Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books

Yukun Zhu [*,1] Ryan Kiros[*,1] Richard Zemel[1] Ruslan Salakhutdinov[1]
Raquel Urtasun[1] Antonio Torralba[2] Sanja Fidler[1]
[1]University of Toronto [2]Massachusetts Institute of Technology
{yukun,rkiros,zemel,rsalakhu,urtasun,fidler}@cs.toronto.edu, torralba@csail.mit.edu

Abstract

Books are a rich source of both fine-grained information, how a character, an object or a scene looks like, as well as high-level semantics, what someone is thinking, feeling and how these states evolve through a story. This paper aims to align books to their movie releases in order to provide rich descriptive explanations for visual content that go semantically far beyond the captions available in current datasets. To align movies and books we exploit a neural sentence embedding that is trained in an unsupervised way from a large corpus of books, as well as a video-text neural embedding for computing similarities between movie clips and sentences in the book. We propose a context-aware CNN to combine information from multiple sources. We demonstrate good quantitative performance for movie/book alignment and show several qualitative examples that showcase the diversity of tasks our model can be used for.

1. Introduction

A truly intelligent machine needs to not only parse the surrounding 3D environment, but also understand why people take certain actions, what they will do next, what they could possibly be thinking, and even try to empathize with them. In this quest, language will play a crucial role in grounding visual information to high-level semantic concepts. Only a few words in a sentence may convey really rich semantic information. Language also represents a natural means of interaction between a naive user and our vision algorithms, which is particularly important for applications such as social robotics or assistive driving.

Combining images or videos with language has gotten significant attention in the past year, partly due to the creation of CoCo [20], Microsoft's large-scale captioned image dataset. The field has tackled a diverse set of tasks such as captioning [15, 13, 40, 39, 24], alignment [13, 17, 38], Q&A [22, 21], visual model learning from textual descriptions [9, 29], and semantic visual search with natural multi-sentence queries [19].

[*]Denotes equal contribution

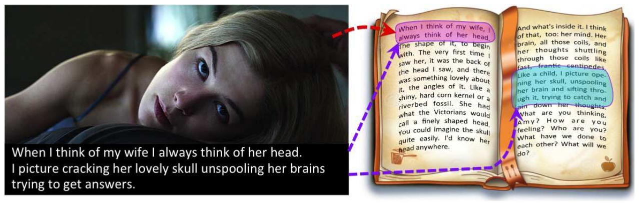

When I think of my wife I always think of her head.
I picture cracking her lovely skull unspooling her brains trying to get answers.
When I think of my wife, I always think of her head. The shape of it, to begin with. The very first time I saw her, it was the back of the head I saw, and there was something lovely about it, the angles of it. Like a shiny, hard corn kernel or a riverbed fossil. She had what the Victorians would call a finely shaped head. You could imagine the skull quite easily. I'd know her head anywhere.
And what's inside it. I think of that, too: her mind. Her brain, all those coils, and her thoughts shuttling through those coils like fast, frantic centipedes. Like a child, I picture opening her skull, unspooling her brain and sifting through it, trying to catch and pin down her thoughts. What are you thinking, Amy? How are you feeling? Who are you? What have we done to each other? What will we do?

Figure 1: Shot from the movie *Gone Girl*, along with the subtitle, aligned with the book. We reason about the visual and dialog (text) alignment between the movie and a book.

Books provide us with very descriptive text that conveys both fine-grained visual details (how things look like) as well as high-level semantics (what people think, feel, and how their states evolve through a story). This source of knowledge, however, does not come with associated visual information that would enable us to ground it with natural language. Grounding descriptions in books to vision would allow us to get textual *explanations* or stories about the visual world rather than short *captions* available in current datasets. It could also provide us with a very large amount of data (with tens of thousands books available online).

In this paper, we exploit the fact that many books have been turned into movies. Books and their movie releases have a lot of common knowledge as well as they are complementary in many ways. For instance, books provide detailed descriptions about the intentions and mental states of the characters, while movies are better at capturing visual aspects of the settings.

The first challenge we need to address, and the focus of this paper, is to align books with their movie releases in order to obtain rich descriptions for the visual content. We aim to align the two sources with two types of information: *visual*, where the goal is to link a movie shot to a book paragraph, and *dialog*, where we want to find correspondences between sentences in the movie's subtitle and sentences in the book (Fig. 1). We introduce a novel sentence similarity measure based on a neural sentence embedding trained on millions of sentences from a large corpus of books. On the visual side, we extend the neural image-sentence embeddings to the video domain and train the model on DVS descriptions of movie clips. Our approach combines different similarity measures and takes into account contextual information contained in the nearby shots and book sentences. Our final alignment model is formulated as an energy minimization problem that encourages the alignment to follow a similar timeline. To evaluate the book-movie alignment model we collected a dataset with 11 movie/book pairs annotated with 2,070 shot-to-sentence correspondences. We demonstrate good quantitative performance and show several qualitative examples that showcase the diversity of tasks our model can be used for. All our data and code are available: http://www.cs.utoronto.ca/~mbweb/.

The alignment model enables multiple applications. Imagine an app which allows the user to browse the book as the scenes unroll in the movie: perhaps its ending or acting are ambiguous, and one would like to query the book for answers. Vice-versa, while reading the book one might want to switch from text to video, particularly for the juicy scenes. We also show other applications of learning from movies and books such as book retrieval (finding the book that goes with a movie and finding other similar books), and captioning CoCo images with story-like descriptions.

2. Related Work

Most effort in the domain of vision and language has been devoted to the problem of image captioning. Older work made use of fixed visual representations and translated them into textual descriptions [7, 18]. Recently, several approaches based on RNNs emerged, generating captions via a learned joint image-text embedding [15, 13, 40, 24]. These approaches have also been extended to generate descriptions of short video clips [39]. In [27], the authors go beyond describing *what* is happening in an image and provide explanations about *why* something is happening. Related to ours is also work on image retrieval [11] which aims to find an image that best depicts a complex description.

For text-to-image alignment, [17, 8] find correspondences between nouns and pronouns in a caption and visual objects using several visual and textual potentials. Lin *et al*. [19] does so for videos. In [23], the authors align cooking videos with the recipes. Bojanowski et al. [2] localize actions from an ordered list of labels in video clips. In [13, 34], the authors use RNN embeddings to find the correspondences. [41] combines neural embeddings with soft attention in order to align the words to image regions.

Early work on movie-to-text alignment include dynamic time warping for aligning movies to scripts with the help of subtitles [6, 5]. Sankar *et al*. [31] further developed a system which identified sets of visual and audio features to align movies and scripts without making use of the subtitles. Such alignment has been exploited to provide weak labels for person naming tasks [6, 33, 28].

Closest to our work is [38], which aligns plot synopses to shots in the TV series for story-based content retrieval. This work adopts a similarity function between sentences in plot synopses and shots based on person identities and keywords in subtitles. Our work differs with theirs in several important aspects. First, we tackle a more challenging problem of movie/book alignment. Unlike plot synopsis, which closely follow the storyline of movies, books are more verbose and might vary in the storyline from their movie release. Furthermore, we use learned neural embeddings to compute the similarities rather than hand-designed similarity functions.

Parallel to our work, [37] aims to align scenes in movies to chapters in the book. However, their approach operates on a very coarse level (chapters), while ours does so on the sentence/paragraph level. Their dataset thus evaluates on 90 scene-chapter correspondences, while our dataset draws 1,800 shot-to-paragraph alignments. Furthermore, the approaches are inherently different. [37] matches the presence of characters in a scene to those in a chapter, as well as uses hand-crafted similarity measures between sentences in the subtitles and dialogs in the books, similarly to [38].

Rohrbach *et al*. [30] recently released the Movie Description dataset which contains clips from movies, each time-stamped with a sentence from DVS (Descriptive Video Service). The dataset contains clips from over a 100 movies, and provides a great resource for the captioning techniques. Our effort here is to align movies with books in order to obtain longer, richer and more high-level video descriptions.

We start by describing our new dataset, and then explain our proposed approach.

3. The MovieBook and BookCorpus Datasets

We collected two large datasets, one for movie/book alignment and one with a large number of books.

The MovieBook Dataset. Since no prior work or data exist on the problem of movie/book alignment, we collected a new dataset with 11 movies and corresponding books. For each movie we also have subtitles, which we parse into a set of time-stamped sentences. Note that no speaker information is provided in the subtitles. We parse each book into sentences and paragraphs.

Our annotators had the movie and a book opened side by side. They were asked to iterate between browsing the book and watching a few shots/scenes of the movie, and trying to find correspondences between them. In particular, they marked the exact time (in seconds) of correspondence in the movie and the matching line number in the book file, indicating the beginning of the matched sentence. On the video side, we assume that the match spans across a *shot* (a video unit with smooth camera motion). If the match was longer in duration, the annotator also indicated the ending time. Similarly for the book, if more sentences matched, the annotator indicated from which to which line a match occurred. Each alignment was tagged as a *visual*, *dialogue*, or an *audio match*. Note that even for dialogs, the movie and book versions are semantically similar but not exactly

	BOOK						MOVIE		ANNOTATION	
Title	# sent.	# words	# unique words	avg. # words per sent.	max # words per sent.	# paragraphs	# shots	# sent. in subtitles	# dialog align.	# visual align.
Gone Girl	12,603	148,340	3,849	15	153	3,927	2,604	2,555	76	106
Fight Club	4,229	48,946	1,833	14	90	2,082	2,365	1,864	104	42
No Country for Old Men	8,050	69,824	1,704	10	68	3,189	1,348	889	223	47
Harry Potter and the Sorcerers Stone	6,458	78,596	2,363	15	227	2,925	2,647	1,227	164	73
Shawshank Redemption	2,562	40,140	1,360	18	115	637	1,252	1,879	44	12
The Green Mile	9,467	133,241	3,043	17	119	2,760	2,350	1,846	208	102
American Psycho	11,992	143,631	4,632	16	422	3,945	1,012	1,311	278	85
One Flew Over the Cuckoo Nest	7,103	112,978	2,949	19	192	2,236	1,671	1,553	64	25
The Firm	15,498	135,529	3,685	11	85	5,223	2,423	1,775	82	60
Brokeback Mountain	638	10,640	470	20	173	167	1,205	1,228	80	20
The Road	6,638	58,793	1,580	10	74	2,345	1,108	782	126	49
All	85,238	980,658	9,032	15	156	29,436	19,985	16,909	1,449	621

Table 1: Statistics for our **MovieBook Dataset** with ground-truth for alignment between books and their movie releases.

# of books	# of sentences	# of words	# of unique words	mean # of words per sentence	median # of words per sentence
11,038	74,004,228	984,846,357	1,316,420	13	11

Table 2: Summary statistics of our **BookCorpus** dataset. We use this corpus to train the sentence embedding model.

the same. Thus deciding on what defines a match or not is also somewhat subjective and may slightly vary across our annotators. Altogether, the annotators spent 90 hours labeling 11 movie/book pairs, locating 2,070 correspondences.

Table 1 presents our dataset, while Fig. 6 shows a few ground-truth alignments. The number of sentences per book vary from 638 to 15,498, even though the movies are similar in duration. This indicates a huge diversity in descriptiveness across literature, and presents a challenge for matching. Sentences also vary in length, with those in Brokeback Mountain being twice as long as those in The Road. The longest sentence in American Psycho has 422 words and spans over a page in the book.

Aligning movies with books is challenging even for humans, mostly due to the scale of the data. Each movie is on average 2h long and has 1,800 shots, while a book has on average 7,750 sentences. Books also have different styles of writing, formatting, language, may contain slang (*going* vs *goin'*, or even *was* vs *'us*), etc. Table 1 shows that finding visual matches was particularly challenging. This is because descriptions in books can be either very short and hidden within longer paragraphs or even within a longer sentence, or very verbose – in which case they get obscured with the surrounding text – and are hard to spot. Of course, how close the movie follows the book is also up to the director, which can be seen through the number of alignments that our annotators found across different movie/books.

BookCorpus. In order to train our sentence similarity model we collected a corpus of 11,038 books from the web. These are free books written by yet unpublished authors. We only included books that had more than 20K words in order to filter out perhaps noisier shorter stories. The dataset has books in 16 different genres, e.g., *Romance* (2,865 books), *Fantasy* (1,479), *Science fiction* (786), etc. Table 2 highlights the summary statistics of our corpus.

4. Aligning Books and Movies

Our approach aims to align a movie with a book by exploiting visual information as well as dialogs. We take shots as video units and sentences from subtitles to represent dialogs. Our goal is to match these to the sentences in the book. We propose several measures to compute similarities between pairs of sentences as well as shots and sentences. We use our novel deep neural embedding trained on our large corpus of books to predict similarities between sentences. Note that an extended version of the sentence embedding is described in detail in [16] showing how to deal with million-word vocabularies, and demonstrating its performance on a large variety of NLP benchmarks. For comparing shots with sentences we extend the neural embedding of images and text [15] to operate in the video domain. We next develop a novel contextual alignment model that combines information from various similarity measures and a larger time-scale in order to make better local alignment predictions. Finally, we propose a simple pairwise Conditional Random Field (CRF) that smooths the alignments by encouraging them to follow a linear timeline, both in the video and book domain.

We first explain our sentence, followed by our joint video to text embedding. We next propose our contextual model that combines similarities and discuss CRF in more detail.

4.1. Skip-Thought Vectors

In order to score the similarity between two sentences, we exploit our architecture for learning unsupervised representations of text [16]. The model is loosely inspired by the skip-gram [25] architecture for learning representations of words. In the word skip-gram model, a word w_i is chosen and must predict its surrounding context (e.g. w_{i+1} and w_{i-1} for a context window of size 1). Our model works in a similar way but at the sentence level. That is, given a sentence tuple (s_{i-1}, s_i, s_{i+1}) our model first encodes the sentence s_i into a fixed vector, then conditioned on this vector tries to reconstruct the sentences s_{i-1} and s_{i+1}, as shown in Fig. 2. The motivation for this architecture is inspired by the distributional hypothesis: sentences that have similar surrounding context are likely to be both semantically and syntactically similar. Thus, two sentences that have similar

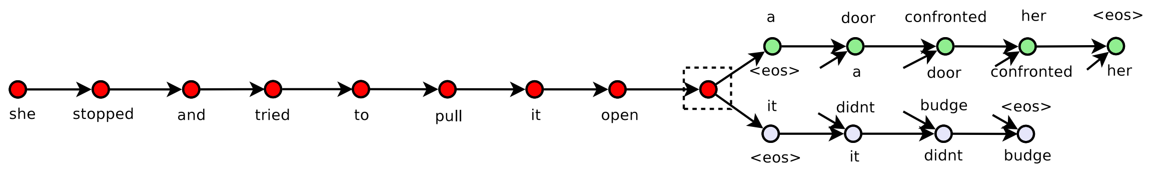


Figure 2: Sentence neural embedding [16]. Given a tuple (s_{i-1}, s_i, s_{i+1}) of contiguous sentences, where s_i is the i-th sentence of a book, the sentence s_i is encoded and tries to reconstruct the previous sentence s_{i-1} and next sentence s_{i+1}. Unattached arrows are connected to the encoder output. Colors indicate which components share parameters. $\langle$eos$\rangle$ is the end of sentence token.

he drove down the street off into the distance .	he started the car , left the parking lot and merged onto the highway a few miles down the road . he shut the door and watched the taxi drive off . she watched the lights flicker through the trees as the men drove toward the road .
the most effective way to end the battle .	a messy business to be sure , but necessary to achieve a fine and noble end . they saw their only goal as survival and logically planned a strategy to achieve it . there would be far fewer casualties and far less destruction .

Table 3: Qualitative results from the sentence skip-gram model. For each query sentence on the left, we retrieve the 4 nearest neighbor sentences (by inner product) chosen from books the model has not seen before. More results in supplementary.

syntax and semantics are likely to be encoded to a similar vector. Once the model is trained, we can map any sentence through the encoder to obtain vector representations, then score their similarity through an inner product.

The learning signal of the model depends on having contiguous text, where sentences follow one another in sequence. A natural corpus for training our model is thus a large collection of books. Given the size and diversity of genres, our BookCorpus allows us to learn very general representations of text. For instance, Table 3 illustrates the nearest neighbours of query sentences, taken from held out books that the model was not trained on. These qualitative results demonstrate that our intuition is correct, with resulting nearest neighbors corresponds largely to syntactically and semantically similar sentences. Note that the sentence embedding is general and can be applied to other domains not considered in this paper, which is explored in [16].

To construct an encoder, we use a recurrent neural network, inspired by the success of encoder-decoder models for neural machine translation [12, 3, 1, 35]. Two kinds of activation functions have recently gained traction: long short-term memory (LSTM) [10] and the gated recurrent unit (GRU) [4]. Both types of activation successfully solve the vanishing gradient problem, through the use of gates to control the flow of information. The LSTM unit explicity employs a cell that acts as a carousel with an identity weight. The flow of information through a cell is controlled by input, output and forget gates which control what goes into a cell, what leaves a cell and whether to reset the contents of the cell. The GRU does not use a cell but employs two gates: an update and a reset gate. In a GRU, the hidden state is a linear combination of the previous hidden state and the proposed hidden state, where the combination weights are controlled by the update gate. GRUs have been shown to perform just as well as LSTM on several sequence prediction tasks [4] while being simpler. Thus, we use GRU as the activation function for our encoder and decoder RNNs.

Suppose we are given a sentence tuple (s_{i-1}, s_i, s_{i+1}), and let w_i^t denote the t-th word for s_i and let $\mathbf{x}_i^t$ be its word embedding. We break the model description into three parts: the encoder, decoder and objective function.

Encoder. Let $w_i^1, \ldots, w_i^N$ denote words in sentence s_i with N the number of words in the sentence. The encoder produces a hidden state $\mathbf{h}_i^t$ at each time step which forms the representation of the sequence $w_i^1, \ldots, w_i^t$. Thus, the hidden state $\mathbf{h}_i^N$ is the representation of the whole sentence. The GRU produces the next hidden state as a linear combination of the previous hidden state and the proposed state update (we drop subscript i):

$$\mathbf{h}^t = (1 - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t \tag{1}$$

where $\bar{\mathbf{h}}^t$ is the proposed state update at time t, $\mathbf{z}^t$ is the update gate and $(\odot)$ denotes a component-wise product. The update gate takes values between zero and one. In the extreme cases, if the update gate is the vector of ones, the previous hidden state is completely forgotten and $\mathbf{h}^t = \bar{\mathbf{h}}^t$. Alternatively, if the update gate is the zero vector, than the hidden state from the previous time step is simply copied over, that is $\mathbf{h}^t = \mathbf{h}^{t-1}$. The update gate is computed as

$$\mathbf{z}^t = \sigma(\mathbf{W}_z \mathbf{x}^t + \mathbf{U}_z \mathbf{h}^{t-1}) \tag{2}$$

where $\mathbf{W}_z$ and $\mathbf{U}_z$ are the update gate parameters. The proposed state update is given by

$$\bar{\mathbf{h}}^t = \tanh(\mathbf{W}\mathbf{x}^t + \mathbf{U}(\mathbf{r}^t \odot \mathbf{h}^{t-1})) \tag{3}$$

where $\mathbf{r}_t$ is the reset gate, which is computed as

$$\mathbf{r}^t = \sigma(\mathbf{W}_r \mathbf{x}^t + \mathbf{U}_r \mathbf{h}^{t-1}) \tag{4}$$

If the reset gate is the zero vector, than the proposed state update is computed only as a function of the current word. Thus after iterating this equation sequence for each word, we obtain a sentence vector $\mathbf{h}_i^N = \mathbf{h}_i$ for sentence s_i.

Decoder. The decoder computation is analogous to the encoder, except that the computation is conditioned on the sentence vector $\mathbf{h}_i$. Two separate decoders are used, one for the previous sentence s_{i-1} and one for the next sentence s_{i+1}. These decoders use different parameters to compute their hidden states but both share the same vocabulary matrix $\mathbf{V}$ that takes a hidden state and computes a distribution over words. Thus, the decoders are analogous to an RNN language model but conditioned on the encoder sequence. Alternatively, in the context of image caption generation, the encoded sentence $\mathbf{h}_i$ plays a similar role as the image.

We describe the decoder for the next sentence s_{i+1} (computation for s_{i-1} is identical). Let $\mathbf{h}_{i+1}^t$ denote the hidden state of the decoder at time t. The update and reset gates for the decoder are given as follows (we drop $i+1$):

$$\mathbf{z}^t = \sigma(\mathbf{W}_z^d \mathbf{x}^{t-1} + \mathbf{U}_z^d \mathbf{h}^{t-1} + \mathbf{C}_z \mathbf{h}_i) \quad (5)$$

$$\mathbf{r}^t = \sigma(\mathbf{W}_r^d \mathbf{x}^{t-1} + \mathbf{U}_r^d \mathbf{h}^{t-1} + \mathbf{C}_r \mathbf{h}_i) \quad (6)$$

the hidden state $\mathbf{h}_{i+1}^t$ is then computed as:

$$\bar{\mathbf{h}}^t = \tanh(\mathbf{W}^d \mathbf{x}^{t-1} + \mathbf{U}^d(\mathbf{r}^t \odot \mathbf{h}^{t-1}) + \mathbf{C}\mathbf{h}_i) \quad (7)$$

$$\mathbf{h}_{i+1}^t = (1 - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t \quad (8)$$

Given $\mathbf{h}_{i+1}^t$, the probability of word w_{i+1}^t given the previous $t-1$ words and the encoder vector is

$$P(w_{i+1}^t | w_{i+1}^{<t}, \mathbf{h}_i) \propto \exp(\mathbf{v}_{w_{i+1}^t} \mathbf{h}_{i+1}^t) \quad (9)$$

where $\mathbf{v}_{w_{i+1}^t}$ denotes the row of $\mathbf{V}$ corresponding to the word of w_{i+1}^t. An analogous computation is performed for the previous sentence s_{i-1}.

Objective. Given (s_{i-1}, s_i, s_{i+1}), the objective optimized is the sum of log-probabilities for the next and previous sentences conditioned on the representation of the encoder:

$$\sum_t \log P(w_{i+1}^t | w_{i+1}^{<t}, \mathbf{h}_i) + \sum_t \log P(w_{i-1}^t | w_{i-1}^{<t}, \mathbf{h}_i)$$

The total objective is the above summed over all such training tuples. Adam algorithm [14] is used for optimization.

4.2. Visual-semantic embeddings of clips and DVS

The model above describes how to obtain a similarity score between two sentences, whose representations are learned from millions of sentences in books. We now discuss how to obtain similarities between shots and sentences.

Our approach closely follows the image-sentence ranking model proposed by [15]. In their model, an LSTM is used for encoding a sentence into a fixed vector. A linear mapping is applied to image features from a convolutional network. A score is computed based on the inner product between the normalized sentence and image vectors. Correct image-sentence pairs are trained to have high score, while incorrect pairs are assigned low scores.

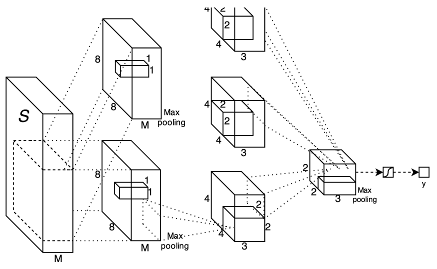


Figure 3: Our CNN for context-aware similarity computation. It has 3 conv. layers and a sigmoid layer on top.

In our case, we learn a visual-semantic embedding between movie clips and their DVS description. DVS ("Descriptive Video Service") is a service that inserts audio descriptions of the movie between the dialogs in order to enable the visually impaired to follow the movie like anyone else. We used the movie description dataset of [30] for learning our embedding. This dataset has 94 movies, and 54,000 described clips. We represent each movie clip as a vector corresponding to mean-pooled features across each frame in the clip. We used the GoogLeNet architecture [36] as well as hybrid-CNN [42] for extracting frame features. For DVS, we pre-processed the descriptions by removing names and replacing these with a *someone* token.

The LSTM architecture in this work is implemented using the following equations. As before, we represent a word embedding at time t of a sentence as $\mathbf{x}^t$:

$$\mathbf{i}^t = \sigma(\mathbf{W}_{xi}\mathbf{x}^t + \mathbf{W}_{hi}\mathbf{m}^{t-1} + \mathbf{W}_{ci}\mathbf{c}^{t-1}) \quad (10)$$

$$\mathbf{f}^t = \sigma(\mathbf{W}_{xf}\mathbf{x}^t + \mathbf{W}_{hf}\mathbf{m}^{t-1} + \mathbf{W}_{cf}\mathbf{c}^{t-1}) \quad (11)$$

$$\mathbf{a}^t = \tanh(\mathbf{W}_{xc}\mathbf{x}^t + \mathbf{W}_{hc}\mathbf{m}^{t-1}) \quad (12)$$

$$\mathbf{c}^t = \mathbf{f}^t \odot \mathbf{c}^{t-1} + \mathbf{i}^t \odot \mathbf{a}^t \quad (13)$$

$$\mathbf{o}^t = \sigma(\mathbf{W}_{xo}\mathbf{x}^t + \mathbf{W}_{ho}\mathbf{m}^{t-1} + \mathbf{W}_{co}\mathbf{c}^t) \quad (14)$$

$$\mathbf{m}^t = \mathbf{o}^t \odot \tanh(\mathbf{c}^t) \quad (15)$$

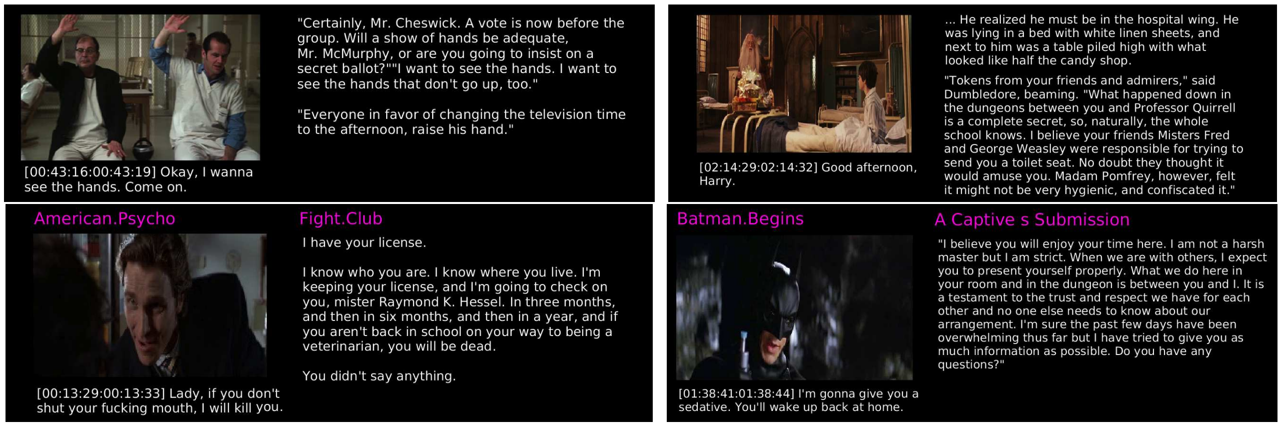
where (σ) denotes the sigmoid activation function and $(\odot)$ indicates component-wise multiplication. The states $(\mathbf{i}^t, \mathbf{f}^t, \mathbf{c}^t, \mathbf{o}^t, \mathbf{m}^t)$ correspond to input, forget, cell, output and memory vectors, respectively. If sentence is N long, then $\mathbf{m}^N = \mathbf{m}$ is the vector representation of the sentence.

Let $\mathbf{q}$ denote a movie clip vector, and let $\mathbf{v} = \mathbf{W}_I\mathbf{q}$ be the embedding of the movie clip. We define a scoring function $s(\mathbf{m}, \mathbf{v}) = \mathbf{m} \cdot \mathbf{v}$, where $\mathbf{m}$ and $\mathbf{v}$ are first scaled to have unit norm (making s equivalent to cosine similarity). We then optimize the following pairwise ranking loss which sums over the training $(\mathbf{m}, \mathbf{v})$ pairs:

$$\min_{\boldsymbol{\theta}} \sum_{\mathbf{m}} \sum_k \max\{0, \alpha - s(\mathbf{m}, \mathbf{v}) + s(\mathbf{m}, \mathbf{v}_k)\} \quad (16)$$

$$+ \sum_{\mathbf{v}} \sum_k \max\{0, \alpha - s(\mathbf{v}, \mathbf{m}) + s(\mathbf{v}, \mathbf{m}_k)\}, \quad (17)$$

with $\mathbf{m}_k$ a contrastive (non-descriptive) sentence vector for a clip embedding $\mathbf{v}$, and vice-versa with $\mathbf{v}_k$. We train our model with stochastic gradient descent without momentum.



[00:43:16:00:43:19] Okay, I wanna see the hands. Come on.

"Certainly, Mr. Cheswick. A vote is now before the group. Will a show of hands be adequate, Mr. McMurphy, or are you going to insist on a secret ballot?""I want to see the hands. I want to see the hands that don't go up, too."

"Everyone in favor of changing the television time to the afternoon, raise his hand."

[02:14:29:02:14:32] Good afternoon, Harry.

... He realized he must be in the hospital wing. He was lying in a bed with white linen sheets, and next to him was a table piled high with what looked like half the candy shop.

"Tokens from your friends and admirers," said Dumbledore, beaming. "What happened down in the dungeons between you and Professor Quirrell is a complete secret, so, naturally, the whole school knows. I believe your friends Misters Fred and George Weasley were responsible for trying to send you a toilet seat. No doubt they thought it would amuse you. Madam Pomfrey, however, felt it might not be very hygienic, and confiscated it."

[00:13:29:00:13:33] Lady, if you don't shut your fucking mouth, I will kill you.

I have your license.

I know who you are. I know where you live. I'm keeping your license, and I'm going to check on you, mister Raymond K. Hessel. In three months, and then in six months, and then in a year, and if you aren't back in school on your way to being a veterinarian, you will be dead.

You didn't say anything.

[01:38:41:01:38:44] I'm gonna give you a sedative. You'll wake up back at home.

"I believe you will enjoy your time here. I am not a harsh master but I am strict. When we are with others, I expect you to present yourself properly. What we do here in your room and in the dungeon is between you and I. It is a testament to the trust and respect we have for each other and no one else needs to know about our arrangement. I'm sure the past few days have been overwhelming thus far but I have tried to give you as much information as possible. Do you have any questions?"

Figure 4: Describing movie clips via the book: **(top)** we align the movie to the book, and show a shot and its corresponding paragraph (plus one before and after) from the book, **(bottom)** for a shot we retrieve the best matching paragraph from a corpus of books.

4.3. Context aware similarity

We employ the clip-sentence embedding to compute similarities between each shot in the movie and each sentence in the book. For dialogs, we use several similarity measures each capturing a different level of semantic similarity. We compute BLEU [26] between each subtitle and book sentence to identify nearly identical matches. Similarly to [38], we use a tf-idf measure to find near duplicates but weighing down the influence of the less frequent words. Finally, we use our sentence embedding learned from books to score pairs of sentences that are semantically similar but may have a very different wording (i.e., paraphrasing).

These similarity measures indicate the alignment between the two modalities. However, at the local, sentence level, alignment can be rather ambiguous. For example, despite being a rather dark book, *Gone Girl* contains 15 occurrences of the sentence "I love you". We exploit the fact that a match is not completely isolated but that the sentences (or shots) around it are also to some extent similar.

We design a context aware similarity measure that takes into account all individual similarity measures as well as a fixed context window in both, the movie and book domain, and predicts a new similarity score. We stack a set of M similarity measures into a tensor $S(i,j,m)$, where i, j, and m are the indices of sentences in the subtitle, in the book, and individual similarity measures, respectively. In particular, we use $M = 9$ similarities: visual and sentence embedding, BLEU1-5, tf-idf, and a uniform prior. We want to predict a combined score $\mathrm{score}(i,j) = f(S(\mathbf{I},\mathbf{J},\mathbf{M}))$ at each location (i,j) based on all measurements in a fixed volume defined by a context region $\mathbf{I}$ around i, $\mathbf{J}$ around j, and $\mathbf{M} = 1,\ldots,M$. Evaluating the function $f(\cdot)$ at each location (i,j) on a 3-D tensor S is very similar to applying a convolution using a kernel of appropriate size. This motivates us to formulate the function $f(\cdot)$ as a deep convolutional neural network (CNN). In this paper, we adopt a 3-layer CNN as illustrated in Fig. 3. We use ReLU nonlinearity with dropout to regularize our model. We optimize the cross-entropy loss over the training set using Adam algorithm. Positive examples are from ground truth and we do hard-negative mining for negative examples (chosen from shot-sentence pairs that are far away from GT) in training.

4.4. Global Movie/Book Alignment

So far, each shot/sentence was matched independently. However, most shots in movies and passages in the books follow a similar timeline. We would like to incorporate this prior into our alignment. In [38], the authors use dynamic time warping by enforcing that the shots in the movie can only match forward in time (to plot synopses in their case). However, the storyline of the movie and book can have crossings in time (Fig. 6), and the alignment might contain giant leaps forwards or backwards. Therefore, we formulate a movie/book alignment problem as inference in a Conditional Random Field that encourages nearby shots/dialog alignments to be consistent. Each node y_i in our CRF represents an alignment of the shot in the movie with its corresponding subtitle sentence to a sentence in the book. Its state space is thus the set of all sentences in the book. The CRF energy of a configuration $\mathbf{y}$ is formulated as:

$$E(\mathbf{y}) = \sum_{i=1}^{K} \omega_u \phi_u(y_i) + \sum_{i=1}^{K} \sum_{j \in \mathcal{N}(i)} \omega_p \psi_p(y_i, y_j)$$

where K is the number of nodes (shots), and $\mathcal{N}(i)$ the left and right neighbor of y_i. Here, $\phi_u(\cdot)$ and $\psi_p(\cdot)$ are unary and pairwise potentials, respectively, and $\omega = (\omega_u, \omega_p)$. We directly use the output of the CNN from 4.3 as the unary potential $\phi_u(\cdot)$. For the pairwise potential, we measure the time span $d_s(y_i, y_j)$ between two neighboring sentences in the subtitle and the distance $d_b(y_i, y_j)$ of their state space in the book. One pairwise potential is defined as:

$$\psi_p(s_i, s_j) = \frac{(d_s(s_i,s_j) - d_b(s_i,s_j))^2}{(d_s(s_i,s_j) - d_b(s_i,s_j))^2 + \sigma^2} \qquad (18)$$

Here σ^2 is a robustness parameter to avoid punishing giant leaps too harsh. Both d_s and d_b are normalized to

[0, 1]. In addition, we also employ another pairwise potential $\psi_q(s_i, s_j) = \frac{(d_b(s_i,s_j))^2}{(d_b(s_i,s_j))^2+\sigma^2}$ to encourage state consistency between nearby sentence nodes. This potential is helpful when there is a long silence (no dialog) in the movie.

Inference. Our CRF is a chain, thus exact inference is possible using dynamic programming. We also prune some states that are very far from the uniform alignment (over $1/3$ length of the book) to further speed up computation.

Learning. Since ground-truth is only available for a sparse set of shots, we regard the states of unobserved nodes as hidden variables and learn the CRF weights with [32].

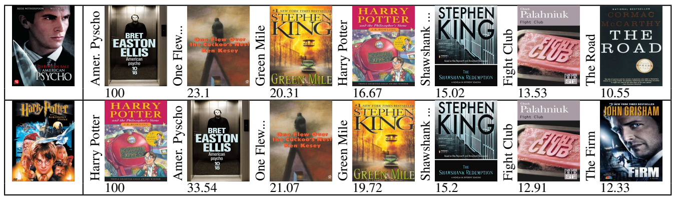
5. Experimental Evaluation

We evaluate our model on our dataset of 11 movie/book pairs. We train the parameters in our model (CNN and CRF) on *Gone Girl*, and test our performance on the remaining 10 movies. In terms of training speed, our video-text model "watches" 1,440 movies per day and our sentence model reads 870 books per day. We also show various qualitative results demonstrating the power of our approach.

5.1. Movie/Book Alignment

Evaluating the performance of movie/book alignment is an interesting problem on its own. This is because our ground-truth is far from exhaustive – around 200 correspondences were typically found between a movie and its book, and likely a number of them got missed. Thus, evaluating the precision is rather tricky. We thus focus our evaluation on recall, similar to existing work on retrieval. For each shot that has a GT correspondence in book, we check whether our prediction is close to the annotated one. We evaluate recall at the paragraph level, i.e., we say that the GT paragraph was recalled, if our match was at most 3 paragraphs away, and the shot was at most 5 subtitle sentences away. As a noisier measure, we also compute recall and precision at multiple alignment thresholds and report AP (avg. prec.).

Results are shown in Table 5. Columns show different instantiations of our model: we show the leave-one-feature-out setting ($\emptyset$ indicates that all features were used), compare how different depths of the context-aware CNN influence performance, and compare it to our full model (CRF) in the last column. We get the highest boost with a deeper CNN – recall improves by 10%, AP doubles. Generally, each feature helps performance. Our sentence embedding (SENT) helps by 5% while video-text embedding (VIS) helps by 4% in recall. CRF which encourages temporal smoothness generally helps, bringing additional 1%. Replacing our sentence embedding by mean-pooling over word2vec features [25] decreases recall by 3.2%. To evaluate our contextual model, we train a linear SVM that combines all similarity measures but ignores context (SVM). Compared with 1-layer CNN, SVM gets a slightly better AP but much



American Psycho (movie)	Amer. Pyscho 100	One Flew... 23.1	Green Mile 20.31	Harry Potter 16.67	Shawshank ... 15.02	Fight Club 13.53	The Road 10.55
Harry Potter (movie)	Harry Potter 100	Amer. Pyscho 33.54	One Flew... 21.07	Green Mile 19.72	Shawshank ... 15.2	Fight Club 12.91	The Firm 12.33

Table 4: **Book "retrieval" results**. For a **movie** (left), we rank books wrt to their alignment similarity with the movie.

worse recall, and our 3 layer CNN outperforms SVM by a large margin. We also show how a uniform timeline (UNI) performs on its own. That is, for each shot in the movie, we find the sentence at the same location (measured in lines) in the book. We plot the alignment for a few movies in Fig. 6.

5.2. Describing Movies via the Book

We next show qualitative results of our alignment. In particular, we run our model on each movie/book pair, and visualize the passage in the book that a particular shot in the movie aligns to. The results are shown in Fig. 4. We can also caption movies by matching shots to paragraphs in a corpus of books, shown in Fig. 4. Here we do not encourage a linear timeline (CRF) since the stories are unrelated, and we only match at the local, shot-paragraph level.

5.3. Book "Retrieval"

In this experiment, we compute alignment between a movie and all (test) 9 books, and check whether our model retrieves the correct book. We achieve perfect performance. Results for two movies are in Table 4, while Suppl. mat. shows results for all movies. Under each book we show the computed similarity. We use the energy from the CRF, and scale all similarities relative to the highest one (100). Interestingly, the second ranked book for *American Psycho* is *Cuckoo's Nest*, both dark movies.

5.4. The CoCoBook: Writing Stories for CoCo

Our next experiment shows that our model is able to "generate" descriptive stories for images. In particular, we used the image-text embedding from [15] and generated a simple caption for an image. We used this caption as a query, and used our sentence embedding to find top 10 nearest sentences from our BookCorpus. We re-ranked these based on the 1-gram precision of non-stop words. We show the best sentence as well as 2 sentences before and after it in the book. Results are in Fig. 5. We are able to retrieve semantically meaningful *stories* to explain the images.

6. Conclusion

We explored a new problem of movie-book alignment. We proposed an approach that computes several similarities between shots and dialogs and sentences in the book. We proposed a novel sentence embedding trained unsupervised from a large corpus of books, used to compute similarities between sentences. We extended the image-text neural embeddings to video, and proposed a context-aware alignment

the club was a little emptier than i would have expected for the late afternoon , and the bartender , in red waistcoat and bowtie , was busy wiping down his counter , replacing peanuts and putting out new coasters . a television with the latest la liga news was hung in an upper corner , and behind him , rows of bottles were reflected in a giant bar mirror . above the stools , a pergola-type overhead structure held rows of wine glasses . it was a classy place , with ferns in the corner , and not the kind of bar to which i was accustomed . my places usually had a more ... relaxed feel .

he felt like an idiot for yelling at the child , but his frustration and trepidation was getting the better of him . he glanced toward the shadowed hall and quickly nodded toward melissa before making his way forward . he came across more children sitting upon a couch in the living room . they watched him , but did n't move and did n't speak . his skin started to feel like hundreds of tiny spiders were running up and down it and he hurried on .

a few miles before tioga road reached highway 395 and the town of lee vining , smith turned onto a narrow blacktop road . on either side were parched , grassy open slopes with barbed-wire fences marking property lines . cattle and horses grazed under trees whose black silhouettes stood stark against the gold-velvet mountains . marty burst into song : "home , home on the range , where the deer and the antelope play ! where seldom is heard a discouraging word and the skies are not cloudy all day!"

"number seventy-three, second to last from the corner. adam slowed the porsche as he approached the quaint-he could think of no other word to use , even though "quaint" was one he normally, manfully, avoided-townhouse, coming to a halt beside a sleek jaguar sedan. it was a quiet street, devoid of traffic at this hour on a monday night. in the bluish-tinted light of a corner street lamp, he developed a quick visual impression of wrought-iron railings on tidy front stoops, window boxes full of bright chrysanthemums, beveled glass in bay windows , and lace curtains. townhouses around here didn't rent cheaply, he couldn't help but observe .

Figure 5: CoCoBook: We generate a caption for a CoCo image via [15] and retrieve its best matched sentence (+ 2 before and after) from a large book corpus. One can see a semantic relevance of the retrieved passage to the image.

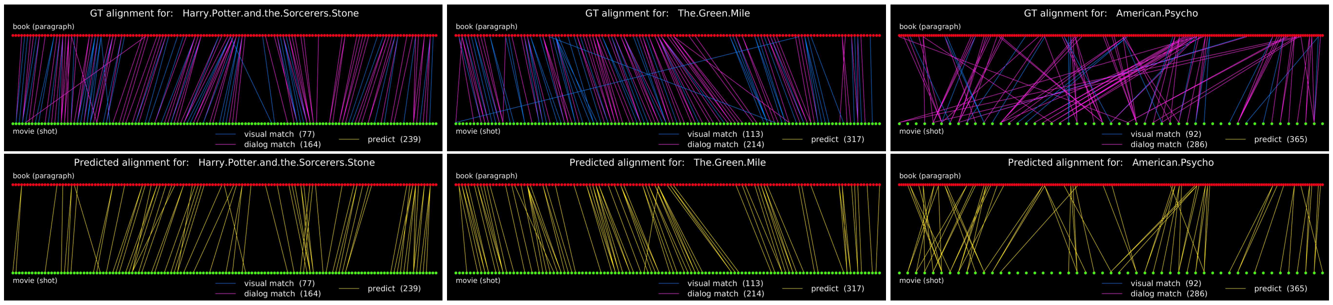


Figure 6: Alignment results of our model compared to ground-truth alignment.

		UNI	SVM	1 layer CNN w/o one feature							CNN-3	CRF
				∅	BLEU	TF-IDF	SENT	VIS	SCENE	PRIOR		
Fight Club	AP	1.22	0.73	0.45	0.41	0.40	0.50	0.64	0.50	0.48	1.95	**5.17**
	Recall	2.36	10.38	12.26	12.74	11.79	11.79	12.74	11.79	11.79	17.92	**19.81**
The Green Mile	AP	0.00	14.05	14.12	14.09	6.92	10.12	9.83	13.00	14.42	**28.80**	27.60
	Recall	0.00	51.42	62.46	60.57	53.94	57.10	55.52	60.57	62.78	74.13	**78.23**
Harry Potter and the Sorcerers Stone	AP	0.00	10.30	8.09	8.18	5.66	7.84	7.95	8.04	8.20	**27.17**	23.65
	Recall	0.00	44.35	51.05	52.30	46.03	48.54	48.54	49.37	52.72	76.57	**78.66**
American Psycho	AP	0.00	14.78	16.76	17.22	12.29	14.88	14.95	15.68	16.54	**34.32**	32.87
	Recall	0.27	34.25	67.12	66.58	60.82	64.66	63.56	66.58	67.67	**81.92**	80.27
One Flew Over the Cuckoo Nest	AP	0.00	5.68	8.14	6.27	1.93	8.49	8.51	9.32	9.04	14.83	**21.13**
	Recall	1.01	25.25	41.41	34.34	32.32	36.36	37.37	36.36	40.40	49.49	**54.55**
Shawshank Redemption	AP	0.00	8.94	8.60	8.89	4.35	7.99	8.91	9.22	7.86	19.33	**19.96**
	Recall	1.79	46.43	78.57	76.79	73.21	73.21	78.57	75.00	78.57	94.64	**96.79**
The Firm	AP	0.05	4.46	7.91	8.66	2.02	6.22	7.15	7.25	7.26	18.34	**20.74**
	Recall	1.38	18.62	33.79	36.55	26.90	23.45	26.90	30.34	31.03	37.93	**44.83**
Brokeback Mountain	AP	2.36	24.91	16.55	17.82	14.60	15.16	15.58	15.41	16.21	**31.80**	30.58
	Recall	27.0	74.00	88.00	92.00	86.00	86.00	88.00	86.00	87.00	98.00	**100.00**
The Road	AP	0.00	13.77	6.58	7.83	3.04	5.11	5.47	6.09	7.00	**19.80**	19.58
	Recall	1.12	41.90	43.02	48.04	32.96	38.55	37.99	42.46	44.13	**65.36**	65.10
No Country for Old Men	AP	0.00	12.11	9.00	9.39	8.22	9.40	9.35	8.63	9.40	28.75	**30.45**
	Recall	1.12	33.46	48.90	49.63	46.69	47.79	51.10	49.26	48.53	71.69	**72.79**
Mean Recall		3.88	38.01	52.66	52.95	47.07	48.75	50.03	50.77	52.46	66.77	**69.10**
AP		0.40	10.97	9.62	9.88	5.94	8.57	8.83	9.31	9.64	22.51	**23.17**

Table 5: Performance of our model for movies in our dataset under different settings and metrics. UNI: uniform timeline, SVM: linear SVM trained to combine different similarities measures. The 1-layer CNN columns evaluate the leave-one-fature-out setting, where ∅ indicates all features are used, BLEU: no BLEU sentence similarity measure is used, TF-IDF: no tf-idf sentence similarity measure, SENT: our sentence embedding is not used, VIS: our video-text embedding not used, SCENE: no hybrid-CNN [42] is used in video representation, PRIOR: no uniform prior is used. CNN-3 is a 3-layer context-aware similarity measure, and CRF is our full model that uses CNN-3 features.

model that takes into account all the available similarities. We showed results on a new dataset of movie/book alignments as well as several quantitative results that showcase the power and potential of our approach.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015. 4

[2] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014. 2

[3] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *EMNLP*, 2014. 4

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 4

[5] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription. In *ECCV*, 2008. 2

[6] M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – Automatic Naming of Characters in TV Video. *BMVC*, pages 899–908, 2006. 2

[7] A. Farhadi, M. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences for images. In *ECCV*, 2010. 2

[8] S. Fidler, A. Sharma, and R. Urtasun. A sentence is worth a thousand pixels. In *CVPR*, 2013. 2

[9] A. Gupta and L. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, 2008. 1

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[11] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *JAIR*, 47:853–899, 2013. 2

[12] N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. In *EMNLP*, pages 1700–1709, 2013. 4

[13] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 1, 2

[14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[15] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539, 2014. 1, 2, 3, 5, 7, 8

[16] R. Kiros, Y. Zhu, R. Salakhutdinov, R. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. *arXiv preprint arXiv*, 2015. 3, 4

[17] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler. What are you talking about? text-to-image coreference. In *CVPR*, 2014. 1, 2

[18] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. Berg, and T. Berg. Baby talk: Understanding and generating simple image descriptions. In *CVPR*, 2011. 2

[19] D. Lin, S. Fidler, C. Kong, and R. Urtasun. Visual Semantic Search: Retrieving Videos via Complex Textual Queries. *CVPR*, pages 2657–2664, 2014. 1, 2

[20] 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*, pages 740–755. 2014. 1

[21] X. Lin and D. Parikh. Don't just listen, use your imagination: Leveraging visual common sense for non-visual tasks. In *CVPR*, 2015. 1

[22] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014. 1

[23] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy. Whats Cookin? Interpreting Cooking Videos using Text, Speech and Vision. In *NAACL*, 2015. 2

[24] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. In *arXiv:1410.1090*, 2014. 1, 2

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 3, 7

[26] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. BLEU: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002. 6

[27] H. Pirsiavash, C. Vondrick, and A. Torralba. Inferring the why in images. *arXiv.org*, jun 2014. 2

[28] V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei. Linking People in Videos with "Their" Names Using Coreference Resolution. In *ECCV*, pages 95–110. 2014. 2

[29] V. Ramanathan, P. Liang, and L. Fei-Fei. Video event understanding using natural language descriptions. In *ICCV*, 2013. 1

[30] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *CVPR*, 2015. 2, 5

[31] P. Sankar, C. V. Jawahar, and A. Zisserman. Subtitle-free Movie to Script Alignment. In *BMVC*, 2009. 2

[32] A. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient Structured Prediction with Latent Variables for General Graphical Models. In *ICML*, 2012. 7

[33] J. Sivic, M. Everingham, and A. Zisserman. "Who are you?" - Learning person specific classifiers from video. *CVPR*, pages 1145–1152, 2009. 2

[34] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *ACL*, 2:207–218, 2014. 2

[35] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. 4

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 5

[37] M. Tapaswi, M. Bauml, and R. Stiefelhagen. Book2Movie: Aligning Video scenes with Book chapters. In *CVPR*, 2015. 2

[38] M. Tapaswi, M. Buml, and R. Stiefelhagen. Aligning Plot Synopses to Videos for Story-based Retrieval. *IJMIR*, 4:3–16, 2015. 1, 2, 6

[39] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. J. Mooney, and K. Saenko. Translating Videos to Natural Language Using Deep Recurrent Neural Networks. *CoRR abs/1312.6229*, cs.CV, 2014. 1, 2

[40] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *arXiv:1411.4555*, 2014. 1, 2

[41] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *arXiv:1502.03044*, 2015. 2

[42] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning Deep Features for Scene Recognition using Places Database. In *NIPS*, 2014. 5, 8