if they are associated with the same or a very similar text description regardless of their relative visual quality (e.g., a blurry segment may be considered as good as a sharper one). As in prior work [25, 166], we report the recall and f-measure on each dataset using the ROUGE-SU score, which demonstrated the strongest correlation with human judgment [172]. We use the same ROUGE parameters as in [25, 172], obtained through personal communication with the authors.

In our evaluation, we compare the following baselines and variants of our method:

1. *Sampling.* Baselines that sample segments in the testing video uniformly or randomly. We run these baselines five times each and report the mean results.

2. *Video MMR.* The approach of [176] as implemented by the authors of [172]. They provided us their output summaries on the UTE dataset only, and we evaluated them using our ROUGE settings.

3. *seqDPP.* The approach of [177] using their code. We replace their SIFT-based feature representation [178] with our ResNet features which we also use to compute the context-based representation required in this method. We concatenate these with features computed over a saliency map [179] as in [177].

4. *Submod-V.* The original Submod approach using the code of Gygli *et al.* [25] and their visual-only objectives.

5. *Submod-S.* Submod which replaces visual-only representativeness and interestingness with the semantic versions.

6. *Submod-V + Sem. Inter.* Combination of the semantic interestingness objective with the visual-only objectives.

7. *Submod-V + Sem. Rep.* Combination of the semantic representativeness objective with the visual-only objectives.

	Method	F-measure	Recall
(a)	Baselines		
	Random	26.51	25.23
	Uniform	28.13	25.76
	Video MMR [176]	22.73	20.80
	seqDPP [177]	28.87	26.83
	Submod-V [25]	29.35	27.43
(b)	Flickr30k Embedding		
	Submod-S	27.18	29.69
	Submod-V+Sem. Inter.	31.44	28.28
	Submod-V+Sem. Rep.	32.40	30.00
	Submod-V+Both	33.50	31.16
(c)	UTE Embedding		
	Submod-S	29.54	31.01
	Submod-V+Sem. Inter.	31.58	29.24
	Submod-V+Sem. Rep.	33.24	30.84
	Submod-V+Both	34.15	31.59

Table 7.1: **UT Egocentric summarization performance.** (a) contains our baselines including our reproduction of [177, 25] using their code with updated visual features. (b-c) demonstrates the effectiveness of our vision-language objectives on this task using embeddings trained on different datasets.

8. *Submod-V + Both.* Combination of the semantic interestingness and semantic representativeness objectives with the visual-only objectives.

Note that variants 6 and 8 above are only available on the UTE dataset since it is the only one that has an interestingness function.

UTE Dataset Results

For this dataset, Lee *et al.* [171] have provided importance annotations that can be used to train an interestingness classifier. Following [25], we learn to predict the interestingness of a video segment (as a binary label) using a support vector machine with a radial basis function kernel over our visual or vision-language features. As in [25], we compute features on the whole image rather than on regions as in [171]. For reference, the resulting classifier using the visual features has an average precision of 56.2 on the annotated frames.

We evaluate our approach on two-minute-long summaries in Table 7.1. Our new semantic features trained on UTE data provide a combined improvement in f-measure of nearly 5%, with a 4% improvement in recall as shown in the last line of Table 7.1(c). A majority of that

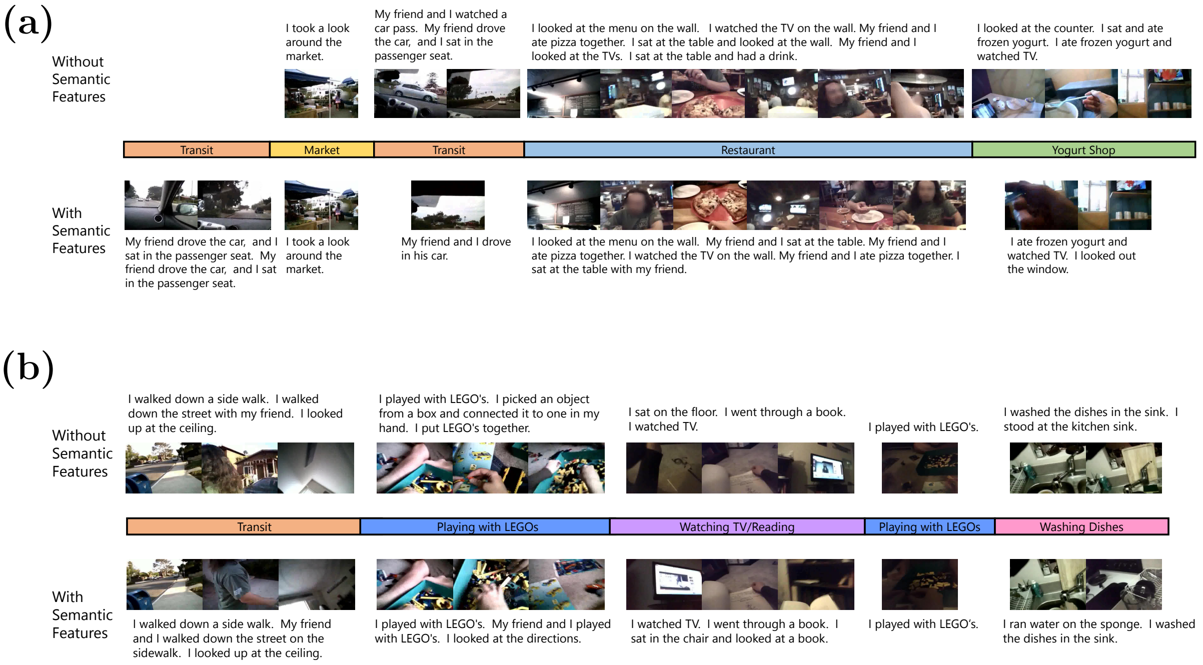

(a)
Without Semantic Features
I took a look around the market.
My friend and I watched a car pass. My friend drove the car, and I sat in the passenger seat.
I looked at the menu on the wall. I watched the TV on the wall. My friend and I ate pizza together. I sat at the table and looked at the wall. My friend and I looked at the TVs. I sat at the table and had a drink.
I looked at the counter. I sat and ate frozen yogurt. I ate frozen yogurt and watched TV.
Transit
Market
Transit
Restaurant
Yogurt Shop
With Semantic Features
My friend drove the car, and I sat in the passenger seat. My friend drove the car, and I sat in the passenger seat.
I took a look around the market.
My friend and I drove in his car.
I looked at the menu on the wall. My friend and I sat at the table. My friend and I ate pizza together. I watched the TV on the wall. My friend and I ate pizza together. I sat at the table with my friend.
I ate frozen yogurt and watched TV. I looked out the window.
(b)
Without Semantic Features
I walked down a side walk. I walked down the street with my friend. I looked up at the ceiling.
I played with LEGO's. I picked an object from a box and connected it to one in my hand. I put LEGO's together.
I sat on the floor. I went through a book. I watched TV.
I played with LEGO's.
I washed the dishes in the sink. I stood at the kitchen sink.
Transit
Playing with LEGOs
Watching TV/Reading
Playing with LEGOs
Washing Dishes
With Semantic Features
I walked down a side walk. My friend and I walked down the street on the sidewalk. I looked up at the ceiling.
I played with LEGO's. My friend and I played with LEGO's. I looked at the directions.
I watched TV. I went through a book. I sat in the chair and looked at a book.
I played with LEGO's.
I ran water on the sponge. I washed the dishes in the sink.

Figure 7.3: Output summaries of UT Egocentric Video 2 produced with and without the semantic features, corresponding to models in the last lines of Table 7.1(c) and (a), respectively. Parts (a) and (b) show the first and second halves of the summary. For better readability, we add a color-coded timeline hand-annotated with high-level scenes (e.g., *Transit*, *Market*). **(a)** The first *Transit* scene is captured with the semantic features and missed otherwise. **(b)** While the two summaries represent each scene with an equal number of segments, we can see a difference in the precise segments that are selected: In the *Washing Dishes* scene, the summary based on semantic features selects segments more representative of dishwashing, rather than simply standing at the sink.

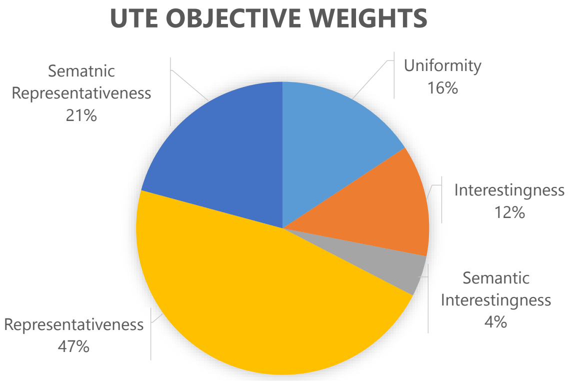


Figure 7.4: Learned weights for the five objectives of our best-performing model on the UT Egocentric dataset, averaged across the four training-test splits.

	Method	F-measure	Recall
(a)	Baselines		
	Random	32.83	28.88
	Uniform	33.90	29.15
	seqDPP [177]	35.39	32.12
	Submod-V [25]	38.18	33.47
(b)	Flickr30k Embedding		
	Submod-S	38.92	35.28
	Submod-V+Sem. Rep.	39.87	36.50
(c)	TV Episodes Embedding		
	Submod-S	37.29	32.75
	Submod-V+Sem. Rep.	40.90	37.02

Table 7.2: **TV Episodes summarization performance.** (a) Baselines, including our reproduction of [177, 25] using their code with updated visual features. (b,c) Different combinations of vision-language objectives using embeddings trained on Flickr30k and TV Episodes, respectively.

gain comes from our semantic representativeness objective. Despite having very different text annotations on images with different statistics, the semantic features trained on the Flickr30k dataset perform nearly as well as the UTE-trained features.

Figure 7.4 visualizes the weights of the five objectives in our best-performing model. We can see that visual and semantic representativeness get the two highest weights, adding up to more than 60% of the total, followed by uniformity. The two interestingness objectives have the smallest (though still non-negligible) contribution, indicating that representativeness does most of the job of capturing story elements.

Qualitatively, the performance gains afforded by the semantic features appear to stem primarily from the addition of missing story elements. An example of this is shown in Figure 7.3(a), where the car drive to the market is completely missing from the output summary without the semantic features. Another manifestation, although more subtle, can be seen in the *Washing Dishes* section of Figure 7.3(b). The segment being chosen with semantic features corresponds to an action that is common in washing dishes (rinsing a sponge), while without semantic features the user is just standing there.

TV Episodes Results

The TV Episodes dataset does not provide per-frame importance annotations with which to train a semantic interestingness classifier, so we do not use the interestingness objective of Eq. (7.4) here. The results in Table 7.2 show that augmenting the visual representativeness

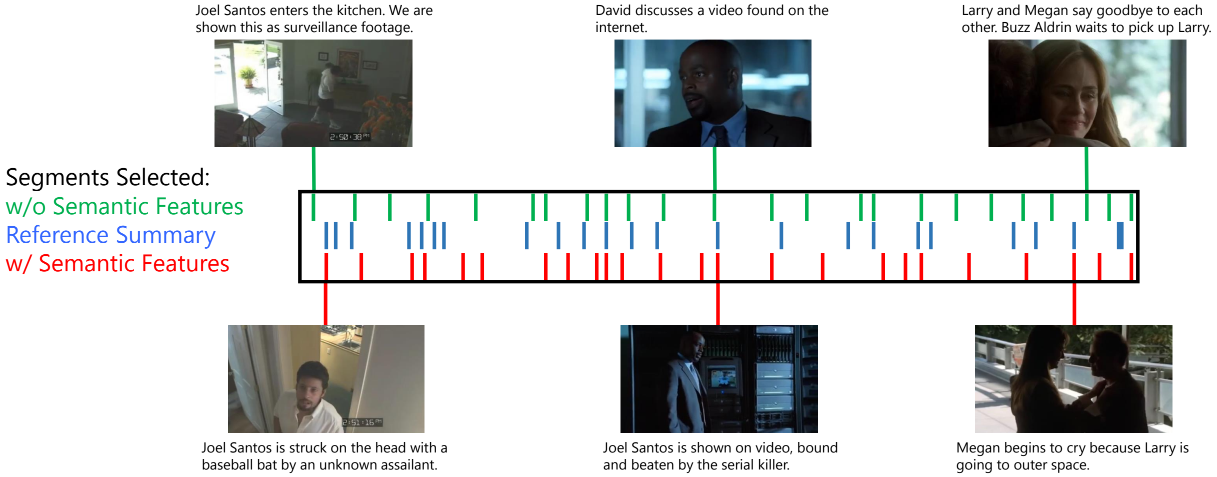

Joel Santos enters the kitchen. We are shown this as surveillance footage.
David discusses a video found on the internet.
Larry and Megan say goodbye to each other. Buzz Aldrin waits to pick up Larry.
Segments Selected:
w/o Semantic Features
Reference Summary
w/ Semantic Features
Joel Santos is struck on the head with a baseball bat by an unknown assailant.
Joel Santos is shown on video, bound and beaten by the serial killer.
Megan begins to cry because Larry is going to outer space.

Figure 7.5: Comparison between the video summaries on Video 4 from the TV Episodes dataset produced with and without the semantic representativeness objective. For completeness, we also show the frames from the reference summary. The summary with semantic features more commonly selects segments found in the reference summary. The figure shows frames from three such occurrences along with the closest selected segment in the summary without semantic features.

and uniformity objectives with the semantic representativeness objective once again provides an improvement. As can be seen from Table 7.2(c), semantic representativeness computed on top of the TV Episodes embedding increases the f-measure by 1.5%, and recall by just over 3%. As on the UTE dataset, the embedding trained on the dataset itself performs slightly better than the Flickr30K-trained one.

Overall, the absolute improvement in the ROUGE scores here is smaller than on UTE. In fact, of the four training-test splits, adding semantic representativeness improves results in two cases and actually makes them worse in the other two, though the absolute improvements end up being larger. We also see much higher variance in the per-objective weights learned by the Submod method on TV Episodes than on UTE. Part of the problem is the limited amount of training data. We also suspect an interestingness objective as used for the UTE dataset would help stabilize the summaries and make them more meaningful.

Figure 7.5 compares summaries produced with and without semantic representativeness on the fourth TV Episodes video. The result with the semantic objective more commonly agrees with the segments in the reference summary. On the left, the segment with the semantic features focuses on selecting a segment deemed more critical to the story of the original video (i.e. Joel being attacked vs. him walking around his house). For the center pair of segments, semantic representativeness selects the segment when a video of Joel's attack is shown at the police department, instead of a segment where the video is simply mentioned.

Dataset	Text Guidance	F-measure	Recall
UTE	Unconstrained	34.90	31.77
	Constrained	35.21	32.31
TV Eps.	Unconstrained	41.18	38.14
	Constrained	41.17	38.11

Table 7.3: Performance on text-constrained summarization, when the written description of the desired summary is given as an additional input at test time. We are using our full models with the vision-language embedding trained on the respective datasets (corresponding to the last lines of Tables 7.1 and 7.2).

Text-Guided Summarization Results

Table 7.3 shows the evaluation of text-guided summarization, where a reference text description is provided as an additional input at test time. These results are obtained with our full models with the vision-language embedding trained on the respective datasets. Comparing the results in Table 7.3 with the last lines of Tables 7.1 and 7.2, we see gains across both datasets. While one might think the constrained version, where the written description is provided in temporal order, would perform better, we only see this manifest on the UTE dataset. On the TV Episodes dataset, the two versions perform about the same. We believe this is not only due to the differences in length of the raw videos, but also the repetitive nature of the different scenes. Although the videos in the UTE dataset form a continuous stream and tend to change gradually, once a place is left is isn't often revisited. Looking at the different story elements in Figure 7.3(a) and Figure 7.3(b), only *Transit* and *Playing with LEGOs* is repeated. In contrast, the nature of the TV Episodes dataset means that the general visual elements corresponding to different sets may occur multiple times. The offices where the people work, the homes of suspects, or crime scenes (as these TV Episodes are of crime shows) are often repeated, making it challenging to identify the specific scene being described without considering the audio as well. The unconstrained model appears to be more robust to this kind of confusion.

While our work shows the promise of video summarization datasets accompanied by rich text annotations, like the ones released by Yeung et al. as part of their VideoSET framework [172], it also shows their limitations. In particular, these datasets have only a few videos that can be highly variable. Thus, the amounts of training and test data are not necessarily sufficient to draw firm conclusions about the relative advantages of different summarization methods (in our case, we struggled with instability issues on the TV Episodes dataset). Compounding the problem are the inconsistencies in the kinds of annotations that are available for different datasets (in particular, annotations that can be used to train good interestingness objectives) and the evaluation methodologies that are proposed in the literature. While efforts like VideoSET are a good start, they need to be greatly expanded in scope.

7.2 LOCALIZING DESCRIBABLE MOMENTS IN VIDEO

In this section we take a natural language query as input and retrieve the segment of the video which it corresponds to, thus enabling video summarization approaches like ours to make small edits with minimal user input. This is closely related to the video-sentence retrieval task, except we are finding a relevant video segment rather than retrieving the whole video. We train our joint vision-language representation using the CITE network described in Chapter 5. This network trains a set of embeddings, which share some weights, along with an attention model which maps queries to the trained embeddings (see Figure 5.1 for a visualization of the network architecture). We replace the first fully connected layer, which operate on the vision and language features in isolation before the element-wise product, with a pair of gated recurrent units (GRUs) [180]. Rather than train two separate models for video inputs using the optical flow and RGB visual representations we shall use to describe the visual information in the videos, we train a single model which share the same language representation. We include a diversity term during training using the DeCov loss of Cogswell *et al.* [181] to encourage the visual features being fed into the conditional embeddings to be decorrelated. The output of these models are combined with two priors which leverage a bias towards different segments being selected before being fed into a global context re-ranker which uses the predictions made across an entire video to re-score each segment's likelihood of being associated with a query. A summary of our modified CITE network is provided in Figure 7.6.

The DiDeMo dataset was introduced by Hendricks *et al.* [141], who localized moments using a pair of separately trained two branch networks (one for RGB features and another for optical flow features), using a combination of local, global, and temporal features. By contrast, we train model which combines both these feature representations in a single end-to-end model. Other methods on similar tasks are limited by the domain, *e.g.* Tellex *et al.* [182] supported a limited number of queries over surveillance videos and Lin *et al.* [183] learned a model which could search for a video segment which was relevant to a query from video taken from a dashboard car camera.

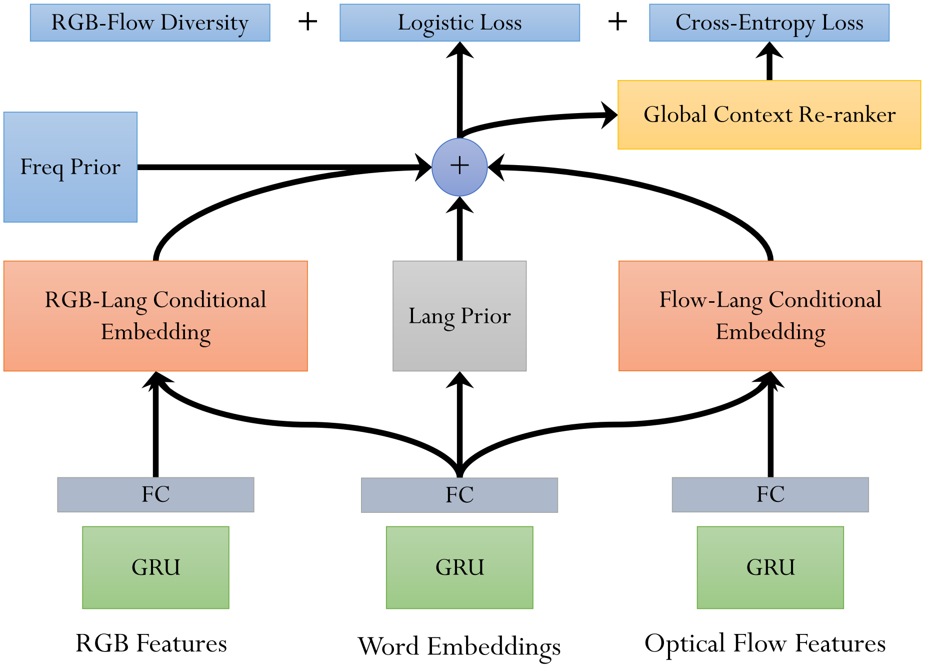


Figure 7.6: An overview of our language-segment localization approach. Each feature representation for each word or image frame is fed into a GRU followed by a fully connected layer to obtain a joint encoding over the entire input. These are fed into conditional embeddings contain the joint vision-language components of the CITE network in Figure 5.1 (*i.e.* starting from the element-wise product).

7.2.1 Video Segment Localization Approach

To retrieve segments from a video which are relevant to a natural language query we build upon the CITE network introduced in Chapter 5. This model learns a set of K conditional embeddings, each meant to capture a different concept which is important for localizing a query. The input to these conditional embeddings is a joint vision-language representation obtained from the output of a pair of fully connected layers which take an elementwise product of the vision and language representations obtained from our GRUs as input. We use the natural language queries to learn an attention model over the K concepts. A single fully connected layer is used to produce a single score for a video segment and language query pair. See Figure 7.6 for an overview of the approach. The loss function used to train our model is provided in Eq. (5.2) which consists of a logistic loss with L1 regularization being applied to the outputs of the attention model in order to promote sparsity.

Visual Features and Priors

In this work we take advantage of two primary sources of visual features: RGB and optical flow. Rather than learn two separate models which we combine after training. we train a single model which takes both representations as input. This way we can reduce the size of our model by sharing redundant components (*e.g.* we use a single recurrent model to learn a representation for our language inputs). We encourage the two different types of visual features being used as input to our conditional embeddings to learn complementary information by using the DeCov loss of Cogswell *et al.* [181]. Let C be the covariance matrix between RGB features used as inputs to the RGB-Lang Conditional Embedding in Figure 7.6(a) and the optical flow features used as inputs to the Flow-Lang Conditional Embedding. The DeCov loss is defined as:

$$L_{DeCov} = \frac{1}{2}(\|C\|_F^2 - \|diag(C)\|_2^2), \tag{7.6}$$

where $\|\cdot\|_F$ is the frobenius norm.

To take advantage of the biases that arise in where queries may be located we also use two priors which capture different information. First, we use a moment frequency prior when we have video inputs which assigns likelihoods to video segments based on how often temporally aligned segments were observed in the training data (this is represented as the Freq Prior in Figure 7.6(a)). Second, we predict the location of a query based entirely on the language features (shown as the Lang Prior in Figure 7.6(a)), which consists of two fully connected layers followed by a softmax. This can be seen as analogous to the word prior in Yeh *et al.* [94] and the candidate position cue used in Chapter 4. These priors are linearly summed together along with the predictions made by the conditional embeddings.

Global Context Re-ranker

So far predictions are made on an individual basis. However, two video segments in a sequence with high likelihoods to relate to a text query can be indicative that the union of the two segments may actually be the correct segment. To take advantage of these types of cues, we re-rank our predictions based on all the scores made for a single video. After getting our initial set of predictions, we use a pair of fully connected layers, the first of which projects the scores into a higher dimensional space before the second is used to produce the final rankings. We use a softmax with a cross entropy loss to train the weights of our re-ranker which we shall refer to as L_{RR}. Each component of our localization module is trained as a single model in and end-to-end fashion. Thus, our total loss is,

	Method	R@1	R@5	mIOU
(a)	LSTM-RGB	13.10	44.82	25.13
	LSTM-Flow	18.35	56.25	31.46
	LSTM-Late Fusion	18.71	57.47	32.32
	LSTM-Late Fusion+global	19.88	62.39	33.51
	LSTM-Late Fusion+global+tef (MCN)	27.57	79.69	41.70
(b)	CITE-GRU-RGB	16.11	54.92	27.22
	CITE-GRU-Flow	21.74	65.33	36.58
	CITE-GRU-Fusion	21.78	65.52	37.00
	CITE-GRU-Fusion+Diversity	22.98	65.97	36.92
	CITE-GRU-Fusion+Diversity+global	22.14	62.67	36.25
	CITE-GRU-Fusion+Diversity+tef	21.90	66.47	37.09
	CITE-GRU-Fusion+Diversity+LP	26.76	75.51	41.68
	CITE-GRU-Fusion+Diversity+LP+RR	28.24	80.03	42.27
	CITE-GRU-Fusion+Diversity+LP+RR+Freq	29.08	81.70	42.61

Table 7.4: Localization ablation study on the DiDeMo validation set. **(a)** contains results taken from Hendricks *et al.* [141], **(b)** contains the results from our approach.

$$L_{total} = L_{CITE} + \alpha_1 L_{DeCov} + \alpha_2 L_{RR} \tag{7.7}$$

where L_{CITE} refers to the localization loss described by Eq. (5.2) and α_{1-2} are scalar parameters.

7.2.2 Localization Experiments

We evaluate our approach using the DiDeMo dataset [141] which consists of just over 10,000 videos each of which is paired with 3-5 video segment descriptions. These videos are split into 8,395 for training, 1,004 for testing, and 1,065 for validation. Following Hendricks *et al.* [141], we use the VGG19 layer network [104] pretrained on ImageNet [105] to compute our RGB features and the activity recognition model of Wang *et al.* [184] to compute optical flow features. Each video is broken up into 5 second disjoint video segments with a maximum video length of 30 seconds. Segments can be combined into longer moments as long as they result into a continuous segment, resulting in a total of 21 possible combinations. A segment is deemed to be correctly localized if the segment related to a natural language input is within the top K results. We also report the mean intersection-over-union of the the top ranked segment. Following [141], rather than consolidate all human annotations into one ground truth, the highest scoring ground truth segment for each query is used for evaluation.

Method	R@1	R@5	mIOU
Freq [141]	19.40	66.38	26.65
CCA [141]	18.11	52.11	37.82
SCRC [88]	15.57	48.32	30.55
MCN [141]	28.10	78.21	41.08
CITE-GRU-Fusion+Diversity+LP+RR+Freq	29.46	80.16	42.34

Table 7.5: Localization performance on the DiDeMo test set.

Results. We provide an ablation study of our approach in Table 7.4 which reports performance on the validation set. The first two lines of Table 7.4(b) show that our model has a 2-5% improvement in R@1 and mIOU using the multiple-embedding approach of our CITE network using just the RGB and optical flow features compared to the single embedding baseline of Hendricks *et al*. [141], with a 9-10% improvement in R@5. We see in the third and fourth lines of Table 7.4(b) that including our diversity term is key to obtaining performance improvements using the two different feature representations. This constitutes our best model using only visual features that don't take into account the bias in the dataset, which is 3% better than the analogous model in the baseline on fourth line of Table 7.4(a). The fifth and sixth lines of Table 7.4(b) report the effect of concatenating the features of the entire video to provide context (global) and the location of the segment in the video (tef). Notably, while these features were key in obtaining good performance in Hendricks *et al*. [141], they actually hurt performance in our model. The last three lines of Table 7.4(b) show the further improvement we can obtain by using our language and frequency priors, as well as a global context re-ranker, where our best model obtains a 1-2% improvement over the approach of Hendricks *et al*. [141]. These results are mimicked by our experiments on the test set shown in Table 7.5.

CHAPTER 8: CONCLUSIONS

Phrase grounding is an important constituent task for many applications in computer vision. This dissertation introduced a new dataset, Flickr30K Entities, along with a new task of phrase localization in Chapter 3. We built upon our work by introducing a set of hand-crafted and automatically learned concepts which provide significant performance improvements for phrase grounding in Chapters 4 and 5. While these have focused on learning better ways of relating regions to text, there has been some disjoint ground work on obtaining a better set of bounding box proposals [91, 92, 94]. Combining the two should lead to better grounding performance. In addition, many of the global inference methods (*e.g.* the approach in Chapter 4) have only been applied to the Flickr30K Entities dataset. Exploring ways of extending some of these ideas to other grounding datasets whose region level annotations are not based on captions would be an important step in generalizing such approaches. One possibility is consider a scene graph representation [66], which may also help with the annotation sparsity problem discussed in Chapter 6 as it learns alternative ways of referring to the same entity.

Grounding approaches still tend to work poorly on small objects (*e.g.* body parts), those that commonly appear with other entities (*e.g.* instruments which tend to be seen with people), and rare phrases. This is also likely a reason why grounding performance on Flickr30K Entities tends to be higher than datasets like ReferIt Game and Visual Genome. Flickr30K Entities contains (larger) salient entities such are biased towards the center of the image [185], compared with the object-based annotations in ReferIt Game or the dense annotations of Visual Genome. Extending the part-of-speech based cues presented in Chapter 4 to these datasets may help improve performance (additional discussion on future directions on the localization task is provided in Chapter 5).

Despite the models used on phrase localization also being shown to extend to tasks like to localizing events in video (Chapter 7), visual question answering (*e.g.* [80, 82]), and visual relationship detection [29], we find the tasks with they have the most benefit are ones where there is limited variation in the types of visual inputs they must discriminate between (*e.g.* regions from the same image). This manifests itself in the limited performance gains on tasks like image captioning (*e.g.* [74]), bidirectional retrieval (Chapter 3), and our more generalized task of phrase detection, where we both identify if a phrase is associated with an image and localize it in Chapter 6. Promising directions for future work on detection task could include better negative region sampling methods or structured prediction models which could be used to reduce false positive rates as discussed further at the end of Chapter 6.

REFERENCES

[1] X. Chen and C. L. Zitnick, "Learning a recurrent visual representation for image caption generation," *arXiv:1411.5654*, 2014.

[2] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *arXiv:1411.4389*, 2014.

[3] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. Platt et al., "From captions to visual concepts and back," *arXiv:1411.4952*, 2014.

[4] A. Farhadi, S. Hejrati, A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. A. Forsyth, "Every picture tells a story: Generating sentences from images," in *ECCV*, 2010.

[5] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *JAIR*, 2013.

[6] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," *arXiv:1412.2306*, 2014.

[7] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv:1411.2539*, 2014.

[8] B. Klein, G. Lev, G. Sadeh, and L. Wolf, "Fisher vectors derived from hybrid gaussian-laplacian mixture models for image annotation," in *CVPR*, 2015.

[9] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, "Baby talk: Understanding and generating image descriptions," in *CVPR*, 2011.

[10] R. Lebret, P. O. Pinheiro, and R. Collobert, "Phrase-based image captioning," *arXiv:1502.03671*, 2015.

[11] J. Mao, W. Xu, Y. Yang, J. Wang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-RNN)," *arXiv:1412.6632*, 2014.

[12] V. Ordonez, G. Kulkarni, and T. L. Berg, "Im2Text: Describing images using 1 million captioned photographs," *NIPS*, 2011.

[13] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," *arXiv:1411.4555*, 2014.

[14] B. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu, "I2T: Image parsing to text description," *Proc. IEEE*, vol. 98, no. 8, pp. 1485 – 1508, 2010.

[15] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual Question Answering," in *ICCV*, 2015.

[16] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu, “Are you talking to a machine? dataset and methods for multilingual image question answering,” in *NIPS*, 2015.

[17] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, “Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations,” *IJCV*, 2017.

[18] M. Malinowski and M. Fritz, “A multi-world approach to question answering about real-world scenes based on uncertain input,” in *NIPS*, 2014.

[19] M. Ren, R. Kiros, and R. Zemel, “Exploring models and data for image question answering,” in *NIPS*, 2015.

[20] L. Yu, E. Park, A. C. Berg, and T. L. Berg, “Visual Madlibs: Fill in the blank Image Generation and Question Answering,” in *ICCV*, 2015.

[21] J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell, “Language models for image captioning: The quirks and what works,” in *arxiv.org:1505.01809*, 2015.

[22] A. Karpathy, A. Joulin, and L. Fei-Fei, “Deep fragment embeddings for bidirectional image sentence mapping,” in *NIPS*, 2014.

[23] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” *arXiv:1502.03044*, 2015.

[24] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, “ReferItGame: Referring to objects in photographs of natural scenes,” in *EMNLP*, 2014.

[25] M. Gygli, H. Grabner, and L. Van Gool, “Video summarization by learning submodular mixtures of objectives,” in *CVPR*, 2015.

[26] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, “Flickr30K Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models,” in *ICCV*, 2015.

[27] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, “Flickr30K Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models,” *IJCV*, vol. 123, no. 1, pp. 74–93, May 2017.

[28] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, “Visual relationship detection with language priors,” in *ECCV*, 2016.

[29] B. A. Plummer, A. Mallya, C. M. Cervantes, J. Hockenmaier, and S. Lazebnik, “Phrase localization and visual relationship detection with comprehensive image-language cues,” in *ICCV*, 2017.

[30] B. A. Plummer, M. Brown, and S. Lazebnik, "Enhancing video summarization via vision-language embedding," in *CVPR*, 2017.

[31] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan, "Matching words and pictures," *JMLR*, vol. 3, pp. 1107–1135, 2003.

[32] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic image annotation and retrieval using cross-media relevance models," in *ACM SIGIR*, 2003, pp. 119–126.

[33] K. Barnard and D. Forsyth, "Learning the semantics of words and pictures," in *ICCV*, 2001.

[34] A. Gupta and L. S. Davis, "Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers," in *ECCV*, 2008.

[35] R. Fergus, L. F. Fei, P. Perona, and A. Zisserman, "Learning object categories from Google's image search," in *ICCV*, 2005.

[36] V. Ferrari and A. Zisserman, "Learning visual attributes," in *NIPS*, 2007.

[37] M. A. Sadeghi and A. Farhadi, "Recognition using visual phrases," in *CVPR*, 2011.

[38] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik, "Improving image-sentence embeddings using large weakly annotated photo collections," in *ECCV*, 2014.

[39] D. R. Hardoon, S. R. Szedmak, and J. R. Shawe-taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[40] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *ICML*, 2013.

[41] L. Wang, Y. Li, and S. Lazebnik, "Learning two-branch neural networks for image-text matching tasks," *arXiv:1704.03470*, 2017.

[42] H. Hotelling, "Relations between two sets of variates," *Biometrika*, pp. 321–377, 1936.

[43] F. Yan and K. Mikolajczyk, "Deep correlation for matching images and text," in *CVPR*, 2015.

[44] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *ICML*, 2011.

[45] F. Feng, X. Wang, and R. Li, "Cross-modal retrieval with correspondence autoencoder," in *ACM MM*, 2014.

[46] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," *JMLR*, vol. 15, pp. 2949–2980, 2014.

[47] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "DeViSE: A deep visual-semantic embedding model," in *NIPS*, 2013.

[48] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Object detection with discriminatively trained part-based models," *TPAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[49] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[50] R. Girshick, "Fast R-CNN," in *ICCV*, 2015.

[51] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.

[52] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *ECCV*, 2016.

[53] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[54] T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," in *CVPR*, 2013.

[55] M. Jain, J. C. van Gemert, and C. G. M. Snoek, "What do 15,000 object categories tell us about classifying and localizing actions?" in *CVPR*, 2015.

[56] Z. Fu, T. Xiang, E. Kodirov, and S. Gong, "Zero-shot object recognition by semantic manifold distance," in *CVPR*, 2015.

[57] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Synthesized classifiers for zero-shot learning," in *CVPR*, 2016.

[58] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele, "Latent embeddings for zero-shot classification," in *CVPR*, 2016.

[59] D. Jayaraman and K. Grauman, "Zero shot recognition with unreliable attributes," in *NIPS*, 2014.

[60] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *TPAMI*, vol. 36, no. 3, pp. 453–465, 2014.

[61] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting image annotations using Amazon's mechanical turk," in *NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. ACL, 2010, pp. 139–147.

[62] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results," http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html, 2008.

[63] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014.

[64] M. Grubinger, P. Clough, H. Müller, and T. Deselaers, "The iapr tc-12 benchmark: A new evaluation resource for visual information systems," in *International Workshop OntoImage*, 2006, pp. 13–23.

[65] J. Mao, H. Jonathan, A. Toshev, O. Camburu, A. Yuille, and K. Murphy, "Generation and comprehension of unambiguous object descriptions," *CVPR*, 2016.

[66] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *CVPR*, 2015.

[67] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler, "What are you talking about? text-to-image coreference," in *CVPR*, 2014.

[68] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGB-D images," in *ECCV*, 2012.

[69] C. L. Zitnick and D. Parikh, "Bringing semantics into focus using visual abstraction," in *CVPR*, 2013.

[70] L. Ma, Z. Lu, L. Shang, and H. Li, "Multimodal convolutional neural networks for matching image and sentence," in *ICCV*, 2015.

[71] G. Lev, G. Sadeh, B. Klein, and L. Wolf, "RNN fisher vectors for action recognition and image annotation," in *ECCV*, 2016.

[72] I. Misra, C. L. Zitnick, M. Mitchell, and R. Girshick, "Seeing through the Human Reporting Bias: Visual Classifiers from Noisy Human-Centric Labels," in *CVPR*, 2016.

[73] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," in *ICCV*, 2017.

[74] C. Liu, J. Mao, F. Sha, and A. Yuille, "Attention correctness in neural image captioning," in *AAAI*, 2017.

[75] Y. Zhu, O. Growth, M. Berstein, and L. Fei-Fei, "Visual7w: Grounded question answering in images," in *CVPR*, 2016.

[76] K. J. Shih, S. Singh, and D. Hoiem, "Where to look: Focus regions for visual question answering," in *CVPR*, 2016.

[77] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image coattention for visual question answering," in *NIPS*, 2016.

[78] Z. Yang, X. He, J. Gao, L. Deng, , and A. Smola, "Stacked attention networks for image question answering," in *CVPR*, 2016.

[79] A. Mallya and S. Lazebnik, "Learning models for actions and person-object interactions with transfer to question answering," *ECCV*, 2016.

[80] T. Tommasi, A. Mallya, B. A. Plummer, S. Lazebnik, A. C. Berg, and T. L. Berg, "Solving Visual Madlibs with multiple cues," in *BMVC*, 2016.

[81] P. Wang, Q. Wu, C. Shen, and A. van den Hengel, "The VQA-machine: Learning how to use existing vision algorithms to answer new questions," in *CVPR*, 2017.

[82] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," in *EMNLP*, 2016.

[83] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, "Grounding of textual phrases in images by reconstruction," in *ECCV*, 2016.

[84] Y. Zhang, L. Yuan, Y. Guo, Z. He, I.-A. Huang, and H. Lee, "Discriminative bimodal networks for visual localization and detection with natural language queries," in *CVPR*, 2017.

[85] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *CVPR*, 2016.

[86] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," in *ECCV*, 2016.

[87] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," in *CVPR*, 2016.

[88] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell, "Natural language object retrieval," in *CVPR*, 2016.

[89] J. Liu, L. Wang, and M.-H. Yang, "Referring expression generation and comprehension via attributes," in *ICCV*, 2017.

[90] M. Wang, M. Azab, N. Kojima, R. Mihalcea, and J. Deng, "Structured matching for phrase localization," in *ECCV*, 2016.

[91] K. Chen, R. Kovvuri, J. Gao, and R. Nevatia, "MSRC: Multimodal spatial regression with semantic context for phrase grounding," in *ICMR*, 2017.

[92] K. Chen, R. Kovvuri, and R. Nevatia, "Query-guided regression network with context policy for phrase grounding," in *ICCV*, 2017.

[93] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, "Modeling context in referring expressions," in *ECCV*, 2016.

[94] R. A. Yeh, J. Xiong, W. m. W. Hwu, M. N. Do, and A. G. Schwing, "Interpretable and globally optimal prediction for textual grounding using image concepts," in *NIPS*, 2017.

[95] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *TACL*, vol. 2, pp. 67–78, 2014.

[96] M. Hodosh, P. Young, C. Rashtchian, and J. Hockenmaier, "Cross-caption coreference resolution for automatic image understanding," in *CoNLL*. ACL, 2010, pp. 162–171.

[97] W. M. Soon, H. T. Ng, and D. C. Y. Lim, "A machine learning approach to coreference resolution of noun phrases," *Computational Linguistics*, vol. 27, no. 4, pp. 521–544, 2001.

[98] V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei, "Linking people in videos with "their" names using coreference resolution," in *ECCV*, 2014.

[99] J. F. McCarthy and W. G. Lehnert, "Using decision trees for coreference resolution," *arXiv cmp-lg/9505043*, 1995.

[100] H. Su, J. Deng, and L. Fei-Fei, "Crowdsourcing annotations for visual object detection," in *AAAI Technical Report, 4th Human Computation Workshop*, 2012.

[101] A. Sorokin and D. Forsyth, "Utility data annotation with Amazon Mechanical Turk," *Internet Vision Workshop*, 2008.

[102] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[103] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *ECCV*, 2010.

[104] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[105] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[106] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[107] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, "A multi-view embedding space for modeling internet images, tags, and their semantics," *IJCV*, vol. 106, no. 2, pp. 210–233, 2014.

[108] C. L. Zitnick and P. Dollár, "Edge Boxes: Locating object proposals from edges," in *ECCV*, 2014.

[109] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *IJCV*, vol. 104, no. 2, 2013.

[110] L. Wang, Y. Li, and S. Lazebnik, “Learning deep structure-preserving image-text embeddings,” in *CVPR*, 2016.

[111] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy, “Generation and comprehension of unambiguous object descriptions,” in *CVPR*, 2016.

[112] J. Tighe, M. Niethammer, and S. Lazebnik, “Scene parsing with object instances and occlusion ordering,” in *CVPR*, 2014.

[113] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, “Convergence properties of the Nelder-Mead simplex method in low dimensions,” *SIAM Journal of Optimization*, vol. 9, no. 1, p. 112147, 1998.

[114] T. Joachims, “Training linear svms in linear time,” in *SIGKDD*, 2006.

[115] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, “Parsing With Compositional Vector Grammars,” in *ACL*, 2013.

[116] S. Fidler, A. Sharma, and R. Urtasun, “A sentence is worth a thousand pixels,” in *CVPR*, 2013.

[117] S. Harabagiu and S. Maiorano, “Knowledge-lean coreference resolution and its relation to textual cohesion and coherence,” in *Proceedings of the ACL-99 Workshop on the relation of discourse/dialogue structure and reference*, 1999, pp. 29–38.

[118] R. Mitkov, “Robust pronoun resolution with limited knowledge,” in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*, 1998, pp. 869–875.

[119] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, “A multi-view embedding space for modeling internet images, tags, and their semantics,” *IJCV*, vol. 106, no. 2, pp. 210–233, 2014.

[120] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv:1301.3781*, 2013.

[121] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Heber, “An empirical study of context in object detection,” in *CVPR*, 2009.

[122] L.-J. Li, H. Su, Y. Lim, and L. Fei-Fei, “Object bank: An object-level image representation for high-level visual recognition,” *IJCV*, vol. 107, no. 1, pp. 20–39, 2014.

[123] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[124] J. C. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” in *Advances in Large Margin Classifiers.* MIT Press, 1999, pp. 61–74.

[125] F. Sadeghi, S. K. Divvala, and A. Farhadi, "VisKE: Visual knowledge extraction and question answering by visual verification of relation phrases," in *CVPR*, 2015.

[126] A. Veit, S. Belongie, and T. Karaletsos, "Conditional similarity networks," in *CVPR*, 2017.

[127] B. Babenko, S. Branson, and S. Belongie, "Similarity metrics for categorization: from monolithic to category specific," in *ICCV*, 2009.

[128] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[129] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference for Learning Representations*, 2015.

[130] R. Luo and G. Shakhnarovich, "Comprehension-guided referring expressions," in *CVPR*, 2017.

[131] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *CVPR*, 2017.

[132] R. Hinami and S. Satoh, "Query-adaptive R-CNN for open-vocabulary object detection and retrieval," *arXiv:1711.09509*, 2017.

[133] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *CVPR*, 2015.

[134] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[135] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun, "Order-embeddings of images and language," in *ICLR*, 2016.

[136] L. Wang, A. G. Schwing, and S. Lazebnik, "Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space," in *NIPS*, 2017.

[137] Y. Liu, Y. Guo, E. M. Bakker, and M. S. Lew, "Learning a recurrent residual fusion network for multimodal matching," in *ICCV*, 2017.

[138] H. Nam, J.-W. Ha, and J. Kim, "Dual attention networks for multimodal reasoning and matching," in *CVPR*, 2017.

[139] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "VSE++: improving visual-semantic embeddings with hard negatives," *arXiv:1707.05612*, 2017.

[140] Y. Huang, Q. Wu, and L. Wang, "Learning semantic concepts and order for image and sentence matching," *arXiv:1712.02036*, 2017.

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

[142] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *ICCV*, 2017.

[143] A. Li, A. Jabri, A. Joulin, and L. van der Maaten, "Learning visual N-grams from web data," in *ICCV*, 2017.

[144] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 3, no. 1, 2007.

[145] M. Wang, R. Hong, G. Li, Z.-J. Zha, S. Yan, and T.-S. Chua, "Event driven web video summarization by tag localization and key-shot identification," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 975–985, 2012.

[146] B. Xiong, G. Kim, and L. Sigal, "Storyline Representation of Egocentric Videos and Its Applications to Story-based Search," in *ICCV*, 2015.

[147] M. Sun, A. Farhadi, and S. Seitz, "Ranking domain-specific highlights by analyzing edited videos," in *ECCV*, 2014.

[148] W.-S. Chu, Y. Song, and A. Jaimes, "Video co-summarization: Video summarization by visual co-occurrence," in *CVPR*, 2015.

[149] M. Ellouze, N. Boujemaa, and A. M. Alimi, "IM(S)2: interactive movie summarization system," *J. Vis. Comun. Image Represent.*, vol. 21, no. 4, pp. 283–294, 2010.

[150] R. Gomes and A. Krause, "Budgeted Nonparametric Learning from Data Streams," in *ICML*, 2010.

[151] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *ECCV*, 2014.

[152] G. Kim, L. Sigal, and E. P. Xing, "Joint Summarization of Large-scale Collections of Web Images and Videos for Storyline Reconstruction," in *CVPR*, 2014.

[153] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu, "Video summarization via transferrable structured learning," in *WWW*, 2011.

[154] Z. Lu and K. Grauman, "Story-driven summarization for egocentric video," in *CVPR*, 2013.

[155] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," in *ECCV*, 2014.

[156] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *ECCV*, 2016.

[157] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Summary transfer: exemplar-based subset selection for video summarization," in *CVPR*, 2016.

[158] Y. Cong, J. Yuan, and J. Luo, "Towards scalable summarization of consumer videos via sparse dictionary selection," *IEEE Transactions on Multimedia*, vol. 14, no. 1, pp. 66–75, 2012.

[159] J. Kwon and K. M. Lee, "A unified framework for event summarization and rare event detection," in *CVPR*, 2012.

[160] B. Zhao and E. P. Xing, "Quasi real-time summarization for consumer videos," in *CVPR*, 2014.

[161] J. Xu, L. Mukherjee, Y. Li, J. Warner, J. M. Rehg, and V. Singh, "Gaze-enabled egocentric video summarization via constrained submodular maximization," in *CVPR*, 2015.

[162] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan, "Large-scale video summarization using web-image priors," in *CVPR*, 2013.

[163] B. Xiong and K. Grauman, "Detecting snap points in egocentric video with a web photo prior," in *ECCV*, 2014.

[164] M. A. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language," in *International Workshop on Content-Based Access of Image and Video Databases*, 1998.

[165] W. Liu, T. Mei, Y. Zhang, C. Che, and J. Luo, "Multi-task deep visual-semantic embedding for video thumbnail selection," in *CVPR*, 2015.

[166] A. Sharghi, B. Gong, and M. Shah, "Query-focused extractive video summarization," in *ECCV*, 2016.

[167] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "TVSum: Summarizing Web Videos using Titles," in *CVPR*, 2015.

[168] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun, "Order-embeddings of images and language," *ICLR*, 2016.

[169] T.-H. K. Huang, F. Ferraro, N. Mostafazadeh, I. Misra, J. Devlin, A. Agrawal, R. Girshick, X. He, P. Kohli, D. Batra, L. Zitnick, D. Parikh, L. Vanderwende, M. Galley, and M. Mitchell, "Visual storytelling," in *NAACL*, 2016.

[170] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," *ICCV*, 2015.

[171] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *CVPR*, 2012.

[172] S. Yeung, A. Fathi, and L. Fei-Fei, "Videoset: Video summary evaluation through text," *arXiv:1406.5824*, 2014.

[173] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions - I," *Mathematical Programming*, 1978.

[174] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv:1310.1531*, 2013.

[175] C. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proceedings of the ACL-04 workshop. Volume 8.*, 2004.

[176] Y. Li and B. Mérialdo, "Multi-video summarization based on video-MMR," in *WIAMIS*, 2010.

[177] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," in *NIPS*, 2014.

[178] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[179] E. Rahtu, J. Kannala, M. Salo, and J. Heikkil, "Segmenting salient objects from images and videos," in *ECCV*, 2010.

[180] K. Cho, B. van Merrienboer, D. B. Caglar Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP*, 2014.

[181] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, and D. Batra, "Reducing overfitting in deep networks by decorrelating representations," in *ICLR*, 2016.

[182] S. Tellex and D. Roy, "Towards surveillance video search by natural language query," in *ACM International Conference on Image and Video Retrieval*, 2009.

[183] D. Lin, S. Fidler, C. Kong, and R. Urtasun, "Visual semantic search: Retrieving videos via complex textual queries," in *CVPR*, 2014.

[184] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV*, 2016.

[185] V. Ramanishka, A. Das, J. Zhang, and K. Saenko, "Top-down visual saliency guided by captions," in *CVPR*, 2017.