Doctoral Thesis

Automatic Alignment Methods for Visual and Textual Data with Narrative Content

Author(s):
Dogan, Pelin

Publication Date:
2019

Permanent Link:

https://doi.org/10.3929/ethz-b-000359170 →

Diss. ETH No. 25896

Automatic Alignment Methods for Visual and Textual Data with Narrative Content

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by
Pelin Doğan
MSc in Electrical and Electronics Engineering, ÉPFL
Born on 10.04.1989
Citizen of Turkey

accepted on the recommendation of
Prof. Dr. Markus Gross, examiner
Prof. Dr. Leonid Sigal, co-examiner
Prof. Dr. Marc Pollefeys, co-examiner

2019

Abstract

Visual and linguistic tools are the most common medium for exchanging information, self-expression, and storytelling since the ancient times. The advancing technology and humanity enriched these media by introducing novel forms such as digital images, digital videos, online books, blogs, etc. which are tremendously increasing in quantity. Today, we are at a point where we see various manifestations of the same story for which joint analysis for the purposes of summarization, archiving, and automatic meta-data annotation becomes crucial. This leads to challenges in aligning multiple facet stories which would alleviate the difficulties in comprehensive understanding for a joint analysis. Traditional approaches usually require complicated pre-processing steps (*e.g.*, shot segmentation, speech/scene/face recognition and tracking), define a similarity metric between the sequence elements, and perform the alignment with standard techniques based on dynamic programming. Thus, they suffer from the limitations caused by the pre-processing steps, and the inherent drawbacks of Markov assumptions. In this thesis, we focus on aligning multi-modal data, specifically in visual and textual form, which is a fundamental step to learn and analyze correspondences between different manifestations of the same story. To achieve this, we build upon recent advances in deep and recurrent neural networks which provide efficient vectorial and contextual representations of the modalities to be aligned. Our label-based method for automatic alignment of video with narrative sentences proposes a highly efficient alignment technique that does not require heavy pre-processing steps, while enabling a fine level of granularity in the alignment result. Then, we develop an end-to-end differentiable neural architecture that addresses the limitations of the two-stage solutions by optimizing the similarity metric specifically for the alignment task while supporting one-to-one, one-to-many, skipping unmatched elements, and non-monotonic alignment. Expanding on this neural architecture, we develop a sequential spatial phrase grounding network that formulates grounding of multiple phrases as a sequential and contextual process allowing many-to-many matching. In a large variety of experiments, we show that using neural methods for multi-modal data alignment bear potential for more interesting research and applications by alleviating the large manpower that would be needed otherwise.

Zusammenfassung

Seit jeher sind visuelle und linguistische Werkzeuge die meist verbreiteten Medien zum Austausch von Information, zur Selbstdarstellung und zur Erzählung von Geschichten. Der technologische Fortschritt der Menschheit haben diese Medien mit neuen Formen angereichert, die in ihrer Verbreitung rasant wachsen. Zu neuen Formen von Medien zählen zum Beispiel digitale Bilder und Videos, online Bücher, Websites und Blogs, etc. Heute sind wir an einem Punkt angelangt, an dem oft zahlreiche Erscheinungsformen ein und der selben Geschichte veröffentlicht werden. Es ist daher von essentieller Bedeutung die verschiedenen Erscheinungsformen einer Geschichte geimeinsam zu analysieren, um diese anschließend zusammenzufassen, zu archivieren oder um automatisch wichtige Metadaten zu extrahieren. Die Angleichung und Synchronisierung der verschiedenen Erscheinungsformen ist ein wichtiger, methodischer Bestandteil, der die gemeinsame Analyse einer Geschichte signifikant vereinfacht. Traditionelle Methoden erfordern typischerweise komplizierte Vorverarbeitungsschritte, wie zum Beispiel Segmentierung, Sprach- und Szenenanalyse, Gesichtserkennung, etc. Des weiteren definieren traditionelle Methoden normalerweise eine Metrik, um die Ähnlichkeit zwischen einzelnen Elementen der verschiedenen Manifestierungen einer Geschichte zu messen, welche anschließend mit Standardmethoden basierend auf dynamischer Programmierung angeglichen werden. Aus diesem Grund erreichen traditionelle Methoden nur eingeschränkt gute Ergebnisse, da diese durch die Vorverarbeitungsschritte und die inhärenten Annahmen der Markov Theorie limitiert sind. Diese Arbeit befasst sich mit der Angleichung und Synchronisierung von multi-modalen Daten, insbesondere in visueller und textueller Form. Dies ist ein fundamentaler Schritt, um automatisch Korrespondenzen zwischen unterschiedlichen Erscheinungsformen einer Geschichte zu analysieren. Unsere Arbeit basiert auf neuen Erkenntnissen im Bereich von tiefen and rekurrenten neuronalen Netzwerken, die zu effizienten vektoriellen und kontextuellen Repräsentationen der Modalitäten fuhren. Unsere label-basierte Methode zur automatischen Angleichung von Videos mit Textsequenzen ist vergleichsweise effizient und benötigt keine tiefgreifenden Vorverarbeitungsschritte. Das Resultat ist eine detaillierte und granulare Angleichung. Ferner entwickeln wir eine durchgehend differentierbare neuronale Architektur, um die Probleme von zweistufigen Verfahren zu bewältigen. Die entwickelte neuronale Architektur optimiert die Ähnlichkeitsmetrik speziell für die Aufgabe der

Angleichung und Synchronisierung von multi-modalen Daten, die eins-zu-eins, eins-zu-viele, nicht existierende und nicht monotone Elemente enthalten können. Als Erweiterung dieser neuronalen Architektur entwickeln wir außerdem eine Methode zur Lokalisierung von Phrasen in Bildern, die als sequentieller und kontextualer Prozess formuliert ist und viele-zu-viele Beziehungen erlaubt. In einer Vielzahl an Experimenten zeigen wir, dass die Anwendung von neuronalen Netzen für multi-modale Datenangleichung grosses Potential birgt für weitergehende Forschung. Darüber hinaus sind die vorgestellten Methoden von hoher Relevanz für den Einsatz in Produkten, da mit diesen eine Vielzahl von arbeitsintensiven Schritten automatisiert werden können.

Acknowledgments

Without the support, guidance and encouragement of many people, this doctoral thesis would have never been accomplished. Over the last four years, they significantly helped me to succesfully complete this chapter of my life.

First of all, I would like to thank my advisor Prof. Markus Gross for giving me the opportunity to work in such an inspiring and challenging environment, and for guiding me through the development of my Ph.D. The opportunity and freedom to work on challenging research projects and to colloborate with significant researchers over visits and internships were extremely motivating. Many thanks to Prof. Leonid Sigal, who was more than a colloborator, for supporting my work. It has been a great experience to work with him and share his knowledge in many interesting discussions. My gratitude also goes to Dr. Boyang Albert Li for the significantly valuable internship and colloboration experience. Another special thanks to Prof. Jean-Charles Bazin and Dr. Marcel Lancelle for the fruitful collaborations.

I would also like to thank all my friends around the world, and my colleagues from the IVC, DRZ, and DRP for their support and friendship.

Last but not least, I would like to express my sincere gratitude to my family and my partner Johannes for their love and unconditional support throughout my life. Without all the amazing people in my life, this thesis would not have been possible. To them, I dedicate this thesis.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Vision and language are the most common medium of communication and expression throughout the history of humanity [1], [2]. Starting from the ancient times, humans drew and wrote on the walls of their caves by pragmatic exigencies such as exchanging information, expressing themselves, codifying laws, recording history, and storytelling. Considering the origins of writing which emerged smoothly from visual drawings such as logographs, symbolic systems, hieroglyphs, etc., it is safe to say that vision and language are natural allies and often complement each other, see for example Figure 1.1. Tradition did not change today, and we still use language and vision for exchanging information and storytelling. In fact, advancing technology and humanity extended and enriched these media by introducing novel forms such as digital images, digital videos, online books, blogs, etc. which are tremendously increasing in quantity. Nowadays, we are at a point where we see various manifestations of the same story. For example, a large number of books are converted to screen in the form of TV series or motion picture. Even more, some of them are re-adapted to screen multiple times. We see thousands of blog posts online where personal stories are unfolded with informative text and carefully chosen photos to highlight the most attractive part of people's experiences. The list could be continued more considering the steadily increasing amount of content. As fundemental techniques in analyzing images and text advance in natural language processing (NLP) and computer vision (CV), their joint analysis for the purposes of summarization, archiving, and automatic meta-data annotation becomes the next nat-

[1] [Chakravarthi, 1992]
[2] [Diringer, 2013]

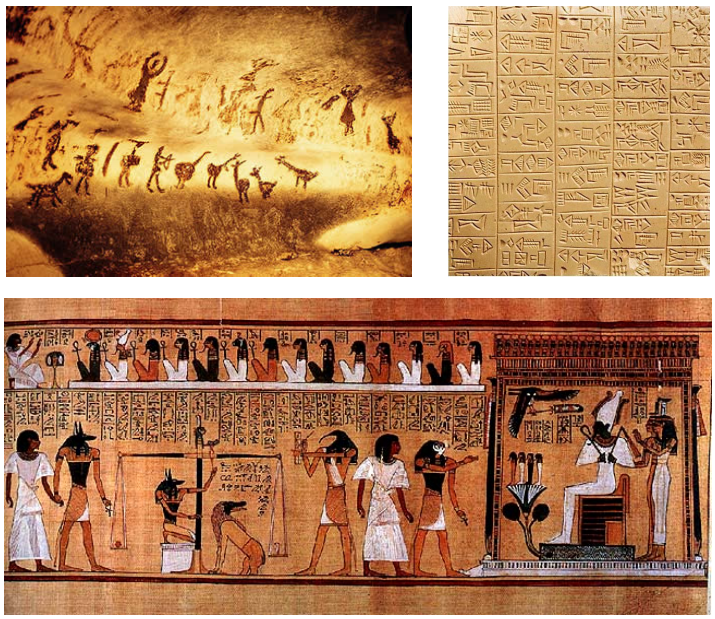


Figure 1.1: *Storytelling in early ages. Upper left: The paintings in the cave Magura representing dancing women, dancing and hunting men, disguised men, large variety of animals, suns, stars, instruments of labour, and plants, dating back to the bronze age. Upper right: Sumerian cuneiform scripts listing gifts to the high priestess of Adap on the occasion of her election, dating back to* 26^{th} *century. Lower row: Judgement scenes from the Book of Dead dating back to 1375 BC, Egypt.*

ural step. This leads to challenges in aligning multiple facet stories which would alleviate the difficulties in comprehensive understanding for a joint analysis.

In this thesis, we focus on aligning multi-modal data, specifically in visual and textual form, which is a fundamental step to learn and analyze correspondences between the different manifestations of the same story. A single picture, a video clip, a drawing, a Hollywood movie, a book, a phrase, a sentence, or a blog post, all these are an example of a story since they are a particular person's representation of the facts of a certain matter, see for example Figure 1.2. Aligned multi-modal data play an important role in various applications including data retrieval, data archiving, data summarization, and even more data generation by creating large corpuses for learning.

For example, for media-services providers automatically aligned visual and textual data would play an important role in fast retrieval of a desired content. In most TV and movie shooting workflows, the screenplays are prepared first, and then actors are chosen, which is followed by shooting. The

(a) (b) (c)

Figure 1.2: *Different manifestations of the story of Chris McCandless: "Into the Wild". (a) A book based on his journal by Jon Krakauer. (b) A documentary based on his journey. (c) The movie adaptation of the book in (a). (In addition, there are also newspaper articles about his story, his journal and various paintings inpired by the events.)*

screenplays may need multiple revisions to accommodate the talent that has been cast in the various roles. The principle filming captures most of the shooting script as much as possible. However, the post-production stage can detect problems and scenes that may need to be added, deleted, or revised. Clearly, there are many opportunities to deviate from the original script on the way to the final version of the video, which would result in unlabeled video/text data aggravating content retrieval. At this point, automatic video-text alignment methods would significantly reduce the manual work needed to align the script to the movie. Even more, these precisely aligned movie-script pairs can be used to create audio description[3] (AD) for a movie by retrieving the corresponding sentences from the script for each scene. Another benefit of such alignment would be automatic meta-data annotation in videos, considering scripts come with associated meta information such as scenes, characters and dialogues. Furthermore, by localizing the noun phrases of the aligned sentences in the individual frames of the corresponding video section, additional meta-data extraction can be performed, see Figure 1.3.

Many of these applications require finding complex correspondences within the visual and textual input sequences, which can then be used to align the sequence elements. These correspondences are often obtained by sophisticated extraction of comparable feature representations in each modality, often performed by a deep neural network. Common approaches to this

[3] Audio description is an additional narration track intended primarily for blind and visually impaired consumers of visual media. It consists of a narrator talking through the presentation, describing what is happening on the screen or stage during the natural pauses in the audio, and sometimes during dialogue if deemed necessary. [aud, 2009]

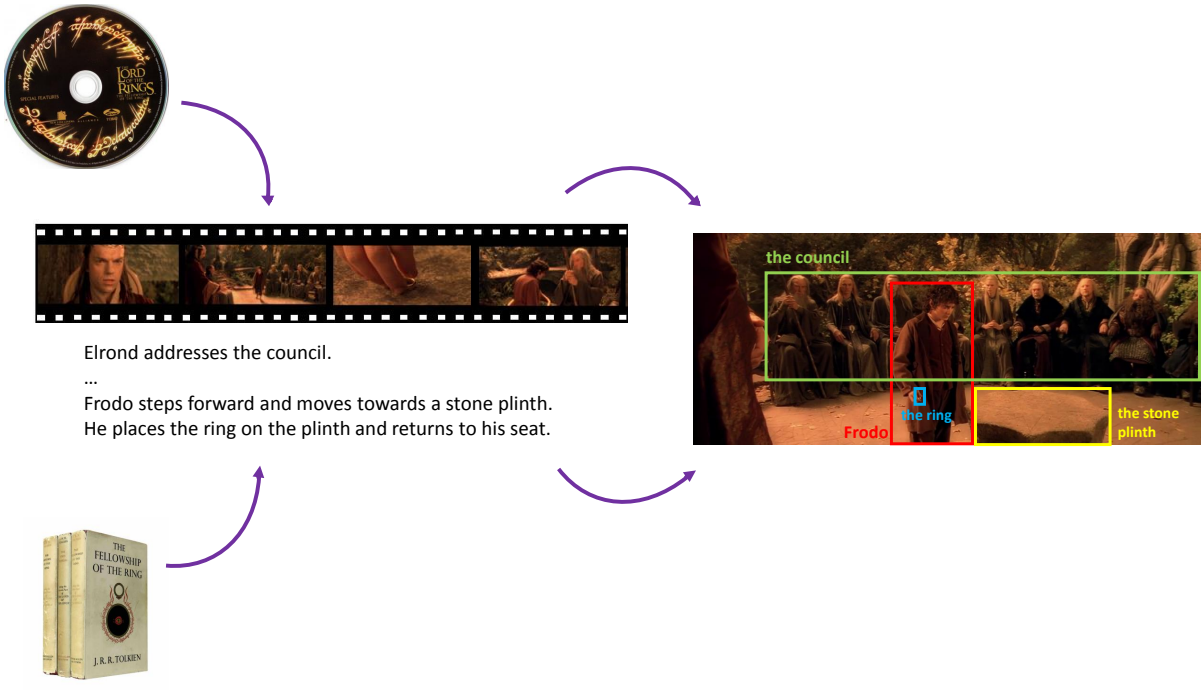


Figure 1.3: *Meta-data extraction by aligning multi-modal data. First, the sentences are aligned with the corresponding shots. Then, the phrases of the sentences are aligned with the entities and propagated along the frames*

alignment problem usually define or learn some sort of a similarity metric between elements of the sequences, and then find the optimal alignment between the sequences. However, the defined similarity metric is mostly not optimal for the alignment task. Furhermore, the performed alignment techniques take limited local context into account, but contextual information conducive to alignment may be scattered over the entire sequence due to stroy-telling nature of the data.

The goal of this thesis is to do basic research and to investigate innovative methods for automatic alignment of visual and textual data containing narrative content which is an important link in joint understanding of multimodal content, and is closely related to activity recognition, dense caption generation, and multimedia content retrieval. The focus lies on the comparable representation of the sequences, and their optimal alignment taking account of causal and temporal/spatial interactions between the sequence elements. To achieve this, we build upon recent advances in deep and recurrent neural networks which provide efficient vectorial representations of the modalities to be aligned.

Recently, a number of novel visual and textual sequence alignment tech-

niques have been proposed that are able to provide solutions to certain types of problems with different granularity levels and various degrees of global contextual information. For the video-text alignment problem, most solutions provide alignment at a coarse level, such as matching a book chapter to an episode of a TV show, or aligning a book paragraph to a long scene. The majority of these approaches use similarity between the written dialogues and the captions in the video as a cue. However, basing upon only dialogues results in imprecise alignment for the video shots that do not contain dialogues, but action. Furthermore, they mostly do shot threading as a preprocessing step which poses limitations on the granularity and the precision of the resulting alignment from the very beginning. Even if the used shot threading method results in a perfect shot segmentation, there could be long continuous shots, which would need to be parsed even more due to changing semantics and action.

Another direction to overcome the limitations of the prior methods is using a neural alignment process. With a carefully designed end-to-end neural architecture, the similarity metric between the sequence elements can be learned and optimized specifically for, and jointly with, the alignment task. In this way, a more optimal solution can be computed compared to the *optimal* solutions of the two-stage solutions. Furthermore, the alignment process can be formulated to consider the global context rather than local context.

Motivated by the successes of the prior approaches, the goal of this thesis is to analyze and improve alignment techniques for visual and textual data, as well as applying the advances in neural networks to design novel methods.

First, we investigate in Chapter 3 a label-based method for automatic alignment of video with narrative sentences, which provides automatic timestamps for each narrative sentence. Our approach segments the video into semantic shots and aligns them with the sentences in an iterative way by exploiting vector descriptors for text representation. We compute the similarity between both types of information using vectorial descriptors and propose to cast this alignment task as a matching problem that we solve via dynamic programming. The presented method is simple to implement, highly efficient, and does not require the presence of frequent dialogues, subtitles, and character face recognition. In contrast to previous two-stage solutions, our approach does not assume any pre-segmentation or shot threading of the video, instead works on the raw video. We introduce the novel term *semantic cut* to describe semantic change in a continuous camera shot, and we use frame-based high-level labels to group the frames in order to detect these semantic changes through the video. In this way, each shot contains relatively different semantics knowing that the information given by

the different sentences is relatively different. Then, we formulate the problem of text-video alignment as sentence-shot alignment by finding similarity between the high-level labels in the shots and the words of the sentences. Our final alignment is formulated in a graph-based approach computing the minimum distance path from the first to the last sentence-shot pair.

In Chapter 4, we explore a novel method by proposing an end-to-end differentiable neural architecture for heterogeneous sequence alignment which addresses the limitations of two-stage solutions by optimizing the similarity metric specifically for the alignment task. Standard techniques for the alignment task, including Dynamic Time Warping (DTW) and Conditional Random Fields (CRFs), suffer from inherent drawbacks. Mainly, the Markov assumption implies that, given the immediate past, future alignment decisions are independent of further history. The separation between similarity computation and alignment decision also prevents end-to-end training. To overcome these limitations, we formulate the alignment problem as a sequential alignment decision classification problem, where alignment actions are implemented as moving data between stacks that represent the currect workspace. This flexible architecture supports a large variety of alignment tasks, including one-to-one, one-to-many, skipping unmatched elements, and (with extensions) non-monotonic alignment. Extensive experiments on semi-synthetic and real datasets show that our algorithm outperforms state-of-the-art baselines.

Finally, in Chapter 5, we propose an end-to-end neural architecture for the phrase grounding problem where the task is to align sentence phrases to the corresponding image regions. Unlike prior methods that typically attempt to ground each phrase independently by building an image-text embedding, our architecture formulates grounding of multiple phrases as a sequential and contextual process. The benefit of this architecture is its ability to utilize rich context of prior matches along the way by introducing the notion of contextual phrase grounding. Furthermore, the resulting architecture, supports many-to-many matching by allowing an image region to be matched to multiple phrases and vice versa. We show competitive performance on the Flickr30K benchmark dataset and, through ablation studies, validate the efficacy of sequential grounding as well as individual design choices in our model architecture.

1.1 Principle Contributions

In the following we list the main contributions of the work presented in this thesis:

- We propose a novel method to align human-written sentences (such as scripts and audio description texts) with the complete set of shots that constitutes the video. Our approach does not require dialogues, subtitles, and face recognition. Unlike prior methods that perfrom pre-segmentation or shot threading, the proposed method directly works on the raw video. We automatically segment the input video into shots by using frame-based high-level semantics so that a semantic change in a continuous camera shot can be detected. We refer to this semantic change as semantic cut throughout this thesis. We also introduce a refinement process to optimize the semantic cuts so that they tend to correspond to one sentence each. Furthermore, we introduce a novel dataset of script sentence alignments of various video sequences which are publicly available on the project page.
- We propose a deep neural architecture for the temporal alignment of heterogeneous sequential data which overcomes the inherent drawbacks of standard techniques including dynamic time warping (DTW) and conditional random fields (CRFs) that cast Markov assumptions. In contrast to two-stage solution of the traditional methods, our framework combines (1) the similarity computation and (2) finding the optimal alignment with end-to-end training where the alignment actions are implemented as moving data between stacks of Long Short-term Memory (LSTM) blocks. This flexible architecture supports a large variety of alignment tasks, including one-to-one, one-to-many, skipping unmatched elements, and (with extensions) non-monotonic alignment. Extensive experiments on semi-synthetic and real datasets show that our algorithm outperforms state-of-the-art baselines.
- We propose the notion of contectual phrase grounding where earlier grounding decisions can inform the latter. We formalize this process in the end-to-end learnable neural architecture we call SeqGROUND. The benefit of this architecture is its ability to sequentially process many-to-many grounding decisions and utilize rich context of prior matches along the way. Furthermore, we show competitive performance both with respect to the prior state-of-the-art and ablation variants of our model. Through ablations we validate the

efficacy of sequential grounding as well as individual design choices in our model.

1.2 Publications

This thesis is based on the following peer-reviewed conference publications:

- P. Dogan, M. Gross, & J.C. Bazin. Label-based automatic alignment of video with narrative sentences. In Proceedings of European Conference on Computer Vision Workshops 2016.
- P. Dogan, B. Li, L. Sigal, & M. Gross. A Neural Multi-sequence Alignment TeCHnique (NeuMATCH). In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018.
- P. Dogan, L. Sigal, & M. Gross. Neural Sequential Phrase Grounding (SeqGROUND). In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019.

During the course of this thesis, the following peer-reviewed papers [Doğan et al., 2015] and [Lancelle et al., 2019] were published, which are not part of this thesis.

- P. Dogan, T. O. Aydin, N. Stefanoski, & A. Smolic. Key-frame based spatiotemporal scribble propagation. In Proceedings of the Eurographics Workshop on Intelligent Cinematography and Editing 2015.
- M. Lancelle, P. Dogan, & M. Gross. Controlling Motion Blur in Synthetic Long Time Exposures. In Proceedings of the Eurographics 2019.

CHAPTER 2

Background and Related Work

This chapter reviews influential research in the areas of joint reasoning of visual and textual data and multi-modal tasks. Because such applications often require representation of unimodal data, we also include a more general review on unimodal representation techniques with a brief overview on neural networks.

2.1 Overview on Neural Networks

Neural networks is one of the most popular machine learning topics at present due to its outstanding performance in accuracy and speed. In this thesis, the representation of visual and textual data, and problem formulations heavily rely on neural networks. Therefore, it is obligatory to review the basic building blocks of neural architectures briefly. A neural network is a collection of connected units or nodes, which loosely model the neurons in a biological brain and produce an output by applying a non-linear function, activation function, to its inputs. The connections between the nodes are called 'edges' that typically have a weight that adjusts as learning proceeds. Typically, neurons are aggregated into layers, and different layers may perform different kinds of transformations on their input. [1]

[1] We refer the reader to [Goodfellow et al., 2016] for a detailed overview over neural networks and related literature, and Christopher Olah's blog on *https://colah.github.io/* for comprehensive explanations.

Activation Functions

Activation functions introduce non-linear properties which are significant to represent non-linear complex arbitrary functional mappings between inputs and outputs. To enable the use of back-propagation optimization strategy, they need to be differentiable. In the following, we review some of the most popular activation functions used in the literature and throughout this thesis.

While being non-linear and bounding the output to the range $(0, 1)$, *sigmoid* activation suffers from vanishing gradients during backpropagation, and slow convergence. Being a rescaled version of the *sigmoid* tthat has an output range of $(-1, 1)$, the *tanh* function centers the data around 0, and results in higher gradients that help in a better learning rate. Being the most commonly used activation function, *ReLU* provides sparse activation in a randomly initialized network and cause fewer vanishing gradient problems compared to sigmoidal activation functions that saturate in both directions. These activation functions and their variants serve different purposes during training which make them more popular relative to each other for certain tasks: *sigmoid* for classification problems, *tanh* often for regression, *ReLU* mostly for intermediate layers.

2.1.1 Convolutional Neural Networks

Convolutional neural network (CNN) is a type of neural network that uses many identical copies of the same neuron with weight-tying. This allows the network to have lots of neurons and express computationally large models while keeping the number of actual parameters that need to be learned relatively small. CNNs typically consist of convolutional layers, pooling layers, fully connected layers, and normalization layers, meaning that convolution and pooling functions are used instead of normal activation functions above. CNNs use relatively little pre-processing compared to other image classification algorithms. Leveraging the structure of the input data for better performance (e.g., close-by words signals in a voice recording or neighboring pixels in an image are related), CNNs have prevalent applications in image recognition, classification, medical image analysis, and speech recognition.

2.1.2 Recurrent Neural Networks

A recurrent neural network (RNN) is another class of neural networks where connections between nodes form a directed graph along a sequence, which

exhibits dynamic behavior for a time sequence using its internal state to process sequences of inputs. They are called *recurrent* because they perform the same task for every element of a sequence, with the output being dependent in the previous computations. Another way to think about RNNs is that they have a "memory" which captures information about what has been calculated so far. In theory, RNNs can make use of information in arbitrarily long sequences, but in practice, they are limited to looking back only a few steps. [2]

Long short-term memory networks (LSTMs), a special kind of RNNs, are explicitly designed to solve the long-term dependency problem, remembering information for long periods of time. LSTMs also have the repeating chain-like structure, however, the repeating module has a relatively more complicated architecture compared to standard RNNs.

Another popular variant of RNNs is a network of *gated recurrent units* (GRUs) introduced by [Cho et al., 2014] that performs similar to LSTMs. However, GRUs have been shown in [Chung et al., 2014] to exhibit better performance on smaller datasets. In general they have fewer parameters than LSTMs since they lack an output gate.

With lots of notable variants, RNNs in general are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. Due to this behaviour, we utilized RNNs, LSTMs in particular, to time sequences such as videos and natural language queries in this thesis.

2.2 Unimodal Representations

2.2.1 Representation of Visual Information

Visual data for computer vision applications can roughly be classified into two main classes: images, representing 2D visual information; and videos, representing 3D visual information as a sequence of images.

Image Representations

In recent years, it was shown that deep CNNs (or ConvNets) can learn image features that are transferable to many different vision tasks. These are

[2] Britz, Denny. Recurrent Neural Networks Tutorial. Available at: http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

a special kind of multi-layer networks which are designed to recognize visual patterns from raw pixels with minimal preprocessing, and to serve as rich feature extractors. The pioneering LeNet-5 [LeCun et al., 1998] model largely introduced the CNN as we know today. Similar to LeNet-5 in general architecture, but considerably larger, AlexNet [Krizhevsky et al., 2012] was introduced in 2012 which leaded significant part of computer vision community to take a serious look at deep learning. Last of the classical network architectures, but not least, VGG [Simonyan and Zisserman, 2014] was introduced in 2014 offering a deeper but simpler variant of the convolutional networks mentioned earlier.

The modern network architectures placed aside the simplicity of the networks above. GoogLeNet [Szegedy et al., 2015] introduced the *Inception* module by straying from the general approach of simply stacking convolutional and pooling layers on top of each other in a sequential structure, which allows some pieces of network to process in parallel. It is an accepted principle that deeper networks are capable of learning more complex functions and representations. However, it was observed that although better initialization and batch normalization techniques allow for deeper networks to converge, they often converge at a higher error rate than their shallower counterparts. To overcome this degradation problem, [He et al., 2016] released ResNet introducing residual blocks that learn residual functions as refinement step. Following these main architectures, more efficient alternatives and simple variants of the these blocks and architectures [Huang et al., 2016], [Xie et al., 2017], [Szegedy et al., 2016], [Szegedy et al., 2017], even automatically learned architectures [Zoph et al., 2018], were developped.

These learned features are transferrable to many vision tasks, [Yosinski et al., 2014], [Oquab et al., 2014] such as image classification, semantic segmentation, object detection/recognition/localization, instance segmentation, see Figure 2.1 [Dai et al., 2016]. [Donahue et al., 2014] and [He et al., 2014] use neural networks for generic visual recongition. Recent works such as [Long et al., 2015], [Noh et al., 2015], [Chen et al., 2018a] use deep neural networks for semantic segmentation where the task is labeling each pixel in the image with a category label. Forming two stage attentional cascades by integrating a region proposal network to base CNNs, region-based CNNs (R-CNNs) are introduced [Girshick et al., 2014], and widely used for object detection and localization as in Fast R-CNN [Girshick, 2015], Faster R-CNN [Ren et al., 2015], Mask R-CNN [He et al., 2017]. Morover [Redmon et al., 2016] and [Zhou et al., 2016] perform object localization without explicit region proposals.

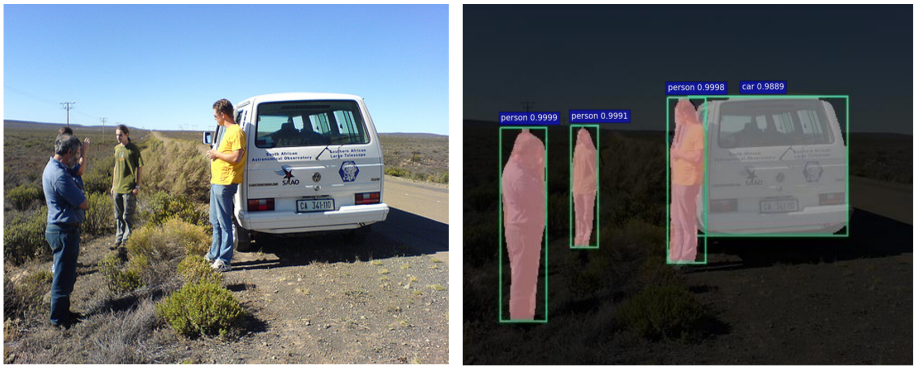


Figure 2.1: *Example of object detection, recognition, and instance segmentation performed by [Dai et al., 2016].*

Video Representations

Generic representations for video have received comparatively less attention. Following the advances in image representation, common encoding techniques for video include pooling and attention over frame features. [Venugopalan et al., 2014] represents videos by mean pooling over the frame features that are obtained by VGG model. [Xu et al., 2015a] and [Yao et al., 2015] are the early works using attention over frame features again using one of the image CNN arhictectures for feature extraction. To learn and represent the long term dependencies in a more comprehensive way compared to simple pooling methods, [Donahue et al., 2015], [Ranzato et al., 2014], and [Venugopalan et al., 2015] propose applying neural recurrence between video frames by feeding the CNN feautures of the video frames to a stack of long term RNNs. They show the effectiveness of such models for various video description tasks. Instead of using CNNs to extract frame features and building on them, [Tran et al., 2015] introduces a simple, yet effective approach for spatiotemporal feature learning using deep 3-dimensional CNNs trained on a large scale supervised video dataset. In addition to these pioneering wokrs, there is a lot of variants and extensions of the models above.

2.2.2 Representation of Textual Information

The initial successes of using deep neural networks to solve computer vision problems have led to efforts to use deep neural networks for learning features in other domains. Natural language processing (NLP) is one of these major domains, where applied deep neural networks can alleviate the challenges due to inherent complexity in representing, learning and using lin-

guistic knowledge which is often influenced by contextual and situational real-world settings.

Word Representation

In sparse representation, words are represented by a one-hot representation, which means each word has a unique symbolic ID. The dimension of this symbollic representation for each word is equal to the size of the vocabulary (number of words represented), where all but one dimension are equal to 0, and one is set to 1. This representation brings important shortcomings: there is no notion of *similarity* between words, and memory problems due to dimensionalty of the vocabulary. To overcome these shortcomings, words are typically mapped to continuous vector space with a much lower dimension, which is called word embedding. Methods to generate this mapping include dimensionality reduction on the word co-occurance matrix, probabilistic models, knowledge-based models, and neural networks.

Although the use of neural networks for word embeddings was initially proposed by [Bengio et al., 2003], it became prominent in NLP by *word2vec* [Mikolov et al., 2013a] and *GloVe* [Pennington et al., 2014], with the recent and rapid expansion and affordability in computational power, considering the computational complexity of these models. The word2vec models are shallow two-layer neural networks that are trained reconstruct linguistic contexts of the words. These models utilize a large corpus of text as input and produces a vector space with each unique word in the corpus assigned a corresponding vector in the space. The resulting word vectors, or embeddings, are positioned such that words that share common contexts in the corpus are located in close proximity to one another in the produced vector space. These neural networks can utilize either of two model architectures to produce a distributed representation of words: continuous bag-of-words (CBOW) or continuous skip-gram (SG). In the continuous bag-of-words architecture, the model predicts the current word from a window of surrounding context words. The order of context words does not influence prediction (bag-of-words assumption). In the continuous skip-gram architecture, the model uses the current word to predict the surrounding window of context words. The skip-gram architecture, [Mikolov et al., 2013b], weighs nearby context words more heavily than more distant context words. Similar to word2vec model, GloVe [Pennington et al., 2014] presents an unsupervised learning algorithm for obtaining distributive vector representations for words Training is performed on aggregated global word-word co-occurrence statistics from a large text corpus. GloVe seeks to produce word embeddings explicitly as a goal, in contrast to word2vec which produces

these as by-product. Following these unsupervised distributive word representation models, *fastText* [Bojanowski et al., 2017] and *ELMo* [Peters et al., 2018] augments the state-of-the-art results even more. Regular neural networks, in comparison to word2vec and GloVe models, generally produce task-specific embeddings with limitations in relation to their use elsewhere. Therefore, we proceed with the universal word embeddings as mentioned above.

Sentence Representation

Distributed word representations are often used in recurrent architectures in order to model sentential semantics. Building upon the word embeddings, there are many competing schemes for learning sentence embeddings: from simple word-vector averaging baselines to novel supervised/unsupervised methods. [Arora et al., 2016] and [Rücklé et al., 2018] provide simple but strong baseline approaches for averaging a sentence's word vectors. Beyond simple-averaging, skip-thoughts model [Kiros et al., 2015] proposes a simple neural network model for learning a fixed-length representations of sentences without labeled data using an objective function that adapts the skip-gram model for words. The only supervision it uses is the ordering of the sentences in the training text corpus. The model consists of an RNN-based encoder-decoder that is trained to reconstruct the surrounding sentences from the current sentence. Developing on skip-thoughts, quick thoughts model [Logeswaran and Lee, 2018] proposes a faster approach in unsupervised way. Overturning the assumption that unsupervised approaches result in lower quality, InferSent [Conneau et al., 2017] presents a simple architecture that learns supervised universal sentence representations using a large annotated corpus [Bowman et al., 2015]. The success of initial supervised approaches posed a the question of which supervised training task would learn sentence embeddings that better generalize on downstream tasks. [Subramanian et al., 2018] and [Cer et al., 2018] are the recent works that try to answer this question by multi-task learning. Sequence-to-sequence (seq2seq) prediction problems for machine translation is another challenging direction where the number of items in the input and output sequences can vary. To adress the seq2seq problems, initial works [Cho et al., 2014], [Sutskever et al., 2014], [Bahdanau et al., 2014], [Luong et al., 2015] drew the attention on RNN-based encoder-decoder architectures mainly using LSTMs and variants. .

2.3 Multimodal Tasks

2.3.1 Joint Reasoning of Text and Image/Video

With the increasing amount of available datasets, deep neural networks are widely used for task-specific applications. Majority of these applications represents images as a single feature vector from the top or mid-layer of a pre-trained convolutional network. Popular research topics in joint reasoning and understanding of visual and textual information include image captioning [Karpathy and Fei-Fei, 2015], [Mao et al., 2014], [Vinyals et al., 2015b], [Xu et al., 2015a], retrieval of visual content [Lin et al., 2014], text grounding in images [Fukui et al., 2016], [Plummer et al., 2017], [Rohrbach et al., 2016], [Wang et al., 2018] and visual question answering [Antol et al., 2015], [Sadeghi et al., 2015], [Xu and Saenko, 2016] and visual question answering [Antol et al., 2015], [Sadeghi et al., 2015], [Xu and Saenko, 2016]. Most approaches along these lines can be classified as belonging to either (i) joint language-visual embeddings or (ii) encoder-decoder architectures. The joint *vision-language embeddings* facilitate image/video or caption/sentence retrieval by learning to embed images/videos and sentences into the same space [Pan et al., 2016], [Torabi et al., 2016], [Xu et al., 2017], [Xu et al., 2015b]. For example, [Hodosh et al., 2013] uses simple kernel CCA and in [Farhadi et al., 2010] both images and sentences are mapped into a common semantic *meaning* space defined by object-action-scene triplets. More recent methods directly minimize a pairwise ranking function between positive image-caption pairs and contrastive (non-descriptive) negative pairs; various ranking objective functions have been proposed including max-margin [Kiros et al., 2014] and order-preserving losses [Vendrov et al., 2015]. The *encoder-decoder* architectures [Torabi et al., 2016] are similar, but instead attempt to encode images into the embedding space from which a sentence can be decoded. Applications of these approaches for video captioning and dense video captioning (multiple sentences) were explored in [Pan et al., 2016] and [Yu et al., 2016a] respectively, for video retrieval in [Donahue et al., 2015], and for visual question answering in [Anderson et al., 2017].

2.3.2 Video-text Alignment

A common solution to the video-text alignment problem consists of two stages that are performed separately: (1) the learning of a similarity metric between elements in the sequences and (2) finding the optimal alignment between the sequences. Alignment techniques based on dynamic programming, such as Dynamic Time Warping (DTW) [Berndt and Clifford,

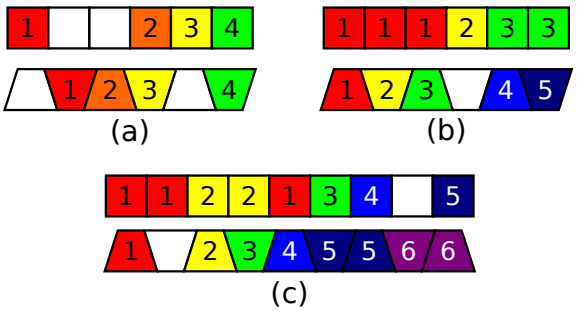


Figure 2.2: *Types of sequence correspondence. Matching blocks in two sequences have identical colors and numbers. (a) A one-to-one matching where the white blocks do not match anything. (b) A one-to-many matching where one block on the bottom sequence matches multiple blocks on the top. (c) A non-monotonic situation where the matching does not always proceed strictly from left to right due to the red-1 block after the yellow-2 on top.*

1994] and Canonical Time Warping (CTW) [Zhou and De la Torre, 2016], are widely popular. In all cases, these approaches are disadvantaged by the separation of the two stages. Conceptually, learning a metric that directly helps to optimize alignment should be beneficial. Further, methods with first-order Markov assumptions take only limited local context into account, but contextual information conducive to alignment may be scattered over the entire sequence. For example, knowledge of the narrative structure of a movie may help to align shots to their sentence descriptions.

Under the dynamic time warping framework, early works on video/image-text alignment adopted a feature-rich approach, utilizing features from dialogs and subtitles [Cour et al., 2008], [Everingham et al., 2006], [Tapaswi et al., 2014]. Adding on these features, [Sankar et al., 2009] uses location, face, and speec recognition for script to TV show alignment. However, the success of the method is mostly limited to TV series, since it needs pre-training of the frequent locations within the video to divide scenes. With the advances in the object detection/recognition and text representation with neural networks, recent works [Kong et al., 2014], [Lin et al., 2014], [Malmaud et al., 2015] used nouns and pronouns between text and objects in the scenes. [Tapaswi et al., 2014] presents an approach to align plot synopses with the corresponding shots with the guidance of subtitles and facial features from characters. They extend the DTW algorithm to allow one-to-many matching. Following the early work, [Tapaswi et al., 2015] presents another extension to allow non-monotonic matching in the alignment of book chapters and video scenes. The above formulations make use of the Markov property, which enables efficient solutions with dynamic programming (DP). At the same time, the historic context being considered is limited. [Zhu et al.,

	Method	End-to-end Training	Historic Context	Supports Non-monotonicity	Visual-textual Granularity
Sankar[2009]	DTW	No	Markov	No	fine
Zhu[2015]	CRF Chain	No	Markov + CoS	Yes	medium
Tapaswi[2015]	DP	No	Markov	Yes	coarse
Tapaswi[2014]	DP	No	Markov	No	fine
Bojanowski[2015]	QIP	No	global	No	fine
NeuMATCH	Neural	Yes	high order	Yes*	fine

Table 2.1: *Comparison of existing video-text alignment approaches. Prior methods are based on DTW, CRF and Convex Quadratic Programming (CQP). Non-monotonicity for NeuMATCH requires extensions in Section 4.1.3.*

2015] develops neural approach for the computation of similarities between videos and book chapters, using Skip-Thought vectors [Kiros et al., 2015]. In order to capture historic context, they use a convolutional network over a similarity tensor. The alignment is formulated as a linear-chain Conditional Random Field (CRF), which again yields efficient solution from DP. Although this method considers historic context, the alignment and similarity are still computed separately. [Bojanowski et al., 2015] formulates alignment as quadratic integer programming (QIP) and solve the relaxed problem. Weak supervision can be introduced as optimization constraints. This method considers the global context, but relates the video and text features by a linear transformation and does not consider non-monotonic alignment. Table 2.1 compares key aspects of these methods as well as our novel method NeuMATCH introduced in Chapter 4.

In summary, existing approaches perform the alignment in two separate stages: (1) extracting visual and textual features in such a way as to have a well defined metric, and (2) performing the alignment using this similarity (and possibly additional side information).

Shot Segmentation

Aligning sentences with the corresponding video parts requires shot detection and shot segmentation. For this, many of the automated shot-change detection methods use color histograms [Nagasaka and Tanaka, 1992], [Hampapur et al., 1995], [Lee and Ip, 1995], [Drew et al., 1999] or visual descriptors [Qu et al., 2009], [Lankinen and Kämäräinen, 2013], [Apostolidis and Mezaris, 2014]. These are mostly successful for shots that are

bounded by camera cuts or abrupt visual transitions. In the context of video-text alignment, distinguishing a semantic change through a single camera shot is valuable because a semantic change in the video is usually associated to a new description sentence within the script. Therefore we explore using semantic features, namely high-level labels, to segment the full video into "semantic shots" and in turn match the sentences with them, in Chapter 3.

2.3.3 Phrase Grounding in Images

Phrase grounding is defined as spatial localization of the natural language phrase in an image, see Figure 2.3. While significant progress has been made in phrase grounding, stemming from release of several benchmark datasets [Kazemzadeh et al., 2014], [Krishna et al., 2017], [Mao et al., 2016], [Plummer et al., 2015], and various neural algorithmic designs, the problem is far from being solved. Most, if not all, existing phrase grounding models can be categorized into two classes: attention-based [Xiao et al., 2017] or region-embedding-based [Plummer et al., 2018], [Zhang et al., 2017]. In the former, neural attention mechanisms are used to localize the phrases by, typically, predicting a course-resolution mask (*e.g.*, over the last convolutional layer of VGG [Simonyan and Zisserman, 2014] or another CNN network [He et al., 2016]). In the latter, the traditional object detection paradigm is followed by first detecting proposal regions and then measuring a (typically learned) similarity of each of these regions to the given language phrase. Importantly, both of these classes of models consider grounding of individual phrases individually (or independently), lacking the ability to take into account visual and, often, lingual context and/or reasoning that may exist among multiple constituent phrases.

We give a brief summary of the most notable approaches that have been proposed for phrase grounding over the years. Among the ranking-based methods [Karpathy et al., 2014] proposes to align sentence fragments and image regions in a subspace using a bi-directional loss, in addition to ensuring that correct phrases for each training image get ranked above incorrect ones, also ensures that for each phrase, the image described by that phrase gets ranked above images described by other phrases. Modeling the problem as a classification task, [Rohrbach et al., 2016] proposes a method to learn grounding in images by reconstructing a given phrase using an attention mechanism. They use a softmax function to estimate the posterior probability of a phrase over all available region proposals in an image. In a subsequent work, [Fukui et al., 2016] uses multimodal compact bilinear (MCB) pooling to represent multimodal features jointly which is then used to pre-

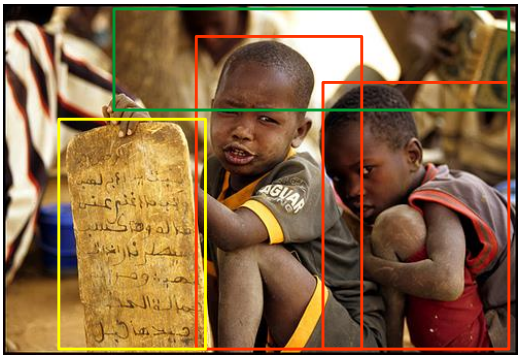


Figure 2.3: *Phrase grounding problem: given a sentence and an image, aligning phrases to the corresponding image regions.*

dict the best candidate bounding box in a similar way with the cost on high memory requisite. [Wang et al., 2016a] proposes an embedding network that learns a joint image-text embedding space using a symmetric distance function which is then used to score the bounding boxes to predict the closest to the given phrase. This embedding network is then extended by introducing a similarity network which aggregates multimodal features into a single vector rather than an explicit embedding space [Wang et al., 2018]. Different from the precursors, [Hu et al., 2016] proposes a recurrent neural network model to learn a scoring function that takes the text query, the candidate regions, their spatial configurations, and global scene-level context as input to output scores for the candidate regions using local image descriptors.

[Plummer et al., 2017] perform global inference using a wide range of image-text constraints derived from attributes, verbs, prepositions, and pronouns. [Yeh et al., 2017] uses word priors with the combination of segmentation masks, geometric features, and detection scores to select the candidate bounding box. [Wang et al., 2016b] proposes a structured matching method which attempts to reflect the semantic relation of phrases onto the visual relations of their corresponding regions without considering the global sentence-level context. [Plummer et al., 2018] proposes to use multiple text-conditioned embeddings in a single end-to-end model with impressive results on Flickr30K Entities dataset [Plummer et al., 2015], which can be added onto the prior methods.

These existing works ground each phrase independently, ignoring the semantic and spatial relations among the phrases and corresponding regions respectively. A notable exception is the approach of [Chen et al., 2017], where a query-guided regression network, designed to regress the rank of candidates phrase-region pairings, is proposed along with a reinforcement learning context policy network for contextual refinement of this rank-

ing. For *referring expression comprehension*, which is closely related to *phrase grounding* problem, [Yu et al., 2016b], [Nagaraja et al., 2016], [Yu et al., 2018] introduce taking account of context. Regarding visual data, they consider local context provided by the surrounding objects only. In addition, [Nagaraja et al., 2016], [Yu et al., 2018] use textual context with an explicit structure, based on the assumption that "referring expressions mention an object in relation with some other object". On the other hand, our method represents visual and textual context in a less structured, but more global, manner which alleviates more explicit assumptions made by other methods. Importantly, unlike [Yu et al., 2016b], [Nagaraja et al., 2016], [Yu et al., 2018], it makes use of prior matches through a sequential decision process. In summary, existing approaches perform grounding with two constraints: a region should be matched to no more than one phrase, or a phrase should be matched to no more than one region. Furthermore, most of these approaches consider the local similarities rather taking account both global image-level and sentence-level context.

CHAPTER 3

Label-Based Automatic Alignment of Video and Text

Audio description consists of an audio narration track where the narrator describes what is happening in the video. It allows visually impaired people to follow movies or other types of videos. However the number of movies that provide it is considerably low, and its preparation is particularly time consuming. On the other hand, scripts of numerous movies are available online although they generally are plain text sentences. Our goal is to temporally align the script sentences to the corresponding shots in the video, i.e. obtain the timing information of each sentence. These sentences can then be converted to audio description by an automatic speech synthesizer or can be read by a human describer. This would provide a wider range of movies to visually impaired people.

Several additional applications could benefit from the alignment of video with text. For example, the resulting correspondences of video frames and sentences can be used to improve image/video understanding and automatic caption generation by forming a learning corpus. Video-text alignment also enables text-based video retrieval since searching for a part of the video could be achieved via a simple text search.

In this chapter, we address temporal alignment of video frames with their descriptive sentences to obtain precise timestamps of the sentences with minimal manual intervention. A representative result is shown in Fig. 3.1. The videos are typically movies or some parts of movies with duration of 10 to 20 minutes. We do not assume any pre-segmentation or shot threading of the video. We start by obtaining the high-level labels of the video frames

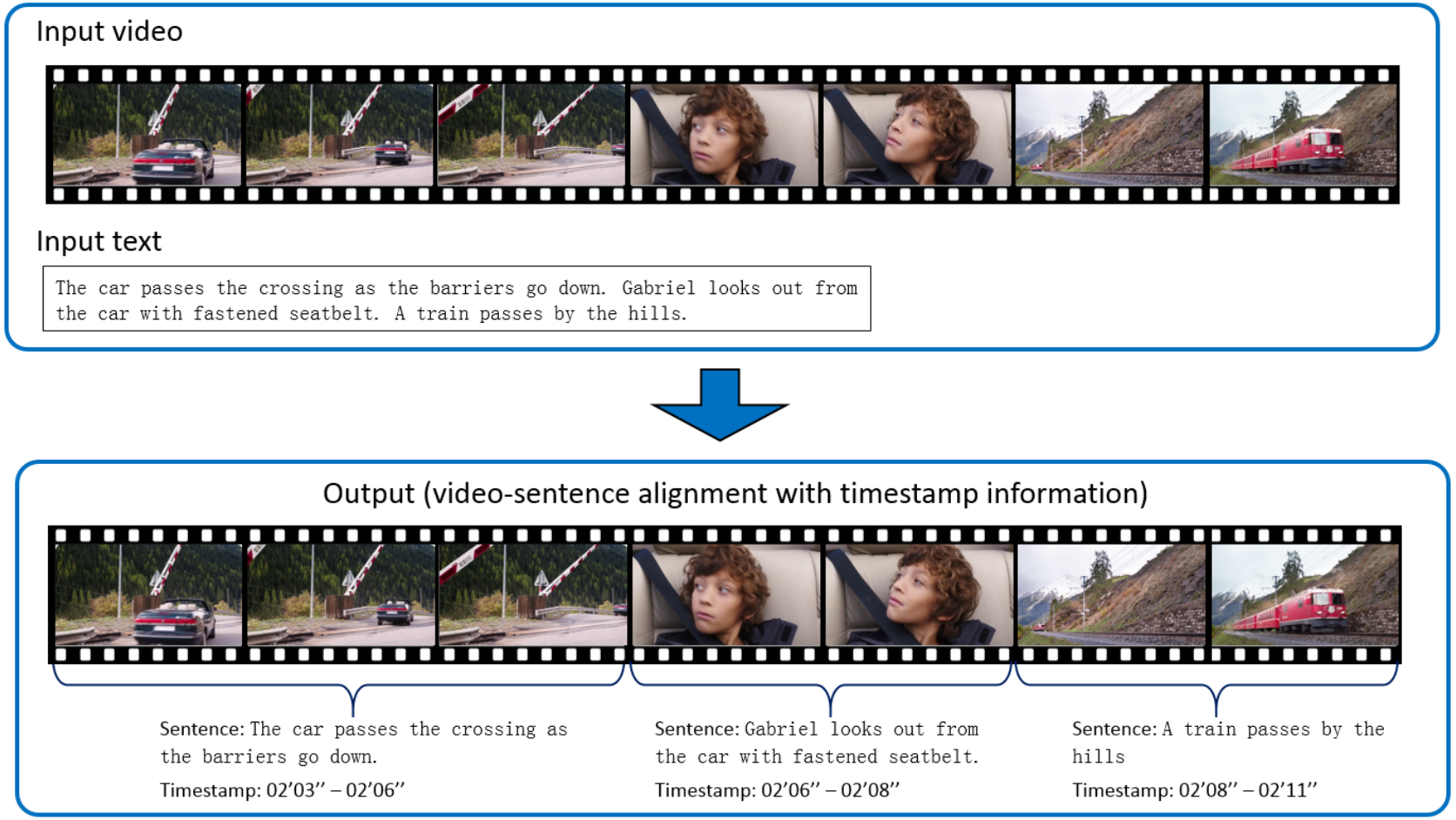


Figure 3.1: *Given an input movie and associated narrative sentences (e.g., from the movie script), our approach temporally aligns the video frames with the sentences and provides the timestamp information of the sentences. This figure illustrates a representative result for a 10-minute long continuous video from the movie Lucid Dreams of Gabriel. For a better visibility of the figure, only a 8-seconds segment is shown here.*

(e.g., "car", "walking", "street") with deep learning techniques [Jia et al., 2014] and use these labels to group the video frames into semantic shots. In this way, each shot contains relatively different semantics knowing that the information given by the different sentences is relatively different. Then we formulate the problem of text-video alignment as sentence-shot alignment by finding the similarity between the high-level labels in shots, and the words of sentences. This similarity is computed using the vectorial features of words and word-to-word distances. Our final alignment is formulated in a graph based approach computing the minimum distance path from the first to the last sentence-shot pair. The main contributions of the proposed approach are:

- We align human-written sentences (such as scripts and audio description texts) with the complete set of shots that constitutes the video. Our approach does not require dialogues, subtitles, and face recognition. Our approach directly works on the raw video, i.e. no presegmentation or cut is needed.
- We automatically segment the input video into shots by using frame based high-level semantics so that a semantic change in a continuous camera shot can be detected. We refer this semantic change as *semantic cut* through the chapter. We also introduce a refinement process to optimize the semantic cuts so that they tend to correspond to one sentence each.

3.1 Algorithm

3.1.1 Overview

In this section, we present our approach for aligning a video with its narrative sentences, which results in a time-stamp for each sentence [Dogan et al., 2016]. To have an accurate alignment, the text input should provide at least one sentence for each shot in the movie. By the term *shot* we refer to a series of frames that runs for an uninterrupted period of time with the same semantics, not necessarily defined by camera cuts. An example of text input for our algorithm can be a movie script (dialogues not required). Another example would be a transcribed audio description of the movie containing rich descriptions for visually impaired people. We assume that the sentences are in the same temporal order as the movie, like movie scripts and audio descriptions. Our approach is designed for videos having a dynamic plot with different scenes and actions as in the typical Hollywood movies. A

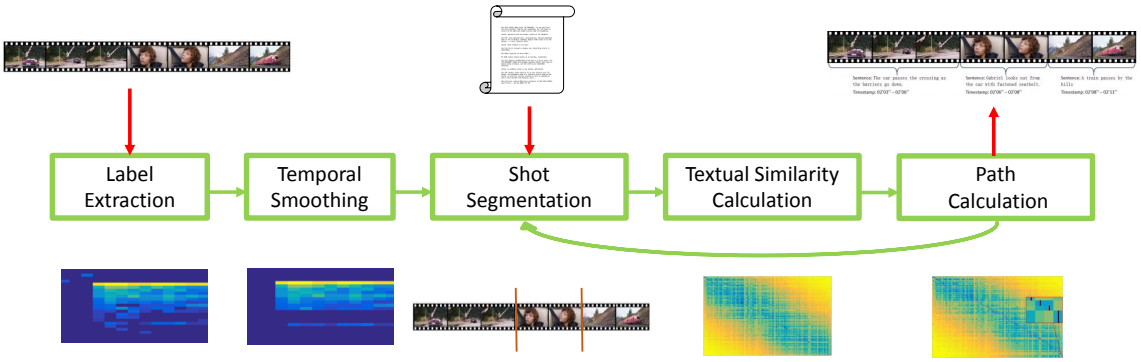


Figure 3.2: *Block diagram of our framework.*

counter-example is a biographical documentary film, such as an interview, where a person speaks to the camera during the whole duration of the video, i.e. without any changes of scene or action.

We start by obtaining the high-level labels for all the video frames in the form of words, as well as their confidence scores, using deep learning techniques [Jia et al., 2014]. Then we smooth these through the time domain to obtain temporal coherency. The temporally coherent results are used to detect the semantic changes in the video, which correspond to the beginnings and ends of the shots. Then, the labels and their confidence scores of the frames of each detected shot are grouped together to represent the shots. We then calculate a similarity score for each shot-sentence pair using the labels from shots and sentence words. This provides a cost matrix, and we then compute the minimum distance path assuming the matching of the first and last sentence-shot pairs are given. The nodes of the calculated path provide the matching of the sentence-shot pairs. This results in the annotation of each input sentence with the time-stamp of the matched shot. An overview on the main block of our framework is shown in Figure 3.2

3.1.2 High-Level Features and Temporal Coherency

We start by obtaining the high-level features (labels) of each frame of the input video. Each video frame is processed independently and thus can be processed in parallel. These high-level labels are in the form of text words and typically refer to an object (e.g. "car"), a scene (e.g. "street") or an action (e.g. "walking") visible in the video frame. We automatically obtain these labels, as well as their confidence score, by the deep learning based cloud service Clarifai[1] or Caffe framework [Jia et al., 2014] with pre-trained

[1] https://www.clarifai.com/

tunnel	cave	crystal	quartz	hole	mineral	crypt	dungeon	leisure activities	walking
0.98	0.98	0.61	0.61	0.61	0.61	0.16	0.15	0.13	0.13

Figure 3.3: *Representative example of high-level labels and their confidence scores for a given input video frame. Top: the input frame i from the movie Lucid Dreams of Gabriel. Bottom: the top 10 labels (in terms of confidence, out of* 1000*) and their confidence scores. The full confidence score vector (over all the labels) at frame i is written* w_i*.*

models for its CNN architecture. As a result, for each video frame i we obtain a feature vector w_i whose number of entries is the total number of labels (around 1000) and the entry values are the confidence scores for the label corresponding to that entry index. A representative result vector for a frame from the movie *Lucid Dreams of Gabriel* is shown in Fig. 3.3.

By concatenating these column vectors w_i over time, we obtain a matrix $\mathbf{W}$ containing the confidence scores of the labels through time. A representative example is shown in Fig. 3.4-top. Each row of this matrix represents the scores of the label corresponding to that row index (e.g., "car") through time. If the entries of this row are all zero or very small, it means the corresponding label is not seen in the frames, e.g., no "car" object is visible in the entire video. The values in the matrix rows are noisy due to motion blur, occlusions, lighting change, and all the effects that decrease the performance of the automatic object/scene recognition tools. Therefore the obtained matrix requires smoothing in the temporal axis (x-axis) to provide temporal coherency between the neighboring frames. We aim to find the labels that have high confidence scores while eliminating the labels that are not temporally consistent. We find the labels by a graph based shortest path approach.

We empirically set N to 10 and observed that higher values did not significantly change the final alignment results. We refer to the set of labels, one per frame through time, as a "path" q through the cost matrix, and our aim is to find the N shortest paths which will give us the N most dominant and temporally coherent labels for each frame. For this, we apply a shortest path algorithm N times in the following way. To find the first shortest path q_1, we consider the matrix $\mathbf{W}$ as a directed graph where the nodes are each $<$*frame*,*label*$>$ pair and the edges are defined using the entries of the matrix $\mathbf{W}$ (see Fig. 3.4). The weight of the edge from node (i,l) to node (i',l') is defined as

$$\phi\left((i,l),(i',l')\right) = \begin{cases} \lambda(1-w_{i'}(l')) + \varphi(l,l') & \text{if } i' = i+1 \\ \infty & \text{else} \end{cases} \tag{3.1}$$

where $\varphi(l,l')$ returns 1 when $l \neq l'$ and 0 otherwise, and where $w_i(l)$ is the score of the label indexed by l at frame i, i.e. node (i,l). The scaling factor λ sets the desired smoothness by penalizing the change of the label through the path and we set it to $\lambda = \frac{framerate}{10}$, where $framerate$ is the frame rate of the input video (usually 24fps). We apply Dijkstra's algorithm [Dijkstra, 1959] to obtain the minimum distance path solution. After finding the first path, we remove the edges pointing to the nodes of the calculated path so that those nodes cannot be selected for the future paths. We repeat this procedure to find the N shortest paths, that is to say the N most dominant labels. After the calculation of paths $q_1, ..., q_N$, the scores of the labels on the paths are smoothed with weighted moving average filter. A resulting temporally coherent matrix can be seen in Fig. 3.4-bottom. For writing simplicity, we still name this processed matrix as $\mathbf{W}$.

3.1.3 Shot Segmentation

So far, we explained how to obtain the temporally coherent labels and scores per frame stored in $\mathbf{W}$. We now aim to segment the whole input video into shots by processing the matrix $\mathbf{W}$. For a frame to be the beginning of a new shot, it has to be different than the past neighboring frame and similar to the future neighboring frame. Since we already have applied temporal filtering, the scores in $\mathbf{W}$ carry temporal information from neighborhood, not just from the surrounding frames. We calculate a score S_i that represents the score of frame i to be the beginning of a new shot:

$$S_i = |D_C(w_i, w_{i-1})|(1 - |D_C(w_i, w_{i+1})|) \tag{3.2}$$

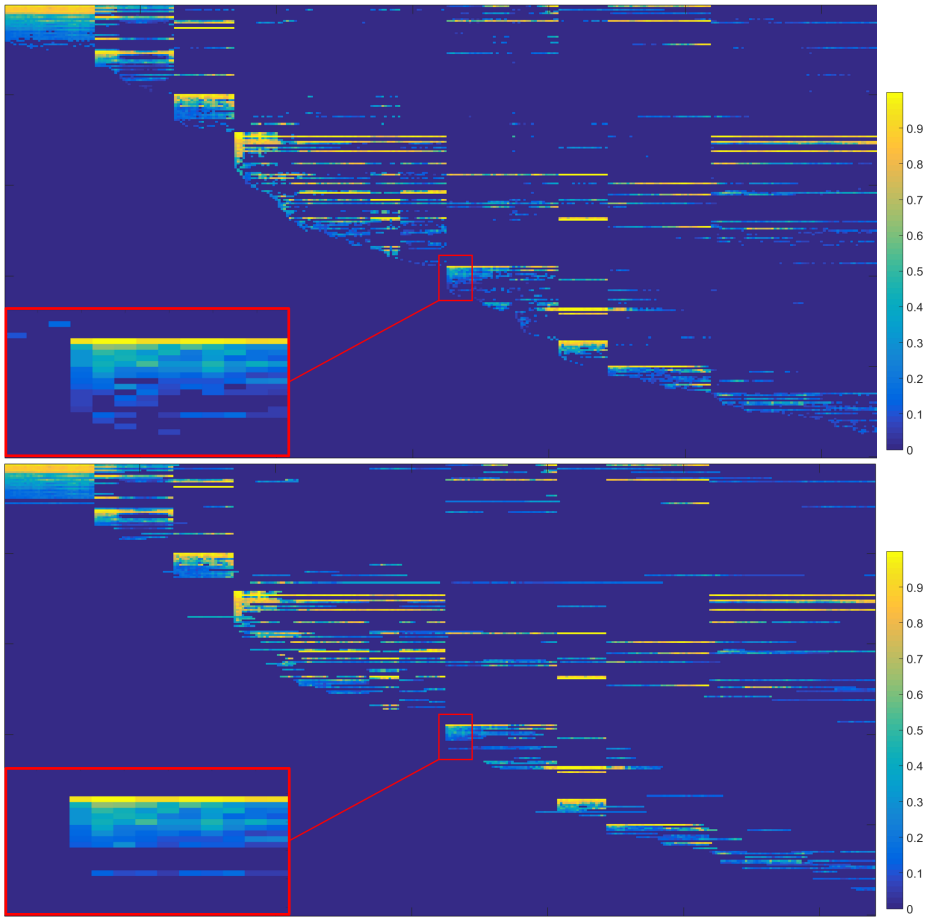


Figure 3.4: *Top: concatenated label vectors w_i. The height of this matrix depends on the number of unique labels detected through the whole video. Bottom: Temporally coherent result after path calculations where noisy labels are removed or smoothed.*

where w_i is the vector of label scores of the frame i and D_C computes the cosine distance of the input vectors.

Then we find the top K local maxima among all S_i, where K is the number of sentences in the input text. The frames corresponding to these maxima are our initial shot beginnings. It is important to note that we do not define *shots* by camera cuts. As discussed earlier, we refer to "shot" as a sequence of consecutive frames that have a similar semantic. Other than camera cuts, semantic cuts are considered as "shots" as well. For example, a continuous panning shot might have two different semantics with a soft border around the middle of the pan. This panning shot needs to be segmented into two shots since there might be two different sentences describing it due to semantic change. Therefore our aim is not finding the camera cuts, but opti-

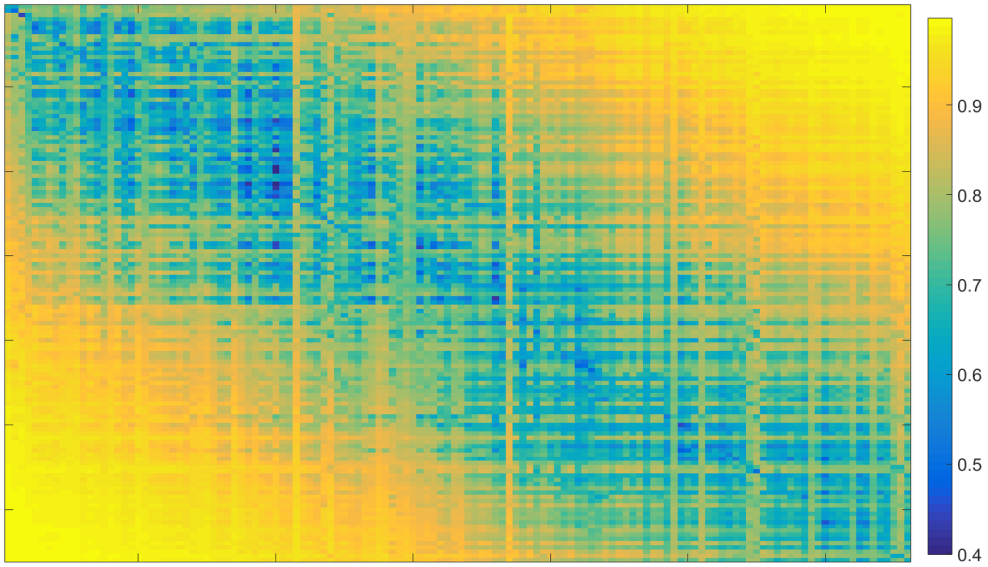


Figure 3.5: *Cost matrix whose elements c_{ij} are computed using similarity score between shot i (y-axis) and sentence j (x-axis).*

mizing (and thus detecting) the semantic cuts -including camera cuts- that will match the sentences in the best way.

3.1.4 Optimal Alignment

Cost matrix

In the previous sections, we have automatically segmented the input video into shots according to their semantic contents and their smoothed features. As the basis of our method, we need a robust estimate of the alignment quality for all the shot-sentence pairs. We observe that a shot and a sentence are more likely to be alignable together if the words in this sentence and the labels of this shot are semantically similar. Using this concept, we compute a similarity value v_{ij} between each shot i and sentence j. Subsequently, we transform these values into a cost matrix $\mathbf{C} \in \mathbb{R}^{K \times K}$, in which each entry c_{ij} specifies a cost for aligning a pair of shot and sentence.

We represent the shot labels and the sentence words using GloVe word vector descriptors [Pennington et al., 2014] of dimension $b = 300$. For each detected shot, we consider the set of all the N labels and scores found in all the frames of the shot. We denote the l-th label of the i-th shot by its confidence score $f_i(l)$ and its GloVe vector descriptor $d_i(l) \in \mathbb{R}^b$ where $l \in [1...N]$. Similarly, we denote the m-th word of the j-th sentence with its GloVe descriptor

$d_j(m) \in \mathbb{R}^b$. The similarity between the label l and the word with index (j, m) is calculated as

$$z_{ij}(l, m) = |d_i(l) - d_j(m)| \tag{3.3}$$

which is modified by Lorentzian stopping function as

$$y_{ij}(l, m) = \left(1 + \left|\frac{z_{ij}(l, m)}{\sigma}\right|^{\alpha}\right)^{-1} \tag{3.4}$$

where $\alpha = 3$ and $\sigma = 0.5$ for all the experiments shown in this chapter.

Finally the similarity values $y_{ij}(l, m)$ are used to compute the cost matrix $\mathbf{C}$ in which low values indicate shot-sentence pairs that are likely to be a good match. The entries of the cost matrix $\mathbf{C}'$ are computed as

$$c'_{ij} = 1 - \frac{1}{M}\sum_{m=1}^{M} f_i(l) \max_{l \in N} y_{ij}(l, m) \tag{3.5}$$

Lastly, we obtain the values of the cost matrix $\mathbf{C}$ by scaling the values of $\mathbf{C}'$ with an oriented 2D Gaussian factor which penalizes the elements in the upper right and lower left corner. In this way we incorporate the global likelihood of being at any node in the graph to our cost matrix considering passing through the nodes at the top-right or bottom-left corners are very unlikely.

$$c_{ij} = c'_{ij} \exp\left(-\frac{(i-j)^2}{2K^2}\right) \tag{3.6}$$

An example of cost matrix for each pair of sentences and computed shots is available in Fig. 3.5.

Path Calculation

So far we have described mappings between the shots and sentences. We now explain how to find a discrete mapping $p : \mathbb{R} \to \mathbb{R}^2$ in our cost matrix: for a time t, $p(t) = (i, j)$ means that the shot i corresponds to the sentence j. We refer to the discrete representation of a mapping p as a path through the cost matrix $\mathbf{C}$, and consider a graph based solution to find the minimum distance path. This path will provide the optimum shot-sentence pairings.

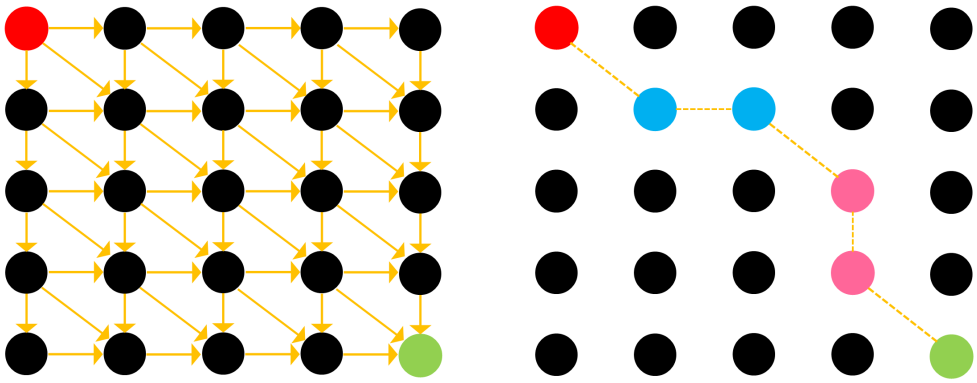


Figure 3.6: *Left: Possible oriented connections (in orange) between the nodes where the red node is considered the source and the green node is the sink. Right: An example path result from the source to the sink. (See text for details.)*

We compute the cost of a path p as the average of all the entries in the cost matrix that the path goes through:

$$\psi(p) = \frac{1}{T}\sum_{t=1}^{T}\mathbf{C}(p(t)) \tag{3.7}$$

where T denotes the number of steps in the path.

To find the path with minimum cost, we consider the cost matrix as a directed graph where a path is defined as the set of connected nodes. We identify a node by its position (i, j) and edge as an ordered pair of nodes. Since we assume the input text sentences are in the same temporal order as the video, we only allow forward motion. In other words each node (i, j) is connected to its three neighbors $(i, j+1)$, $(i+1, j+1)$, and $(i+1, j)$. The weight of each edge is the value of the cost matrix at the node that the edge points to. An example graph of the possible connections is shown in Fig. 3.6.

We use dynamic programming to find the minimum distance path [Sakoe and Chiba, 1978]. Computing the shortest path from the first node $(1, 1)$ to the last node (K, K) provides us the initial result for the shot-sentence pairings. An alignment result is shown in Fig. 3.7. The pink plot on the graph represents the ground truth alignment. The black plot shows the regions where our result is different than the ground truth. It is important to note that the y-axis represents the frames, not the shots. This is why paths have discrete vertical parts which corresponds to the set of frames corresponding to a shot.

Refinement

As mentioned earlier, the sentences in the input text description do not have to correspond to the camera cuts. In addition, the result of the shot segmentation does not have to give the perfect shots for the sentences. This may cause the matching of a shot with more than one sentence (horizontal parts in the path) or matching of a sentence with more than one shot (vertical parts in the path). Therefore, the alignment that is obtained by the current cost matrix may not be the optimum.

We compute the optimum alignment by modifying the cost matrix in an iterative refinement procedure. Starting with the current optimum path, we combine the shots that are matched to the same sentence into a single shot. Conversely we segment the shot that is assigned to more than one sentence for another round. The segmentation of this shot is conducted in a way similar to Sec. 3.1.3. We find $r - 1$ local maxima among S_i in Eq. 3.2 in the corresponding region of frames during this shot, where r is the number of resulting sentences matched with it. In this way we obtain r shots that can be assigned to these r different sentences.

For example, the shots corresponding to the pink nodes (same column) on the path in Fig. 3.6 will be combined together, while the shot corresponding to the blue nodes (same row) will be split into two shots. After this refinement, we repeat all the steps starting from Sec. 3.1.4 to find the new optimal path. In our experiments, we observed that the result converges in less than 4 iterations. The effect of this refinement step is shown in the cost matrices of Fig. 3.7.

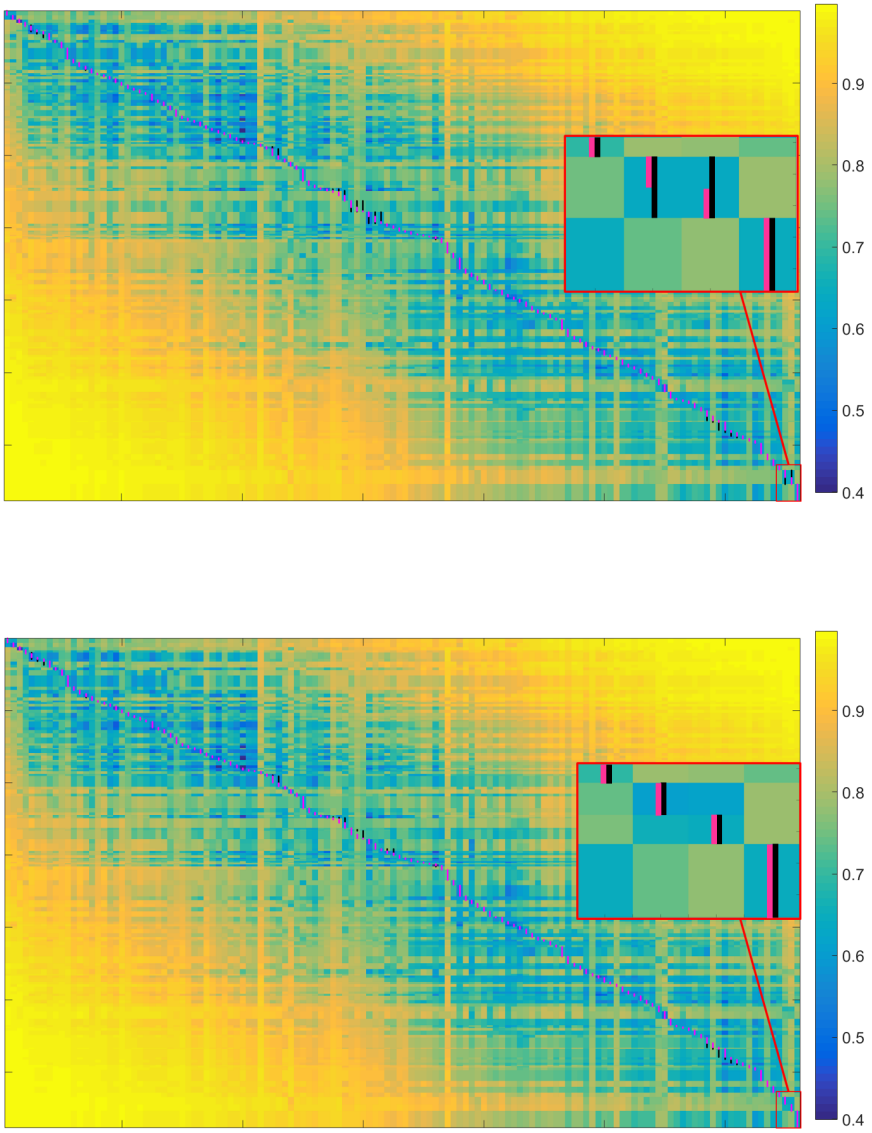


Figure 3.7: *The alignment result automatically obtained by our approach for the full movie (11 minutes) Lucid Dreams of Gabriel with its audio description sentences. Top: The initial alignment result using the initial shot segmentation results. The alignment of a shot to two consecutive sentences is seen in the close-up view (red box). Bottom: Our final alignment result after the refinement process. The close-up view shows that our result exactly matches with the ground truth alignment.*

He leaves by the back door *The mother is in the store and steals a chocolate from the cash desk.*

Figure 3.8: *Two consecutive shots (one frame of each shot is shown here) separated by a sharp camera cut and their aligned sentences, i.e. the nodes for these shot-sentence pairs are on the minimum distance path of the cost matrix.*

3.2 Applications

In this section we demonstrate different applications and the results obtained by our algorithm. Please refer to our project webpage for video results.

3.2.1 Video-sentence alignment.

Aligning descriptive sentences to video in an automatic way can provide rich datasets for modeling video contents. The resulting video-sentence alignments can be used as training data to learn models for the task of automatic generation of video descriptions. An example of video-sentence alignment obtained by our algorithm is available in Fig. 3.8. It shows two consecutive shots separated by a sharp camera cut and the automatic alignment of the corresponding sentences. The sentences are marked automatically by the timestamps that correspond to the very first frame of the shots by our algorithm since the beginning of these shots are captured perfectly.

3.2.2 Shot segmentation.

Shot segmentation is used to split up a video into basic temporal units called shots. These units have consecutive frames taken contiguously by a single camera, representing a continuous action in time and space. Shot segmentation is an important step for various tasks such as automated indexing, content-based video retrieval and video summarization. While detecting sharp camera cuts is not a difficult task (as shown in Fig. 3.8 for a sharp

Figure 3.9: *A continuous camera shot with two semantic shots aligned with the sentences from its audio description. Top row: She opens the car door and gets in. Bottom row: She gives the chocolate to Gabriel that is in the car. (from Lucid Dreams of Gabriel)*

camera cut), detecting only the camera cuts may not be sufficient for video-text alignment or other video retrieval tasks. The target material can have different types of separation. For example two sentences can take place in the same scene with a continuous camera shot while representing two different semantic information. A representative example of such a case is shown in Fig. 3.9 where the shot starts by a woman getting into the car and ends with a child having a chocolate bar. Although this scene is shot continuously by a panning camera (i.e. not camera cut), it represents two different semantics which are expressed by two sentences in the audio description. Our joint segmentation approach is able to successfully detect the semantic cuts indicated by different sentences in the text input.

3.2.3 Image-sentence alignment.

With the increasing trend of social media, a growing number of individuals share their own experiences as a blog post, which is in the form of text with highlighting images and videos, over a multitude of Web platforms. The visuals in such a blog are usually distributed all around the text without a clear connection or caption that is linking it to the text story. Therefore, aligning these visuals to their corresponding sentence in the blog in an auto-

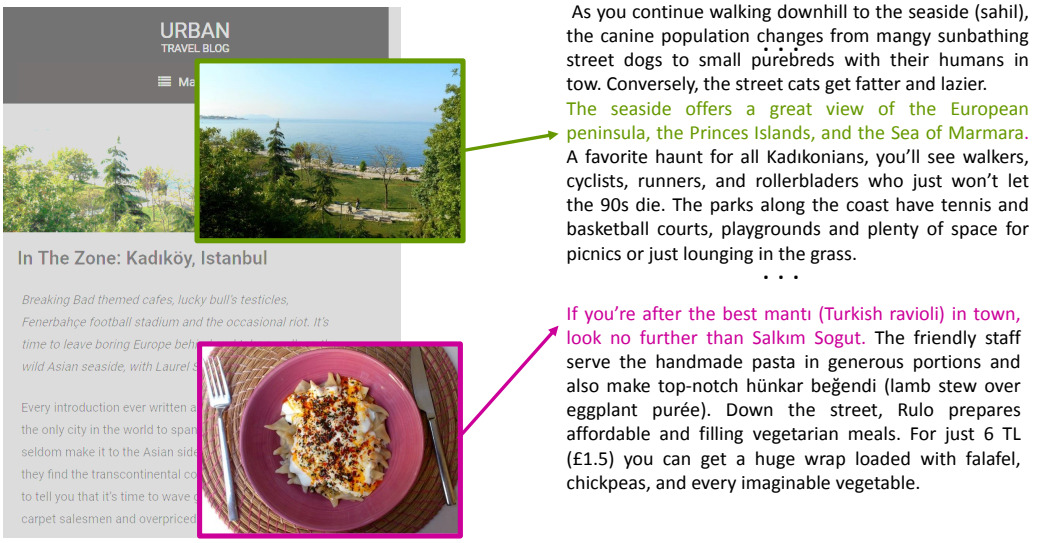


Figure 3.10: *Aligned image-sentence pairs in a blog post.*

matic way is a subsidiary step for the completeness of the blog. An example result computed by our method is shown in Figure 3.10.

3.3 Experimental Evaluation

We evaluated the proposed alignment method on a dataset of 12 videos with the sentences from their scripts or audio descriptions, including continuous long sections from the movies *Lucid Dreams of Gabriel*, *The Ninth Gate*, *The Wolf of Wall Street* and *Yes Man*. The duration of the videos in the dataset ranges from 10 to 20 minutes with 9.51 sentences per minute on average. More examples on video-text alignment by our method are shown at the end of this chapter in Figure 3.13-3.15.

We now present the evaluation of our proposed alignment approach with respect to the manually obtained ground truth data. We measure the alignment accuracy by computing the temporal error between the ground truth timestamps of the sentences and the timestamps obtained by our approach. Fig. 3.11 shows the distribution of the temporal error. It shows that 88.64% of the sentences have a temporal error of 0 second, i.e. our timestamps exactly correspond to the ground truth timestamps. This demonstrates the accuracy of our alignment approach.

We now present the evaluation of our proposed shot segmentation approach with respect to the manually obtained ground truth shot segmentation. We consider two metrics again. Firstly we measure the number of shots de-

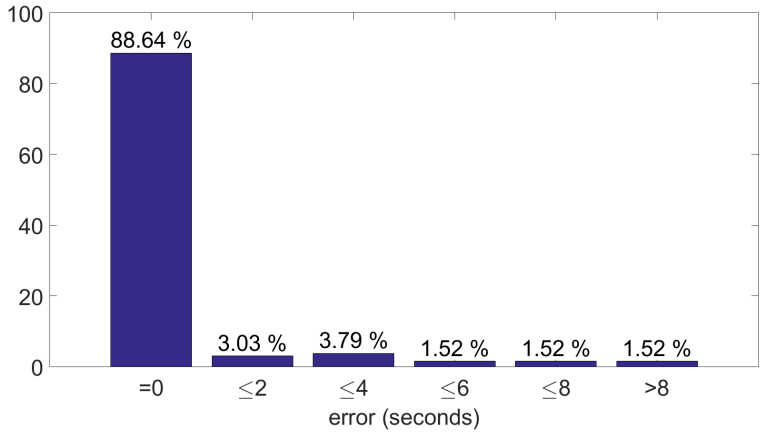


Figure 3.11: *Distribution of the absolute error in seconds on the timestamps obtained by our algorithm with respect to the ground truth timestamps.* 88.64% *of the sentences are matched perfectly to the first frame of the corresponding shot.*

tected by our approach over the total number of ground truth shots in the movie. Secondly, we measure the number of correctly detected shots by our approach over all detected shots, which includes false positives. The evaluation is shown in Fig. 3.12.

Our method has some limitations. First, in order to correctly align the frames with the corresponding sentences, the image labeling tools (e.g. object/scene recognition) should provide sufficiently accurate labels and scores. The accuracy of our method can directly benefit from the next advances of the image labeling tools.

Another limitation is that our method is not designed for videos that mostly consist of close-up shots (e.g. interview videos) rather than scenes, actions and motion. Such video frames would not result in sufficient object/scene labels due to the lack of action and scene changes. We focused on more general movies because we believe they are more common. However, our method is suitable for a simple integration of dialogue-caption alignment approaches used in [Tapaswi et al., 2015], [Zhu et al., 2015] that could be included as another variable in our global cost matrix. In future work, this integration could improve the results in videos that lack narrative sentences during dialogues.

A future application of our approach can be segmentation and structuring of videos that will allow important post-applications in content-based media analysis. Clustering of video units like shots and scenes allows unsupervised or semi-supervised content organization and has direct applications in browsing in massive data sources. Given the framewise high-level labels

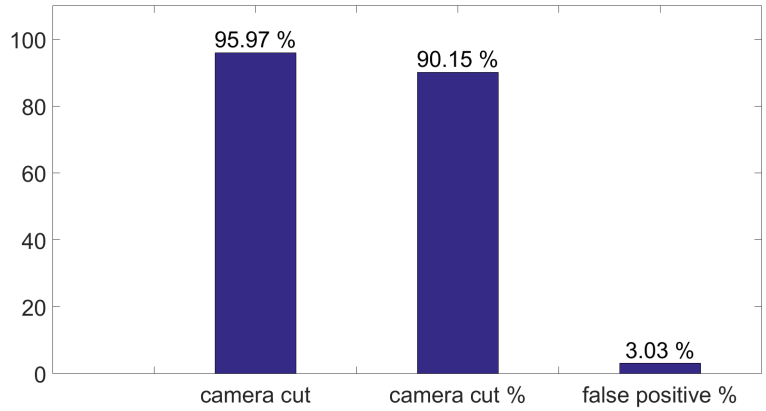


Figure 3.12: *Evaluation of our shot segmentation method.* 95.97% *of the ground truth camera shots are detected by our method.* 90.15% *of the shots detected by our algorithm correspond to the ground truth camera cuts. Meanwhile, only* 3.03% *of the shots detected by our approach are false positives.*

and timestamps of shot intervals of a video obtained by our algorithm, we can easily cluster these shots. Treating the rows of the cost matrix as the features of the segmented shots, one can simply apply a clustering method to obtain shot clusters.

In future work, it would be interesting to extend the proposed approach to cope with different types of media materials by bringing them into a common representation. For example a storyboard with drawing and sketches could be aligned with the corresponding shots in the movie using the high-level labels and their vector descriptors in an analogous way.

3.4 Summary

In this chapter, we considered videos (e.g., Hollywood movies) and their accompanying natural language descriptions in the form of narrative sentences (e.g., movie scripts without timestamps). We proposed a method for temporally aligning the video frames with the sentences using both visual and textual information, which provides automatic timestamps for each narrative sentence. We computed the similarity between both types of information using vectorial descriptors and propose to cast this alignment task as a matching problem that we solve via dynamic programming. Our approach is simple to implement, highly efficient and does not require the presence of frequent dialogues, subtitles, and character face recognition. Experiments on various movies demonstrated that our method can successfully align the

movie script sentences with the video frames of movies which allows detecting semantic changes in the video data. Currently, our method relies on the semtic changes of scenes and actions, which does not allow long pauses and dialogues. As a future direction, the method can be improved even more with and integrated dialogue-caption alignment method, which will avoid performance degredation caused by static scenes with dialogues.

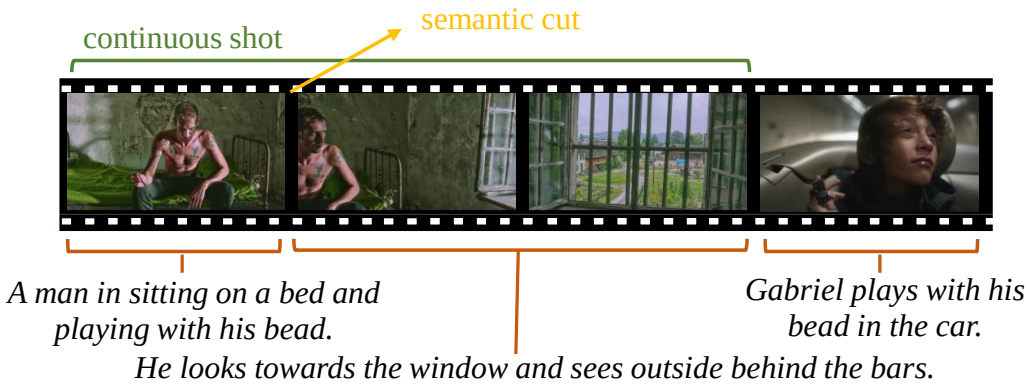


Figure 3.13: *Aligned video-sentence pairs from the movie The Lucid Dreams of Gabriel.*

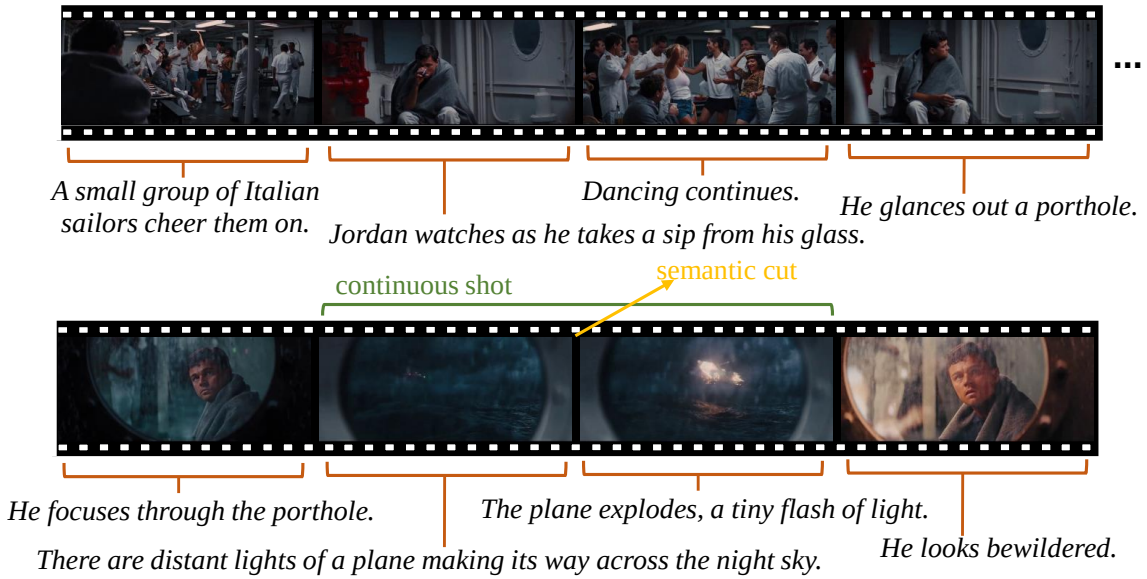


Figure 3.14: *Aligned video-sentence pairs from the movie The Wolf of Wall Street.*

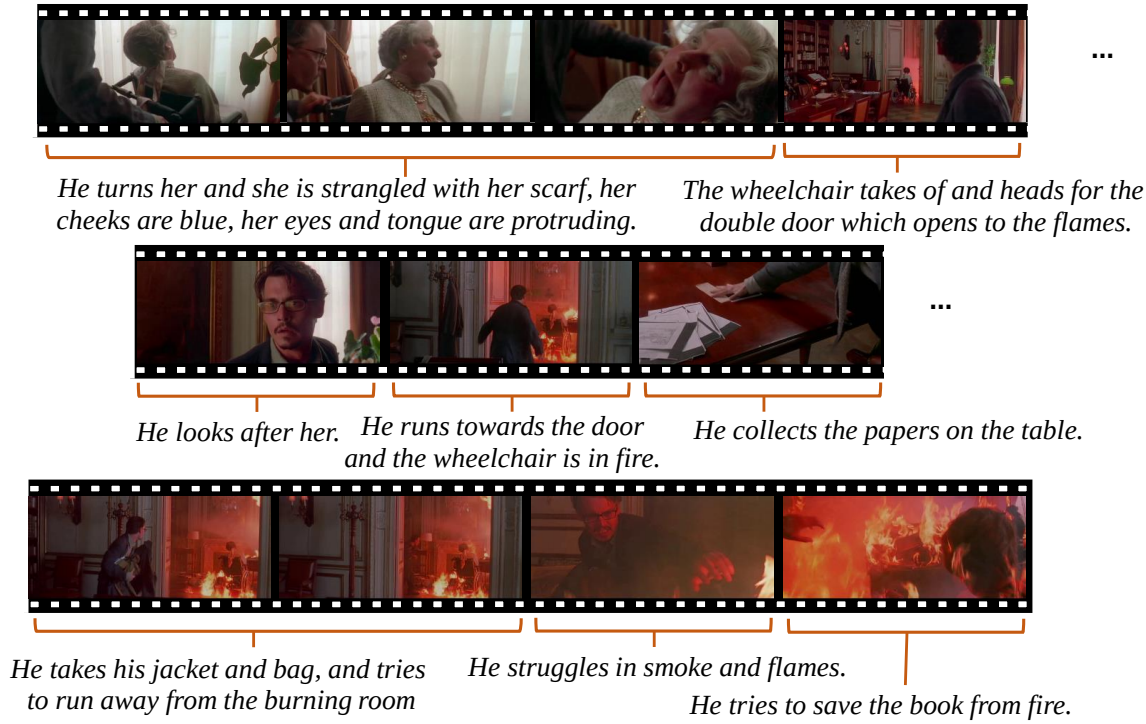


Figure 3.15: *Aligned video-sentence pairs from the movie The Ninth Gate.*

CHAPTER 4

Neural Matching of Video and Text

In this chapter, we focus on aligning heterogeneous sequences with complex correspondences without using dynamic programming (DP) based approaches. Heterogeneity refers to the lack of an obvious surface matching (a literal similarity metric between elements of the sequences). As shown as an example in the previous chapter, in the simple form, DTW/DP can be understood as finding the shortest path where the edge costs are computed with the similarity metric, so the decision is Markov. These approaches are disadvantaged by the separation of two stages, and limited for the alignment of non-monotonic sequences that find diverse applications in molecular biology [Löytynoja and Goldman, 2005], natural language processing [Barzilay and Lee, 2003], historic linguistics [Prokić et al., 2009], and computer vision [Caspi and Irani, 2000].

To address these limitations, we propose an end-to-end differentiable neural architecture, which we call NeuMATCH [Dogan et al., 2018], that considers more than the local similarities for heterogeneous sequence alignment. Inspired by LSTM-powered shift-reduce language parsers [Dyer et al., 2015; Honnibal and Johnson, 2015], we augment LSTM networks with stack operations, such as *pop* and *push*. The advantage of this setup is that the most relevant video clips, sentences, and historic records are always positioned closest to the prediction.

The NeuMATCH architecture represents the current state of the workspace using four Long Short-term Memory (LSTM) chains: two for the partially aligned sequences, one for the matched content, and one for historical alignment decisions. The four recurrent LSTM networks collectively capture the

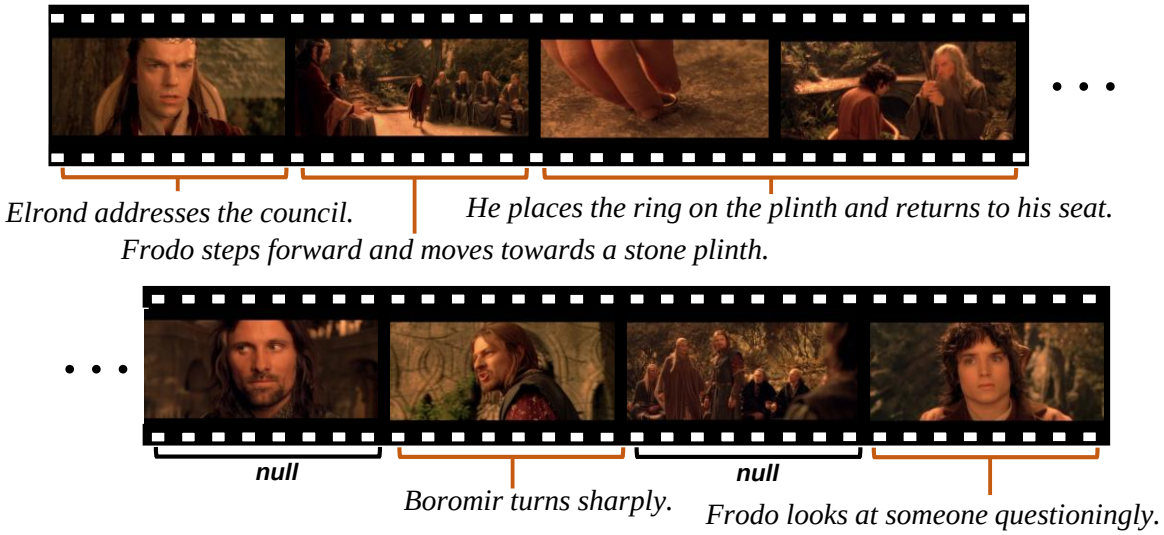


Figure 4.1: *An example alignment between clip sequence and text sequence (from the dataset HM-2 in Section 4.2).*

decision context, which is then classified into one of the available alignment actions. Compared to the traditional two-stage solution, the network can be optimized end-to-end. In addition, the previously matched content and the decision history inform future alignment decisions in a non-Markov manner. For example, if we match a person's face with the name Frodo at the beginning of a movie, we should be able to identify the same person again later (Figure 4.1). Alternatively, if the input sequences are sampled at different rates (e.g., every third video clip is matched to text), the decision history can help to discover and exploit such regularities.

Although the proposed framework can be applied to different types of sequential data, in this chapter, we focus on the alignment of video and textual sequences, especially those containing narrative content like movies. The reason for choosing narrative content is that it is among the most challenging for computational understanding due to a multitude of causal and temporal interactions between events [Sheinfeld et al., 2016]. Disambiguation is difficult with needed contextual information positioned far apart. Thus, narrative contents make an ideal application and testbed for alignment algorithms.

The contributions of this chapter are two-fold. First, we propose a novel end-to-end neural framework for heterogeneous multi-sequence alignment. Unlike prior methods, our architecture is able to take into account rich context when making alignment decisions. Extensive experiments illustrate that the framework *significantly* outperforms traditional baselines in accuracy. Sec-

ond, we annotate a new dataset[1] containing movie summary videos and share it with the research community.

4.1 Algorithm

4.1.1 Overview

We now present NeuMATCH, a neural architecture for temporal alignment of heterogeneous sequences. While the network is general, for this thesis we focus specifically on the video and textual sequence alignment. The video sequence consists of a number of consecutive video clips $\mathcal{V} = \{V_i\}_{i=1...N}$. The textual sequence consists a number of consecutive sentences $\mathcal{S} = \{S_i\}_{i=1...M}$. Our task is to align these two sequences by, for example, finding a function π that maps an index of the video segment to the corresponding sentence: $\langle V_i, S_{\pi(i)} \rangle$. An example input for our algorithm can be a movie segmented into individual shots and the accompanying movie script describing the scenes and actions, which are broken down into sentences (Figure 4.1). The video segmentation could be achieved using any shot boundary detection algorithm; NeuMATCH can handle one-to-many matching caused by over-segmentation.

We observe that the most difficult sequence alignment problems exhibit the following characteristics. First, heterogeneous surface forms, such as video and text, can conceal the true similarity structure, which suggests a satisfactory understanding of the entire content may be necessary for alignment. Second, difficult problems contain complex correspondence like many-to-one matching and unmatched content, which the framework should accommodate. Third, contextual information that are needed for learning the similarity metric are scattered over the entire sequence. Thus, it is important to consider the history and the future when making the alignment decision and to create an end-to-end network where gradient from alignment decisions can inform content understanding and similarity metric learning.

The NeuMATCH framework copes with these challenges by explicitly representing the state of the entire workspace, including the partially matched input sequences and historic alignment decisions. The representation employs four LSTM recurrent networks, including the input video sequence (Video Stack), the input textual sequence (Text Stack), previous alignment actions (Action Stack) as well as previous alignments themselves (Matched Stack). Figure 4.2 shows the NeuMATCH architecture. The final hidden

[1] https://github.com/pelindogan/NeuMATCH

states can be considered to encode information throughout the sequences. The concatenated hidden states are classified into one of the available alignment actions, which subsequently modifies the content of these LSTM networks.

We learn a function that maps the state of workspace Ψ_t to an alignment action A_t at every time step t. The action A_t manipulates the content of the LSTM networks, resulting in a new state Ψ_{t+1}. Executing a complete sequence of actions produces an alignment of the input. The reader may recognize the similarity with policy gradient methods [Sutton and Barto, 2017]. As the correct action sequence is unique in most cases and can be easily inferred from the ground-truth labels, in this chapter, we adopt a supervised learning approach.

The alignment actions may be seen as stack operations because they either remove or insert an element at the first position of the LSTM network (except for non-monotonic matching discussed in Appendix 4.1.3). For example, elements at the first position can be removed (*popped*) or *matched*. When two elements are matched, they are removed from the input stacks and stored in the Matched Stack.

4.1.2 Language and Visual Encoders

We first create encoders for each video clip and each sentence. After that, we perform an optional pre-training step to jointly embed the encoded video clips and sentences into the same space. While the pre-training step produces a good initialization, the entire framework is trained end-to-end, which allows the similarity metric to be specifically optimized for the alignment task.

Video Encoder. We extract features using the activation of the first fully connected layer in the VGG-16 network [Simonyan and Zisserman, 2014], which produces a 4096-dim vector per frame. As each clip is relatively short and homogeneous, we perform mean pooling over all frames in the video, yielding a feature vector for the entire clip. This vector is transformed with three fully connected layers using the ReLU activation function, resulting in encoded video vector v_i for the i^{th} clip.

Sentence Encoder. The input text is parsed into sentences $S_1 \dots S_M$, each of which contains a sequence of words. We transform each unique word into an embedding vector pre-trained using GloVe [Pennington et al., 2014]. The entire sentence is then encoded using a 2-layer LSTM recurrent network,

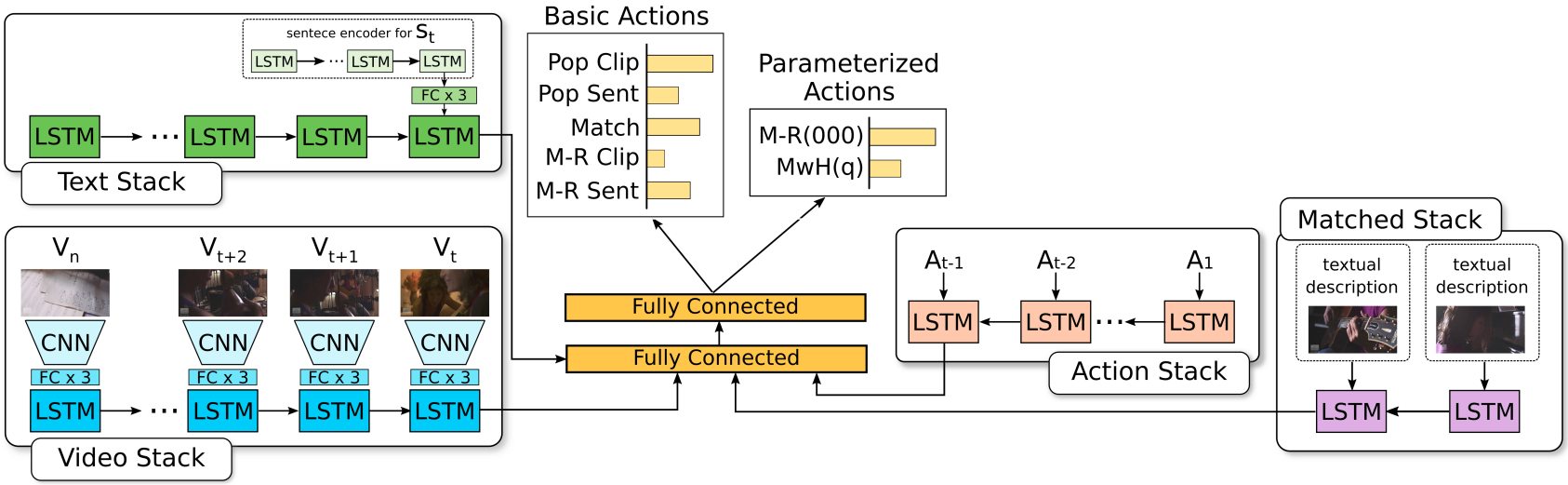


Figure 4.2: *The proposed NeuMATCH neural architecture. The current state as described by the four LSTM chains is classified into one of the alignment decisions. Parameterized actions are explained and illustrated in Section 4.1.3.*

where the hidden state of the first layer, $h_t^{(1)}$, is fed to the second layer:

$$h_t^{(1)}, c_t^{(1)} = \text{LSTM}(x_t, h_{t-1}^{(1)}, c_{t-1}^{(1)}) \tag{4.1a}$$

$$h_t^{(2)}, c_t^{(2)} = \text{LSTM}(h_t^{(1)}, h_{t-1}^{(2)}, c_{t-1}^{(2)}) \ , \tag{4.1b}$$

where $c_t^{(1)}$ and $c_t^{(2)}$ are the memory cells for the two layers, respectively; x_t is the word embedding for time step t. The sentence is represented as the vector obtained by the transformation of the last hidden state $h_t^{(2)}$ by three fully connected layers using ReLU activation function.

Encoding Alignment and Pre-training

Due to the complexity of the video and textual encoders, we opt for pre-training that produces a good initialization for subsequent end-to-end training. For a ground-truth pair (V_i, S_i), we adopt an asymmetric similarity proposed by [Vendrov et al., 2015]

$$F(v_i, s_i) = -||\max(0, v_i - s_i)||^2 \ . \tag{4.2}$$

This similarity function takes the maximum value 0, when s_i is positioned to the upper right of v_i in the vector space. That is, $\forall j, s_{i,j} \geq v_{i,j}$. When that condition is not satisfied, the similarity decreases. In [Vendrov et al., 2015], this relative spatial position defines an entailment relation where v_i entails s_i. Here the intuition is that the video typically contains more information than being described in the text, so we may consider the text as entailed by the video.

We adopt the following ranking loss objective by randomly sampling a contrastive video clip V' and a contrastive sentence S' for every ground truth pair. Minimizing the loss function maintains that the similarity of the contrastive pair is below true pair by at least the margin α.

$$\begin{aligned}\mathcal{L} = \sum_i \Big(& \mathbb{E}_{v' \neq v_i} \max\left\{0, \alpha - F(v_i, s_i) + F(v', s_i)\right\} \\ & + \mathbb{E}_{s' \neq s_i} \max\left\{0, \alpha - F(v_i, s_i) + F(v_i, s')\right\} \Big)\end{aligned} \tag{4.3}$$

Note the expectations are approximated by sampling.

4.1.3 The Alignment Network

With the similarity metric between video and text acquired by pre-training, a naive approach for alignment is to maximize the collective similarity over the matched video clips and sentences. However, doing so ignores the temporal structures of the two sequences and can lead to degraded performance. NeuMATCH considers the history and the future by encoding input sequences and decision history with LSTM networks.

The central idea is that we can store historic information and the future portion of the sequences to be matched in LSTM networks. The final hidden state of the network can be considered to encode information throughout the sequence. The concatenated hidden states are classified into one of the available alignment actions, which subsequently modifies the content of these LSTM networks. We first introduce the four LSTM stacks used by the NeuMATCH framework. The complete framework is illustrated in Figure 4.2.

LSTM Stacks

At time step t, the first stack contains the sequence of video clips yet to be processed $V_t, V_{t+1}, \ldots, V_N$. The direction of the LSTM goes from V_N to V_t, which allows the information to flow from the future clips to the current clip. We refer to this LSTM network as the video stack and denote its hidden state as h_t^V. Similarly, the text stack contains the sentence sequence yet to be processed: $S_t, S_{t+1}, \ldots, S_M$. Its hidden state is h_t^S.

The third stack is the action stack, which stores all alignment actions performed in the past. The actions are denoted as $A_{t-1}, \ldots, A_1$ and are encoded as one-hot vectors $a_{t-1}, \ldots, a_1$. The reason for including this stack is to capture patterns in the historic actions. Different from the first two stacks, the information flows from the first action to the immediate past with the last hidden state being h_{t-1}^A.

The fourth stack is the matched stack, which contains only the texts and clips that are matched previously and places the last matched content at the top of the stack. We denote this sequence as $R_1, \ldots, R_L$. Similar to the action stack, the information flows from the past to the present. In this chapter, we consider the case where one sentence s_i can match with multiple video clips $v_1, \ldots, v_K$. Since the matched video clips are probably similar in content, we perform mean pooling over the video features $v_i = \sum_j^K v_j / K$. The input to the LSTM unit is hence the concatenation of the two modalities $r_i = [s_i, v_i]$. The last hidden state of the matched stack is h_{t-1}^M.

	Video Stack	Text Stack	Matched Stack	Action Stack
Initial	ⓐⓑⓒ	①②③		
Pop Clip	ⓑⓒ	①②③		PC
Pop Sent	ⓐⓑⓒ	②③		PS
Match	ⓑⓒ	②③	[ⓐ①]	M
Match-Retain-C	ⓐⓑⓒ	②③	[ⓐ①]	MRC
Match-Retain-S	ⓑⓒ	①②③	[ⓐ①]	MRS

Table 4.1: *The basic action inventory and their effects on the stacks. Square brackets indicate matched elements.*

Action Prediction

At every time step, the state of the four stacks is $\Psi_t = (V_{t^+}, S_{t^+}, A_{(t-1)^-}, R_{1^+})$, where we use the shorthand X_{t^+} for the sequence $X_t, X_{t+1}, \ldots$ and similarly for X_{t^-}. Ψ_t can be approximately represented by the LSTM hidden states. Thus, the conditional probability of alignment action A_t at time t is

$$P(A_t|\Psi_t) = P(A_t|h_t^V, h_t^S, h_{t-1}^A, h_{t-1}^M) \tag{4.4}$$

The above computation is implemented as a softmax operation after two fully connected layers with ReLU activation on top of the concatenated state $\psi_t = [h_t^V, h_t^S, h_{t-1}^A, h_{t-1}^M]$. In order to compute the alignment of entire sequences, we apply the chain rule.

$$P(A_1, \ldots, A_N|\mathcal{V}, \mathcal{S}) = \prod_{t=1}^{N} P(A_t|A_{(t-1)^-}, \Psi_t) \tag{4.5}$$

The probability can be optimized greedily by always choosing the most probable action or using beam search. The classification is trained in a supervised manner. From a ground truth alignment of two sequences, we can easily derive a correct sequence of actions, which are used in training. In the infrequent case when more than one correct action sequence exist, one is randomly picked. The training objective is to minimize the cross-entropy loss at every time step.

Alignment Actions

The *Pop Clip* action removes the top element, V_t, from the video stack. This is desirable when V_t does not match any element in the text stack. Analogously, the *Pop Sentence* action removes the top element in the text stack, S_t.

The *Match* action removes both V_t and S_t, matches them, and pushes them to the matched stack. The actions *Match-Retain Clip* and *Match-Retain Sentence* are only used for one-to-many correspondence. When many sentences can be matched with one video clip, the *Match-Retain Clip* action pops S_t, matches it with V_t and pushes the pair to the matched stack, but V_t stays on the video stack for the next possible sentence. To pop V_t, the *Pop Clip* action must be used. The *Match-Retain Sentence* action is similarly defined. In this formulation, matching is always between elements at the top of the stacks.

It is worth noting that the five actions do not have to be used together. A subset can be picked based on knowledge about the sequences being matched. For example, for one-to-one matching, if we know some clips may not match any sentences, but every sentence have at least one matching clip, we only need *Pop Clip* and *Match*. Alternatively, consider a one-to-many scenario where (1) one sentence can match multiple video clips, (2) some clips are unmatched, and (3) every sentence has at least one matching clip. We need only the subset *Pop Clip*, *Pop Sentence*, and *Match-Retain Sentence*. It is desirable to choose as few actions as possible, because it simplifies training and reduces the branching factor during inference.

Discussion. The utility of the action stack becomes apparent in the one-to-many setting. As discussed earlier, to encode an element R_i in the matched stack, features from different video clips are mean-pooled. As a result, if the algorithm needs to learn a constraint on how many clips can be merged together, features from the matched stack may not be effective, but features from action stack would carry the necessary information. The alignment actions discussed in the above section allow monotonic matching for two sequences, which is the focus of this chapter and experiments. We discuss extensions that allow multi-sequence matching as well as non-monotonic matching in Section 4.1.3.

Parameterized Actions

The basic action inventory tackles the alignment of two sequences. The alignment of more than two sequences simultaneously, like video, audio, and textual sequences, requires an extension of the action inventory. To this end, we introduce a parameterized *Match-Retain* action. For three sequences, the parameters are a 3-bit binary vector where 1 indicate the top element from this sequence is being matched and 0 otherwise. Table 4.2 shows one example using the parameterized *Match-Retain*. For instance, to match the top elements from Sequence A and B, the action is *Match-Retain* (110). The parameters are implemented as three separate binary predictions.

	Seq A	Seq B	Seq C	Matched Stack
Initial	ⓐⓑⓒ	①②③	ⓧⓨⓩ	
1. M-R(110)	ⓐⓑⓒ	①②③	ⓧⓨⓩ	[ⓐ①]
2. Pop A	ⓑⓒ	①②③	ⓧⓨⓩ	[ⓐ①]
3. Pop B	ⓑⓒ	②③	ⓧⓨⓩ	[ⓐ①]
4. M-R(011)	ⓑⓒ	②③	ⓧⓨⓩ	[②ⓧ][ⓐ①]

Table 4.2: *An example action sequence for aligning three sequences.*

The use of parameterized actions further enables non-monotonic matching between sequences. In all previous examples, matching only happens between the stack tops. Non-monotonic matching is equivalent to allowing stack top elements to match with any element on the matched stack. We propose a new parameterized action *Match-With-History*, which has a single parameter q that indicates position on the matched stack. To deal with the fact that the matched stack has a variable length, we adopt the indexing method from Pointer Networks [Vinyals et al., 2015a]. The probability of choosing the i^{th} matched element r_i is

$$P(q = i|\Psi_t) = \frac{\exp(f(\psi_t, r_i))}{\sum_{j=0}^{L} \exp(f(\psi_t, r_j))} \tag{4.6a}$$

$$f(\psi_t, r_i) = v^\top \tanh\left(W_q \begin{bmatrix} \psi_t \\ r_i \end{bmatrix}\right) \tag{4.6b}$$

where the matrix W_q and vector v are trainable parameters and L is the length of the matched stack.

4.2 Experimental Evaluation

We evaluate NeuMATCH on semi-synthetic and real datasets, including a newly annotated, real-world YouTube Movie Summaries (YMS) dataset. Table 4.3 shows the statistics of the datasets used.

4.2.1 Setup and Training

For the joint pre-training, we use 500 dimensions for the LSTM sentence encoder and 300 for the joint embeddings. The dimensions of the word and image embedding are 300 and 4096, respectively, while the margin in the

	HM-1	HM-2	YMS
# words	4,196,633	4,198,021	54,326
# sent.	458,557	458,830	5,470
# avg. words/sent.	9.2	9.1	9.5
# clips	1,788,056	1,788,056	15,183
# video	22,945	22,931	94
# avg clips/video	77.9	77.9	161.5
# avg sent./video	20.0	20.0	58.2
# clip/sent. (mean(var))	2.0(0.33)	2.0(0.33)	2.6(8.8)

Table 4.3: *Summary statistics of the datasets.*

ranking objective function is $\alpha = 0.05$. L_2 regularization is used to prevent over-fitting. The batch size is set to 32 and the number of contrastive samples is 31 for every positive pair. The model is trained with the Adam optimizer using a learning rate of 10^{-4} and gradient clipping of 2.0. Early stopping on the validation set is used to avoid over-fitting.

The alignment network uses 300 dimensions for the video and text stacks, 20 dimensions for the matched stack and 8 for the history stack. Optionally, we feed two additional variables into the fully connected layer: the numbers of elements left in the video and text stacks to improve the performance on very long sequences in the YMS dataset. The alignment network is first trained with the encoding networks fixed with a learning rate of 0.001. After that, the entire model is trained end-to-end with a learning rate of 10^{-5}. For HM-0, HM-1, and HM-2, we use the original data split of LSMDC. For YMS, we use a 80/10/10 split for training, validation and test sets.

Details of Video Segmentation The video segmentation can be achieved using any shot boundary detection algorithm. In this work, we segment the input videos into video clips by a Python/OpenCV-based scene detection program[2] that uses threshold/content on a given video. For the parameters, we choose the *content-aware* detection method with the *threshold* of 20 and *minimum length* of 5 frames. Having a low threshold and minimum length usually results in over-segmentation. However, NeuMATCH can handle this resulting over-segmentation with the ability of one-to-many matching.

[2]https://github.com/Breakthrough/PySceneDetect

	HM-1				HM-2			
	MD	*CTW*	*DTW*	***Ours***	*MD*	*CTW*	*DTW*	***Ours***
clips	6.4	13.4	13.3	**69.7**	2.5	12.9	13.0	**40.6**
sents.	15.8	21.3	41.7	**58.6**	15.6	25.1	34.2	**43.7**

Table 4.4: *Accuracy of clips and sentences for the 2-action model. Datasets require the detection of* null *clips.*

4.2.2 Datasets

We create the datasets HM-1 and HM-2 based on the LSMDC data [Rohrbach et al., 2015], which contain matched clip-sentence pairs. The LSMDC data contain movie clips and very accurate textual descriptions, which are originally intended for the visually impaired. We generate video and textual sequences in the following way: First, video clips and their descriptions in the same movie are collected sequentially, creating the initial video and text sequences. For HM-1, we randomly insert video clips from other movies into each video sequence. In order to increase the difficulty of alignment and to make the dataset more realistic, we select confounding clips that are similar to the neighboring clips. After randomly choosing an insertion position, we sample 10 video clips and select the most similar to its neighboring clips, using the pre-trained similarity metric (Section 4.1.2). An insertion position can be 0-3 clips away from the last insertion. For HM-2, we randomly delete sentences from the collected text sequences. A deletion position is 0-3 sentences from the last deletion. At this point, HM-1 and HM-2 does not require one-to-many matching, which is used to test the 2-action NeuMATCH model. To allow one-to-many matching, we further randomly split every video clip into 1-5 smaller clips. As a result, the datasets can be used to test the 3-action NeuMATCH model.

YMS dataset. We create the YMS dataset from the YouTube channels *Movie Spoiler Alert* and *Movies in Minutes*, where a narrator orally summarizes movies alongside clips from the actual movie. Two annotators transcribed the audio and aligned the narration text with video clips. The YMS dataset is the most challenging for several reasons: (1) The sequences are long. On average, a video sequence contains 161.5 clips and a textual sequence contains 58.2 sentences. (2) A sentence can match a long sequence of (up to 45) video clips. (3) Unlike LSMDC, YMS contains rich textual descriptions that are intended for storytelling; they are not always faithful descriptions of the video, which makes YMS a challenging benchmark.

4.2.3 Performance Metrics

The ground truth alignment between video and text inputs is performed at the shot level, where each sentence has one or more corresponding shots. Due to the capability of our framework, the video sequences can even be composed of over segmented shots, since it can perform one-to-many matchings. Similar to the existing work in retrieval [Tapaswi et al., 2015], [Bojanowski et al., 2015], [Zhu et al., 2015], we focus our evaluation on recall, and Jaccard index [Jaccard, 1912],[Van Rijsbergen, 1977] which quantifies the difference between the ground truth assignment and the prediction by computing the ratio of intersection over union (IoU).

For the experiments that require one-to-one matching, the evaluation is straight-forward since it is binary. We used recall of clips meaning what percentage of clips are matched correctly either to a sentence or to *null* status, and recall of sentences in a similar way. For one-to-many matching, where one sentence can match multiple clips, we cannot use the same accuracy for sentences. Instead, we turn to the Jaccard Index, which measures the overlap between the predicted range and the ground truth of video clips using the intersection over union (IoU).

4.2.4 Baselines

We create three baselines, Minimum Distance (MD), Dynamic Time Warping (DTW), and Canonical Time Warping (CTW). All baselines use the same jointly trained language-visual neural network encoders (Section 4.1.2), which are carefully trained and exhibit strong performance. Due to space constraints, we discuss implementation details in the supplementary material.

The MD method matches the most similar clip-sentence pairs which have the smallest distance compared to the others. We artificially boost this baseline using specific optimization for the two accuracy measures. For evaluation on video clips, we match every clip with the most similar sentence, but if the distance is greater than the threshold 0.7, we consider the clip to be unmatched (i.e., a *null clip*). For sentence accuracy, we match every sentence with the most similar clip and do not assign *null* sentences.

DTW computes the optimal path on the distance matrix. It uses the fact that the first sentence is always matched with the first clip, and the last sentence is always matched to the last clip, so the shortest path is between the upper

	HM-0				HM-1			
	MD	*CTW*	*DTW*	***Ours***	*MD*	*CTW*	*DTW*	***Ours***
clips	20.7	26.3	50.6	**63.1**	10.5	6.8	17.6	**65.0**
sents IoU	23.0	25.4	42.8	**55.3**	5.7	7.3	18.4	**44.1**

	HM-2				YMS			
	MD	*CTW*	*DTW*	***Ours***	*MD*	*CTW*	*DTW*	***Ours***
clips	10.6	6.9	18.0	**37.7**	4.0	5.0	10.3	**12.0**
sents IoU	9.0	7.6	18.9	**20.0**	2.4	3.6	7.5	**10.4**

Table 4.5: *Alignment performance for 3-action model given in percentage (%) over all data. Datasets HM-1, HM-2, and YMS require the detection of null clips and one-to-many matchings of the sentences. HM-0 only requires one-to-many matching of sentences.*

left corner and lower right corner of the distance matrix. Note this is a constraint that NeuMATCH is not aware of. In order to handle null clips, we make use of the threshold again. In the case that one sentence is matched with several clips, the clips whose distances with the sentence are above the threshold will be assigned to null. We manually tuned the threshold to maximize the performance of all baselines. For CTW, we adopt the source code provided in [Zhou and De la Torre, 2016] with the same assignment method as DTW.

4.2.5 Ablation Studies

In order to understand the benefits of the individual components of NeuMATCH, we perform an ablated study where we remove one or two LSTM stacks from the architecture. The model *No Act&Hist* lacks both the action stack and the matched stack in the alignment network. That is, it only has the text and the video stacks. The second model *No Action* and the third model *No History* removes the action stack and the matched stack, respectively. In the last model *No Input LSTM*, we directly feed features of the video clip and the sentence at the tops of the respective stacks into the alignment network. That is, we do not consider the influence of future input elements.

Table 4.6 shows the performance of four ablated models in the one-to-many setting. The four ablated models perform substantially worse than the complete model. This confirms our intuition that both the history and the future

	HM-1		HM-2	
	clips	**sent. IoU**	**clips**	**sent. IoU**
No Act&Hist	47.3	21.8	11.8	1.6
No Action	49.9	23.0	29.6	16.1
No History	57.6	33.4	28.3	17.0
No Input LSTMs	54.8	24.6	27.9	8.3
NeuMATCH	**65.0**	**44.1**	**37.7**	**20.0**

Table 4.6: *Performance of ablated models in the one-to-many setting (3-action model).*

play important roles in sequence alignment. We conclude that all four LSTM stacks contribute to NeuMATCH's superior performance.

4.2.6 Quantitative Results

Tables 4.4 and 4.5 show the performance under one-to-one and one-to-many scenarios, respectively. On the one-to-one versions of the datasets HM-1 and HM-2, NeuMATCH demonstrates considerable improvements over the best baselines. It improves clip accuracy by 56.3 and 27.6 percentage points and improves sentence accuracy by 16.9 and 9.5 points. Unlike CTW and DTW, NeuMATCH does not have a major gap between clip and sentence performance. We attribute this partially to its superior ability to detect null clips.

On the one-to-many versions of HM-1 and HM-2, as well as the YMS dataset, NeuMATCH again shows superior performance over the baselines. The advantage over the best baselines is 47.4, 19.7, and 1.7 points for clip accuracy, and 25.7, 1.1, and 2.9 for sentence IoU. Interestingly, NeuMATCH performs better on HM-1 than HM-2, but the other baselines are largely indifferent between the two datasets. This is likely due to NeuMATCH's ability to extract information from the matched stack. Since HM-1 is created by inserting random clips into the video sequence, the features of the inserted video clip match surrounding clips, but other aspects such as cinematography style may not match. This makes HM-1 easier for NeuMATCH because it can compare the inserted clip with those in the matched stack and detect style differences. It is worth noting that different cinematographic styles are commonly used to indicate memories, illusions, or imaginations. Being able to recognize such styles can be advantageous for understanding complex narrative content.

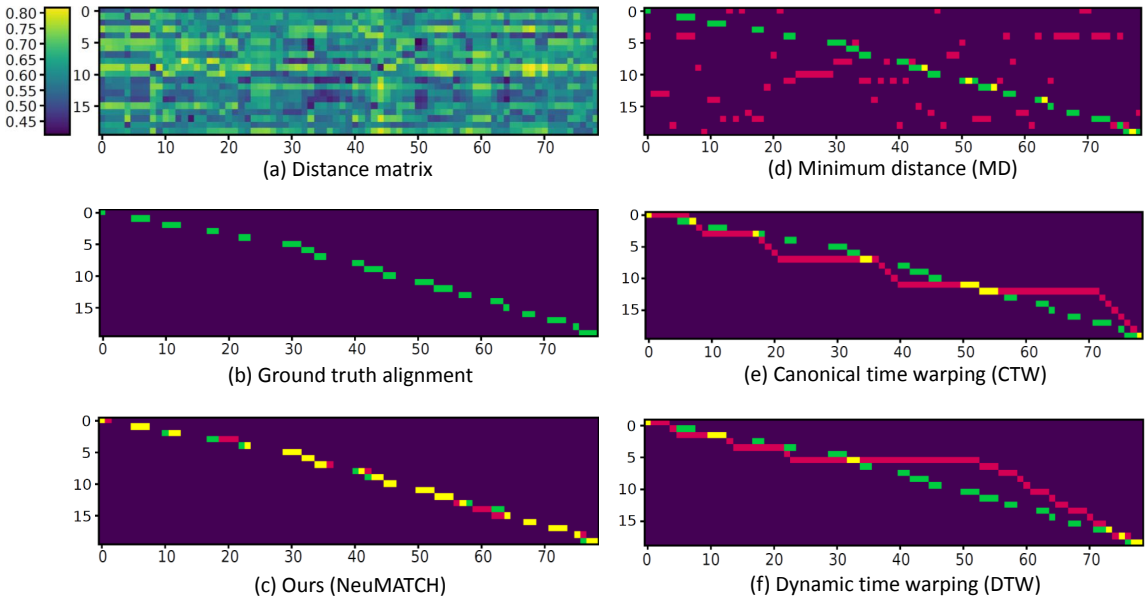


Figure 4.3: *An alignment problem from HM-2 and the results. The vertical and horizontal axes represent the text sequence (sentences) and video sequence (clips) respectively. Green, red and yellow respectively represent the ground-truth alignment, the predicted alignment, and the intersection of two.*

To further investigate NeuMATCH's performance without null clips, we additionally create a one-to-many dataset, HM-0, by randomly dividing every video clip into 1-to-5 smaller clips. Although NeuMATCH's advantage is reduced on HM-0, it's still substantial (12.5 points on both measures), showing that the performance gains are not solely due to the presence of null clips.

As we expect, the real-world YMS dataset is more difficult than HM-1 and HM-2. Still, we have a relative improvement of 17% on clip accuracy and 39% on sentence IoU over the closest DTW baseline. We find that NeuMATCH consistently surpasses conventional baselines across all experimental conditions. This clearly demonstrates NeuMATCH's ability to identify alignment from heterogenous video-text inputs that are challenging to understand computationally.

As a qualitative evaluation, Figure 4.3 shows an alignment example. The ground alignment goes from the top left (the first sentence and the first clip) to the bottom right (the last sentence and the last clip). Dots in green, red, and yellow represent the ground truth alignment, the predicted alignment, and the intersection of the two, respectively. In the ground truth path (e), some columns does not have any dots because those clips are not matched to anything. As shown in (a), the distance matrix does not exhibit any clear

alignment path. Therefore, MD, which uses only the distance matrix, performs poorly. The time warping baselines in (c) and (d) also notably deviate from the correct path, whereas NeuMATCH is able to recover most of the ground-truth alignment. Moreover, it has ability to catch the correct alignment even when it starts diverting at some point.

4.2.7 Qualitative Results

We show more alignment results computed by our approach on the datasets HM-1, HM-2, and YMS that require one-to-many matching and contain clips that do not match any sentences (i.e., *null* clips). For continuity, the all the results mentioned in the following are shown at the end of this chapter. For illustration purposes, each figure represents only a small portion (6-12 consecutive clips) of the entire aligned sequence. Each frame represents a video clip. The aligned sentences are shown with wide brackets below or above the clips.

Successful results for Hollywood Movies 1 (HM-1) The video sequences in HM-1 contain clips from other movies that are inserted into the original sequence, as explained in Section 4.2.2. Figure 4.4 shows a sequence from the movie *Jack and Jill*. The fifth frame is from the movie *This is 40*, which is successfully assigned as *null*. Note the last two frames have very similar content (two women in dresses) to the sentence *"With a fuzzy shawl and cap, and a ruffled skirt."*, but our algorithm was able to identify them correctly. Figure 4.5 shows a sequence from the movie *Blind Dating*. The one-to-many assignment for the last three clips is correctly identified even when there is a significant perspective and content change through the clips. Figure 4.6 shows a sequence from the movie *Juno*. The one-to-many assignment for the last three clips is correctly identified even when there is a significant perspective and content change through the clips.

Successful results for Hollywood Movies 2 (HM-2)

Each video sequence in HM-2 consists of consecutive clips from a single movie, where some sentences were discarded in order to create *null* clips. It still requires one-to-many matching of the sentences and the assignment of *null* clips. Figure 4.10 and Figure 4.7 shows sequences from the movies *Harry Potter and the Prisoner of Azkaban* and *Bad Santa*. In Figure 4.8, a sequence from the movie *The Ugly Truth* is given. The third clip contains a vodka bottle, which is mentioned in first sentence. The fourth and the fifth clips are very similar. However, the algorithm finds the correct alignment. In Figure 4.9, from *Super 8*, the boy and the bicycle are visible in both the second

and the third clips, but the headstones only appear in the third clip. The algorithm makes the correct decision. In Figure 4.11, from *Unbreakable*, the wheelchair is only visible in the last clip and the algorithm successfully picks that up.

Successful results for YouTube Movie Summaries (YMS)

In the YMS dataset, the sentences are longer than HM-1 and HM-2, and they tend to describe multiple events. We asked the annotators to break them down into small units, which allows them to precisely align the text with the video sequence. These sequences tend to be much more complex than HM-1 and HM-2. Examples are shown in Figure 4.13-4.15.

Failure Cases

We present two failure cases in Figure 4.16 and Figure 4.17 from the movies *Friends with Benefits* and *The Ugly Truth*, respectively. The ground truth is shown with green brackets and NeuMATCH's predictions are with orange brackets. In Figure 4.16, the first failure is that the second sentence is matched with two more clips, but the additional clips also contain the "railing" and the "water", which may have confused the algorithm. Similarly, the boat appears in the sixth and seventh clips, which may have caused the wrong alignment with the third sentence.

4.3 Summary

In this chapter, we propose NeuMATCH, an end-to-end neural architecture aimed at heterogeneous multi-sequence alignment, focusing on alignment of video and textural data. Alignment actions are implemented in our network as data moving operations between LSTM stacks. We show that this flexible architecture supports a variety of alignment tasks. Results on semi-synthetic and real-world datasets and multiple different settings illustrate superiority of this model over popular traditional approaches based on time warping. An ablation study demonstrates the benefits of using rich context when making alignment decisions. As a future work, The method can be improved more by experimenting on the extensions for the alignment of multiple sequences and non-monotonicity.

Figure 4.4: *From the movie Jack and Jill in dataset HM-1.*

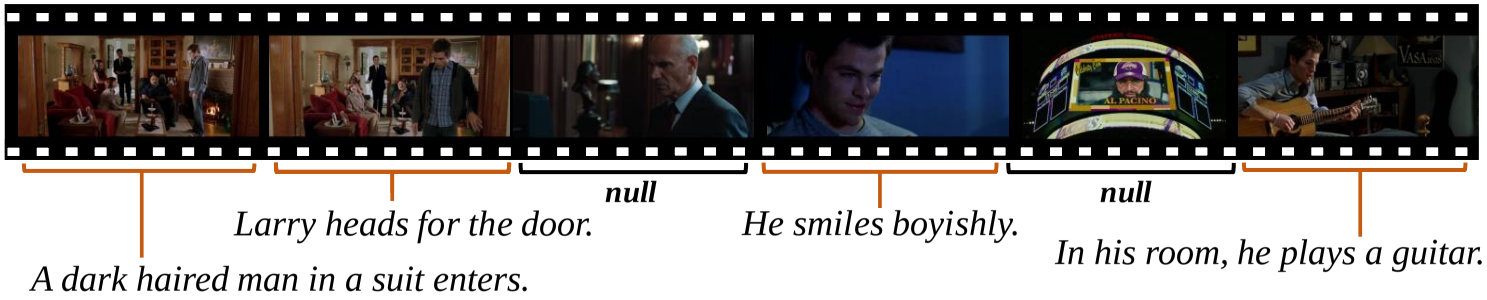


Figure 4.5: *From the movie Blind Dating in dataset HM-1*

Figure 4.6: *From the movie Juno in dataset HM-1.*

Figure 4.7: *From the movie Bad Santa in dataset HM-2.*

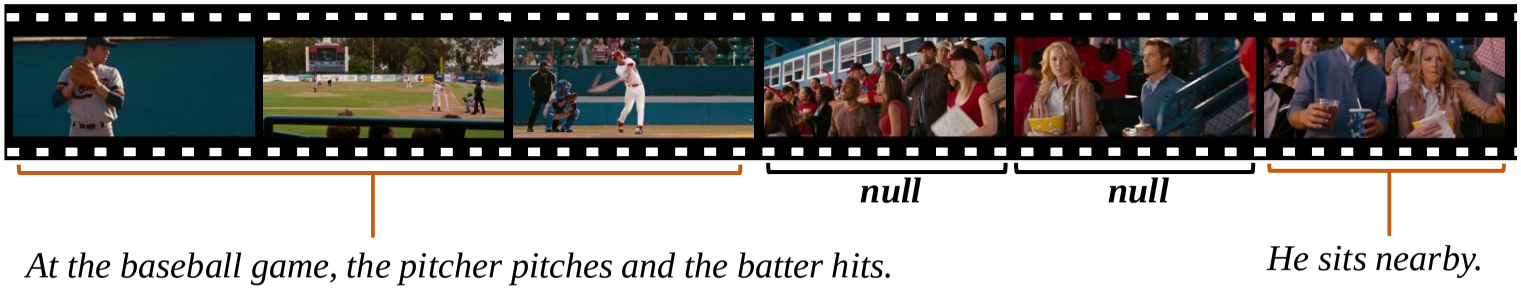


Figure 4.8: *From the movie The Ugly Truth in dataset HM-2*

Figure 4.9: *From the movie The Super 8 in dataset HM-2.*

Figure 4.10: *From the movie Harry Potter and the Prisoner of Azkaban in dataset HM-2*

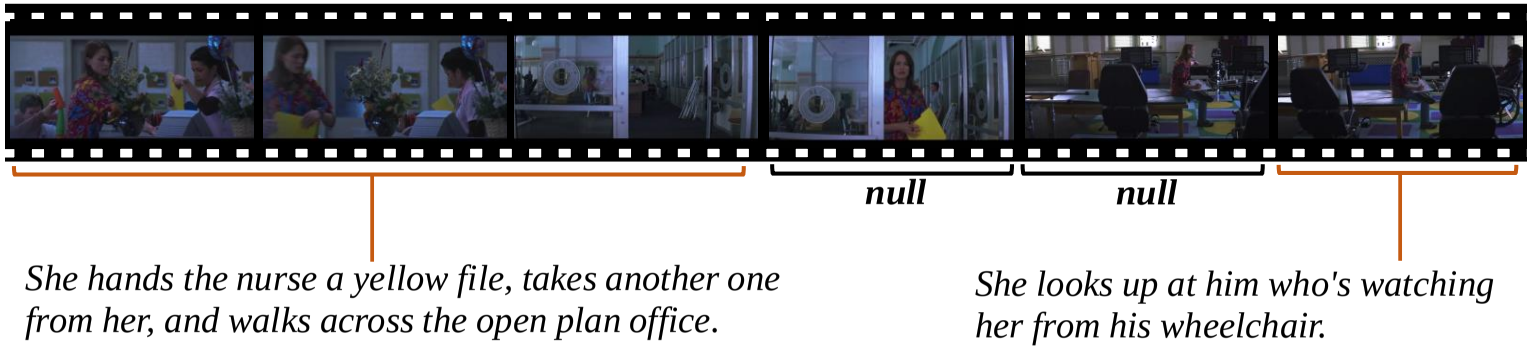


Figure 4.11: *From the movie Unbreakable in dataset HM-2*

Figure 4.12: *From the movie Juno in dataset HM-2.*

Figure 4.13: *From the movie Doctor Strange in dataset YMS. The original video is available at* `https://www.youtube.com/watch?v=fZeW-KUXHKY`.

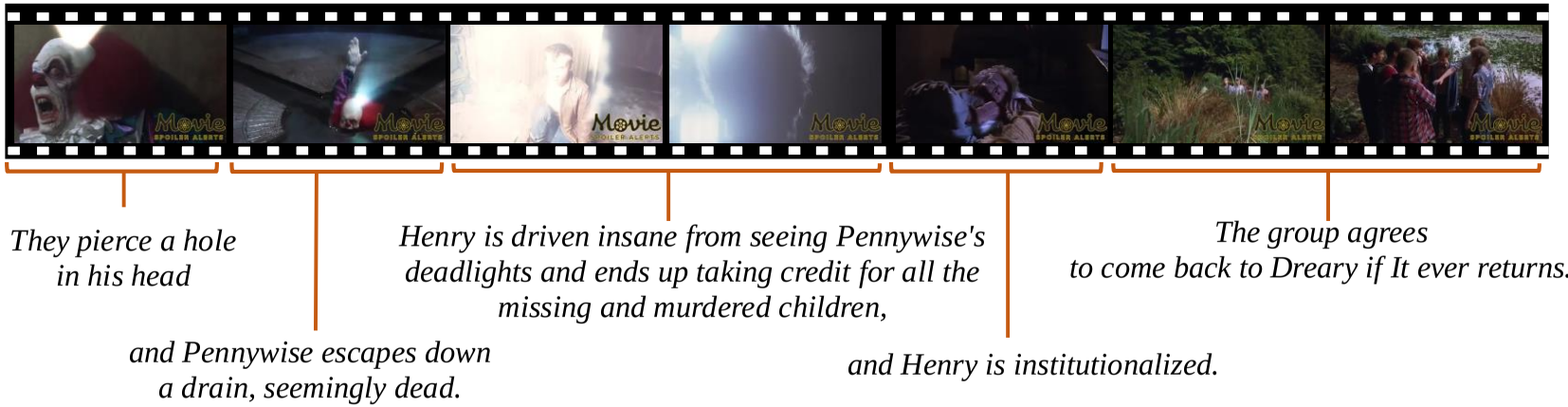


Figure 4.14: *From the movie It (1990) in dataset YMS. The original video is available at* `https://www.youtube.com/watch?v=c-sIoODkpuU`.

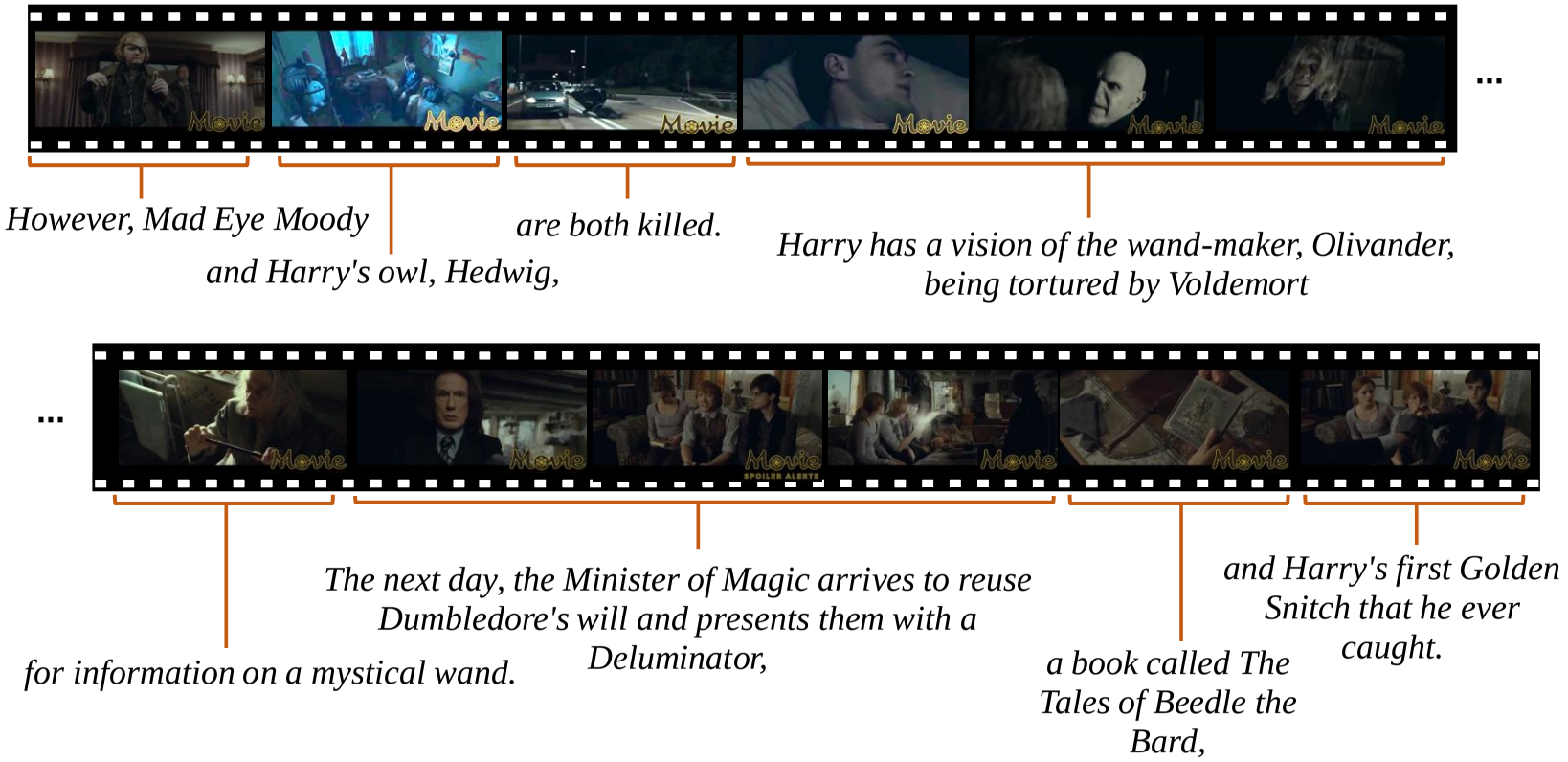


Figure 4.15: *From the movie Harry Potter and the Deathly Hallows in dataset YMS. The original video is available at* `https://www.youtube.com/watch?v=nfuRErj9TkY`.

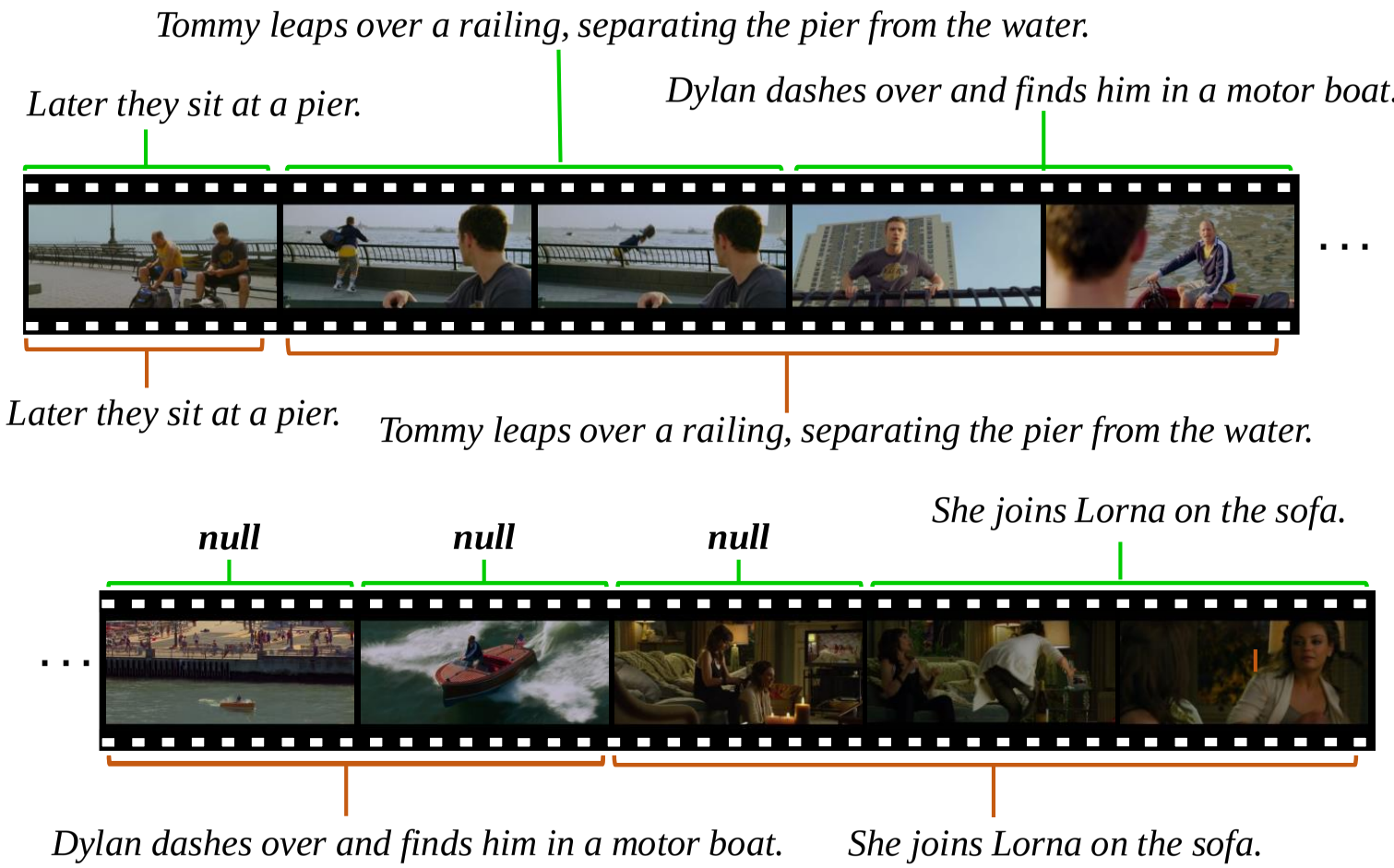


Figure 4.16: *From the movie Friends with Benefits in dataset HM-2.*

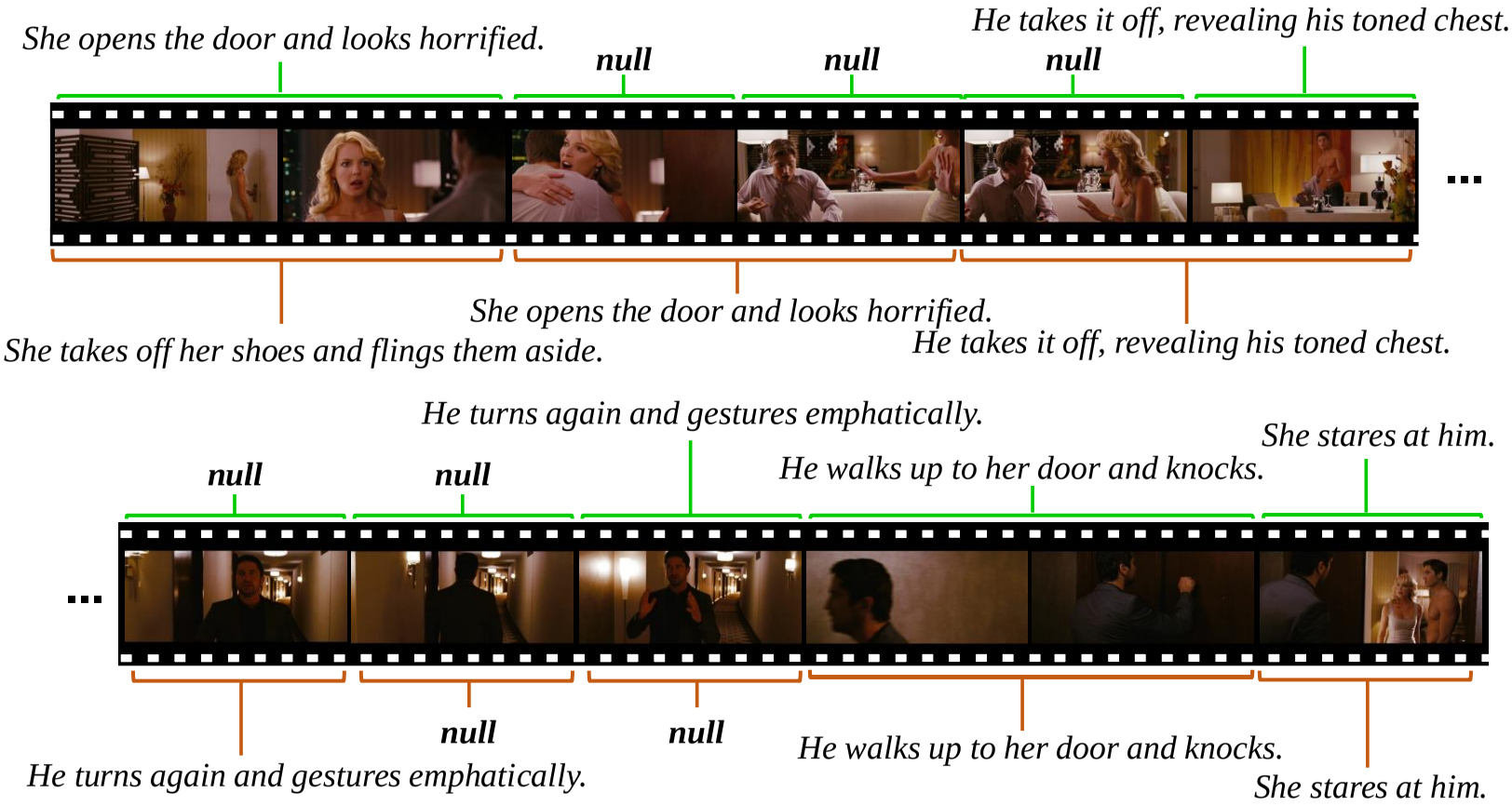


Figure 4.17: *From the movie The Ugly Truth in dataset HM-2.*

CHAPTER 5

Neural Sequential Phrase Grounding

In the previous chapters, we have presented approaches for aligning video and text with narrative content, which mainly provides meta-data extraction at the granularity of shot-sentence level that could be used for various application as mentioned earlier. Taking this even further, a finer granularity level by aligning phrases to image (video frame) regions is the next natural step which poses the *phrase grounding* problem.

Consider image grounding for noun phrases from a given sentence: "*A lady sitting on a colorful decoration with a bouquet of flowers, that match her hair, in her hand.*" Note that while multiple *ladies* may be present in the image, the grounding of "*a colorful decoration*" uniquely disambiguates to which of these instances the phrase "*A lady*" should be grounded to. While contextual reference in the above example is spatial, other context, including visual maybe useful, e.g., between "*her hair*" and "*a bouquet of flowers*".

Conceptually similar contextual relations exist in object detection and have just started to be explored through the use of spatial memory [Chen and Gupta, 2017] and convolutional graph networks (CGNNs) [Chen et al., 2018b], [Yang et al., 2018]. Most assume orderless graph relationships among objects with transitive reasoning. In phrase grounding, on the other hand, the sentence, from which phrases are extracted, may provide implicate linguistic space- and time-order [Hazen, 2014]. We show that such ordering is useful as a proxy for sequentially contextualizing phrase grounding decisions. In other words, the phrase that appears *last* in the sentence is grounded first and is used as context for the next phrase grounding in *reverse* lexical order. This explicitly sequential process is illustrated in Figure 5.1. To

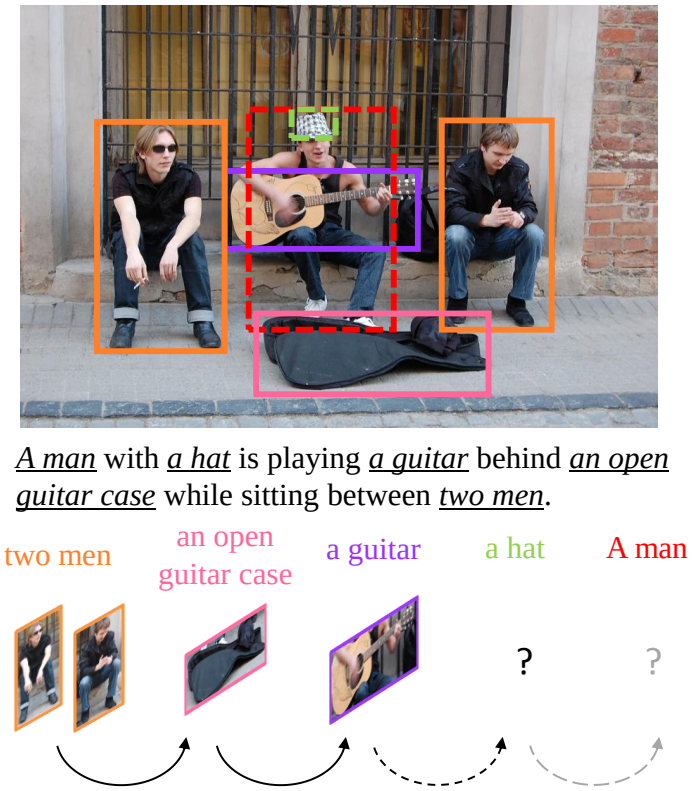


Figure 5.1: Illustration of SeqGROUND. *The proposed neural architecture performs phrase grounding sequentially. It uses the previously grounded phrase-image content to inform the next grounding decision (in reverse* lexical *order).*

our knowledge, our framework is the first to explore such sequential mechanism and architecture for phrase grounding.

In this chapter, expanding on the class of recent temporal alignment networks (e.g., Chapter 4, [Dogan et al., 2018]), that propose neural architectures where discrete alignment actions are implemented by moving data between stacks of Long Short-term Memory (LSTM) blocks, we develop a sequential *spatial* phrase grounding network that we call SeqGROUND [Dogan et al., 2019]. SeqGROUND encodes region proposals and all phrases into two stacks of LSTM cells, along with so-far grounded phrase-region pairings. These LSTM stacks collectively capture the context for the grounding of the next phrase.

5.1 Algorithm

5.1.1 Overview

We now present our neural architecture for grounding phrases in images. We assume that we need to ground multiple, potentially inter-related, phrases in each image. This is the case for the Flickr30k Entities dataset, where phrases/entities come from sentence parsing. Specifically, we parse the input sentence into a sequence of phrases $\mathcal{P} = \left\{P_j\right\}_{j=1...N}$ keeping the sentence order; *i.e.* $j = 1$ is the first phrase and $j = N$ is the last. For a typical sentence in Flickr30k, N is between 1 and 54. The input image I is used to extract region proposals in the form of bounding boxes. These bounding boxes are ordered to form a sequence $\mathcal{B} = \{B_i\}_{i=1...M}$. We discuss the ordering choices, for both $\mathcal{P}$ and $\mathcal{B}$, and their effects in Section 5.2.3. Our overall task is to ground phrases in the image by matching them to their corresponding bounding boxes, for example, finding a function π that maps an index of the phrase to its corresponding bounding boxes $\langle P_j, B_{\pi(j)}\rangle$. Our method allows many-to-many matching of the aforementioned input sequences. In other words, a single phrase can be grounded to multiple bounding boxes, or multiple phrases of the sentence can be grounded to the same bounding box.

Phrase grounding is a very challenging problem exhibiting the following characteristics. First, image and text are heterogeneous surface forms concealing the true similarity structure. Hence, satisfactory understanding of the entire language and visual content is needed for effective grounding. Second, relationships between phrases and boxes are complex. It is possible (and likely) to have many-to-many matchings and/or unmatched content (due to either lack of precision in the bounding box proposal mechanism or hypothetical linguistic references). Such scenarios need to be accommodated by the grounding algorithm. Third, contextual information that is needed for learning the similarity between phrase-box pairs are scattered over the entire image and the sentence. Therefore, it is important to consider all visual and textual context with a strong representation of their dependencies when making grounding decisions, and to create an end-to-end network, where gradient from grounding decisions can inform content understanding and similarity learning.

The SeqGROUND framework copes with these challenges by casting the problem as one of sequential grounding and explicitly representing the state of the entire decision *workspace*, including the partially grounded input phrases and boxes. The representation employs LSTM recurrent networks

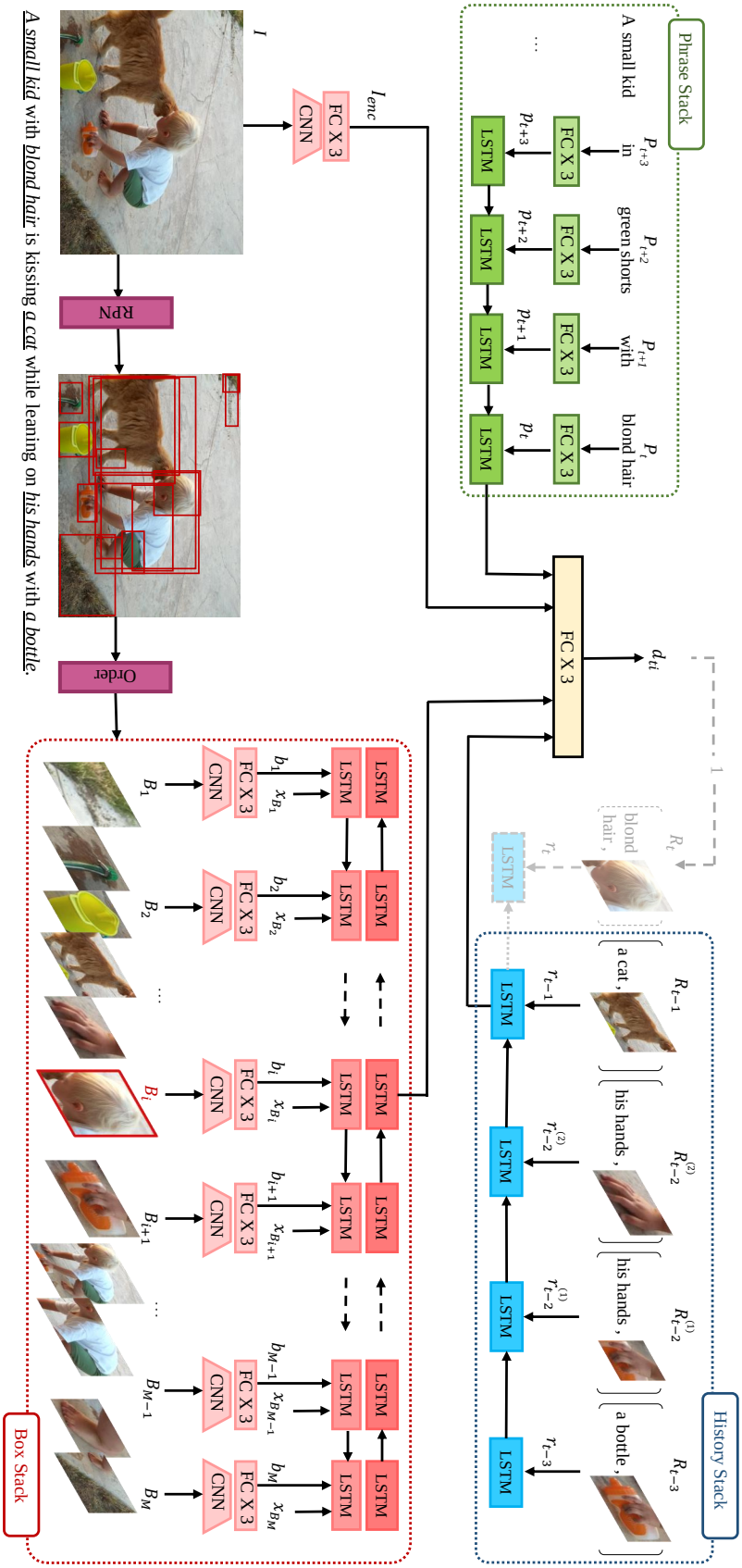


Figure 5.2: *SeqGROUND neural architecture.*

for region proposals, sentence phrases, and the previously grounded content, in addition to dense layers for the full image representation. Figure 5.2 shows the architecture of our framework, where the *phrase stack* contains the sequence of phrases yet to be processed in an order and encodes the linguistic dependencies. The *box stack* contains the sequence of bounding boxes that are ordered with respect to their locations in the image. The *history stack* contains the phrase-box pairs that are previously grounded. The grounding decisions for the input phrases are performed sequentially taking into account of the current states of these LSTM stacks in addition to full image representation. The new grounded phrase-box pairs are added to the top of the *history stack*.

We learn a function that maps the state of workspace Ψ_t to a grounding decision d_{ti} for the bounding box B_i at every time step t, which corresponds to a decision for phrase P_t. The decisions d_{ti} manipulates the content of the LSTM networks, resulting in a new state Ψ_{t+1}. Executing a complete sequence of decisions produces a complete alignment of the input phrases with the bounding boxes. As the correct decision sequence is unique in most cases and can be easily inferred from the ground-truth labels, in this framework, we adopt a supervised learning approach.

5.1.2 Language and Visual Encoders

We first create encoders for each phrase and each bounding box produced by a region proposal network (RPN). After that, we perform an optional pretraining step to jointly embed the encoded phrases and bounding boxes into the same latent space, as discussed in Section 5.1.3.

Phrase Encoder. The input caption is parsed into phrases $P_1 \dots P_N$, each of which contains a word or a sequence of words, using [Chen and Manning, 2014]. We transform each unique phrase into an embedding vector, by performing mean pooling over GloVe [Pennington et al., 2014] features of all its words. This vector is then transformed with three fully connected layers using the ReLU activation function, resulting in the encoded phrase vector p_j for the j^{th} phrase (P_j) of the input sentence.

Visual Encoder. For each proposed bounding box, we extract features using the activation of the first fully connected layer in the VGG-16 network [Simonyan and Zisserman, 2014], which produces a 4096-dim vector per region. This vector is transformed with three fully connected layers using the ReLU activation function, resulting in the encoded bounding box vector b_i

for the i^{th} bounding box (B_i) of the image. The visual encoder is also used to encode the full image I into I_{enc}.

5.1.3 The Grounding Network

Having the encoded phrases and boxes in the same embedding space, a naive approach for grounding would be maximizing the collective similarity over the grounded phrase-box pairs. However, doing so ignores the spatial structures and relations within the elements of the two sequences, and can lead to degraded performance. SeqGROUND performs grounding by encoding the input sequences and the decision history with stacks of recurrent networks. This implicitly allows the network to take into account all grounded as well as ungrounded proposed regions and phrases as context for the current grounding decision. We show in Section 5.2 that this leads to a significant boost in performance.

Recurrent Stacks

Considering the input phrases as a temporal sequence, we let the first stack contain the sequence of phrases yet to be processed $P_t, P_{t+1}, \ldots, P_N$, at the time step t. The direction of the stack goes from P_N to P_t, which allows the information to flow from the future phrases to the current phrase. We refer to this LSTM network as the *phrase stack* and denote its hidden state as h_t^P. The input to the LSTM unit is the phrase features in the latent space obtained by the phrase encoder (see Sec. 5.1.2).

The second stack is a bi-directional LSTM recurrent network that contains the sequence of bounding boxes $B_1, \ldots, B_M$ obtained by the RPN. The boxes are ordered from left to right considering their center on the horizontal axis for the forward network[1]. We refer to this bi-LSTM network as the *box stack* and denote its hidden state for the i^{th} box as h_i^B. The input to the LSTM unit is the concatenation of the box features in the latent space and the normalized location features $[b_i, x_{b_i}]$. Note that the state of the box stack does not change with respect to t. We keep all the boxes in the stack, since a box that is already used to ground a phrase can be used again to grounding another phrase later on.

[1]We experimented with alternative orderings, *e.g.*, max flow computed over pair-wise proposal IoU scores, but saw no appreciable difference in performance. Therefore for cleaner exposition we focus on simpler left-to-right ordering and corresponding results.

The third stack is the *history stack*, which contains only the phrases and the boxes that are previously grounded, and places the last grounded phrase-box pair at the top of the stack. We denote this sequence as $R_1, \ldots, R_L$. The information flows from the past to the present. The input to the LSTM unit is the concatenation of the two modalities in the latent space and the location features of the box. When a phrase p_j is grounded to multiple (K) boxes $b_{\pi(j)} = b_{(p_j,1)}, \ldots, b_{(p_j,K)}$, each grounded phrase-box pair becomes a separate input to the LSTM unit, keeping the spatial order of the boxes. For example, the vector $[p_j, b_{(p_j,1)}, x_{b_{(p_j,1)}}]$ will be the first vector to be pushed to the top of the history stack for the phrase p_j. The last hidden state of the history stack is h^R_{t-1}.

The *phrase stack* and *history stack* both perform encoding using a 2-layer LSTM recurrent network, where the hidden state of the first layer, $h^{(1)}_t$, is fed to the second layer:

$$h^{(1)}_t, c^{(1)}_t = \text{LSTM}(x_t, h^{(1)}_{t-1}, c^{(1)}_{t-1}) \tag{5.1a}$$

$$h^{(2)}_t, c^{(2)}_t = \text{LSTM}(h^{(1)}_t, h^{(2)}_{t-1}, c^{(2)}_{t-1}) \ , \tag{5.1b}$$

where $c^{(1)}_t$ and $c^{(2)}_t$ are the memory cells for the two layers, respectively; x_t is the input for time step t.

Image Context. In addition to the recurrent stacks, we also provide the encoded full image I to the network as an additional global context.

Decision Prediction

At every time step, the state of the three stacks is $\Psi_t = (P_{t^+}, B_t, R_{1^+})$, where we use the shorthand X_{t^+} for the sequence $X_t, X_{t+1}, \ldots$ and similarly for X_{t^-}. The LSTM hidden states can approximately represent Ψ_t. Thus, the conditional probability of grounding decision d_{ti}, which represents the decision for bounding box B_i with the phrase P_t is

$$Pr(d_{ti}|\Psi_t) = Pr(d_{ti}|h^P_t, h^B_i, h^R_{t-1}, I_{enc}). \tag{5.2}$$

In other words, at time step t, a grounding decision is made simultaneously for each box for the phrase at the top of the *phrase stack*. Although it may seem that these decisions are made in parallel independently, the hidden states of the *box stack* encode the relation and dependencies between all the boxes. The above computation is implemented as a sigmoid operation after three fully connected layers on top of the concatenated state

$\psi_t = [h_t^P, \{h_i^B\}, h_{t-1}^R, I_{enc}]$. ReLU activation is used between the layers. Further, each positive grounding decision will augment the *history stack*.

In order to ground the entire phrase sequence with the boxes, we apply the chain rule as follows:

$$Pr(D_1, \ldots, D_N | \mathcal{P}, \mathcal{B}) = \prod_{t=1}^{N} Pr(D_t | D_{(t-1)^-}, \Psi_t) \tag{5.3a}$$

$$Pr(D_t | \mathcal{P}, \mathcal{B}) = \prod_{i=1}^{M} Pr(d_{ti} | D_{(t-1)^-}, \Psi_t) \ , \tag{5.3b}$$

where D_t represents the set of all grounding decisions over all the boxes for the phrase P_t. The probability can be optimized greedily by always choosing the most probable decisions. The model is trained in a supervised manner. From a ground truth grounding of a box and a phrase sequence, we can easily derive the correct decisions, which are used in training. The training objective is to minimize the overall binary cross-entropy loss caused by the grounding decisions at every time step for each $\langle P_t, B_i \rangle$ with $i = 1, \ldots, M$.

Pre-training

As noted in Chapter 4, learning a coordinated representation (or similarity measure) between visual and text data, while also optimizing a decision network, is difficult. Thus, we adopt a pairwise pre-training step to coordinate the phrase and visual encoders to achieve a good initialization for subsequent end-to-end training. Note that this is only done for pre-training; the final model is fully differentiable and is fine-tuned end-to-end.

For a ground-truth pair (P_k, B_k), we adopt an asymmetric similarity proposed by [Vendrov et al., 2015]

$$F(p_k, b_k) = -||\max(0, b_k - p_k)||^2 \ . \tag{5.4}$$

This similarity function, F, takes the maximum value 0, when p_k is positioned to the upper right of b_k in the vector space. When that condition is not satisfied, the similarity decreases. In [Vendrov et al., 2015], this relative spatial position defines an entailment relation where b_k entails p_k. Here, the intuition is that the image typically contains more information than being described in the text form, so we may consider the text as entailed by the image.

We adopt the following ranking loss objective by randomly sampling a contrastive bounding box B' and a contrastive phrase P' for every ground truth pair. Minimizing the loss function maintains that the similarity of the contrastive pair is below the true pair's by at least the margin α:

$$\begin{aligned}\mathcal{L} = \sum_i \Big(& \mathbb{E}_{b' \neq b_k} \max\left\{0, \alpha - F(b_k, p_k) + F(b', p_k)\right\} \\ & + \mathbb{E}_{p' \neq p_k} \max\left\{0, \alpha - F(b_k, p_k) + F(b_k, p')\right\} \Big)\end{aligned} \tag{5.5}$$

Note the expectations are approximated by sampling.

5.2 Experimental Evaluation

5.2.1 Setup and Training

We use Faster R-CNN [Ren et al., 2015] as an underlying bounding box proposal mechanism with ResNet50 as the backbone. The extracted bounding boxes are then sorted from left-to-right by their central x-coordinate to be fed into the Bi-LSTM network of the *box stack*. This way, the objects appearing close tend to be represented closer together, so that the *box stack* can represent the overall context better. Following the prior works (see Tab. 5.2), we assume that the noun phrases that are to be grounded have already been extracted from the descriptive sentences. We also use the intermediate words of the sentences together with the given noun phrases in the phrase stack to preserve the linguistic structure; this also results in a more complex train/test scenario. Note that we do not explicitly distinguish between the intermediate words, meaning that the network implicitly tries to ground them as well.

SeqGROUND is trained in two stages that differ in *box stack* input. In the first stage, we only feed the groundtruth instances to the *box stack*, which are coming from the dataset annotation, for an image. The boxes that have the same label as the phrase are considered as positive samples, while the remaining boxes as negative samples. This set-up provides an easier phrase grounding task due to the low number of input boxes which are contextually distinct and well-defined without being redundant. Thus, it provides a good initialization for the second stage where we use the box proposals by the RPN.

For the second stage, we map each bounding box, coming from the RPN, to the groundtruth instances with which it has IoU overlap equal to or greater than 0.7, and label them as positive samples for the current phrase. The remaining proposed boxes having IoU overlap less than 0.3 with the groundtruth instances are labeled as negative samples for that phrase. The labeled positive and negative samples are sorted and then fed into the Bi-LSTM network. It is possible to optimize for the loss function of all labeled boxes, but this will bias towards negative samples as they dominate. Instead, we randomly sample negative samples that contribute to the loss function in a batch, where the sampled positive and negative boxes have a ratio of 1:3. If the number of negative samples within a batch is not enough, we let all the samples in that batch contribute to the loss. In this way, the spatial context and dependencies are represented without gaps by the Bi-LSTM unit of the *box stack*, while preventing biasing towards negative grounding decisions. After the second stage of training, we adopt the standard hard negative mining method [Felzenszwalb et al., 2010], [Sung and Poggio, 1998] with a single pass on each training sample.

At test time, we use all the proposed boxes to feed them to the *box stack* after ordering them with respect to their locations. When multiple boxes are grounded to the same phrase, we apply non-maximum suppression with an IoU overlap threshold of 0.3, which is tuned on the validation set. In this way, multiple box results for the same instance of a phrase are discarded, while the boxes for different instances of the same phrase are kept.

5.2.2 Dataset and Metrics

We evaluate our approach on the Flickr30K Entities dataset [Plummer et al., 2015] which contains 31,783 images, each annotated with five sentences. For each sentence, the noun phrases are provided with their corresponding bounding boxes in the image. We use the same training/validation/test split as the prior work, which provides 1,000 images for validation, 1,000 for testing, and 29,783 images for training. It is important to note that a single phrase can have multiple groundtruth boxes, while a single box can match multiple phrases within the same sentence. Consistent with the prior work, we evaluate SeqGROUND with the ground truth bounding boxes. If multiple boxes are associated with a phrase, we represent the phrase as the union of all its boxes on the image plane. Following the prior work, successful grounding of a phrase requires predicted area to have at least 0.5 IoU (intersection over union) with the groundtruth area. Based on this criteria, our

	Components				
	Visual context	Bounding box	Phrase	History	**Accuracy**
MSB	none	simple	simple	none	43.85
MSBs	none	simple	simple	none	50.90
NH	global	bi-LSTM	LSTM	none	59.55
NI	none	bi-LSTM	LSTM	LSTM	60.34
SPv	global	bi-LSTM	simple	LSTM	57.94
SBv	global	simple	LSTM	LSTM	55.68
SPvBv	global	simple	simple	LSTM	53.75
SBvPvNH	global	simple	simple	none	52.91
SeqGROUND	global	bi-LSTM	LSTM	LSTM	**61.60**

Table 5.1: *Grounding accuracy of baselines and ablated models.*

measure of performance is grounding *accuracy*, which is the ratio of correctly grounded noun phrases.

5.2.3 Baselines and Ablation Studies

In order to understand the benefits of the individual components of our model, we perform an ablation study where certain stacks are either removed or modified. The model *NH* lacks the *history stack* where the previously grounded phrase-box pairs do not affect the decisions for the upcoming phrases in a sentence. The model *NI* lacks the full image context where the only visual information to the framework is the *box stack*. The model *SBv* (simple box vector) lacks the bi-LSTM network for the boxes, and direclty uses the encoded box features coming from the triple fully connected layers in Figure 5.2. In this way, the decision for a phrase-box pair is made independently of the other box candidates. The model *SPv* (simple phrase vector) lacks the LSTM network for the *phrase stack* and directly uses the encoded phrase features coming from the triple fully connected layers in Figure 5.2. In this design, the framework is not aware of the upcoming phrases so that the decision for a phrase-box pair is made without the linguistic relations. Similarly, *SPvBv* lacks the bi-LSTM and LSTM networks for the box and phrase stacks, respectively. Moreover, *SPvBvNH* lacks the history module as an addition. Moreover, we created a baseline that performs phrase grounding in a non-sequential way by picking the most similar bounding box in the joint embedding space. To encode the phrases and

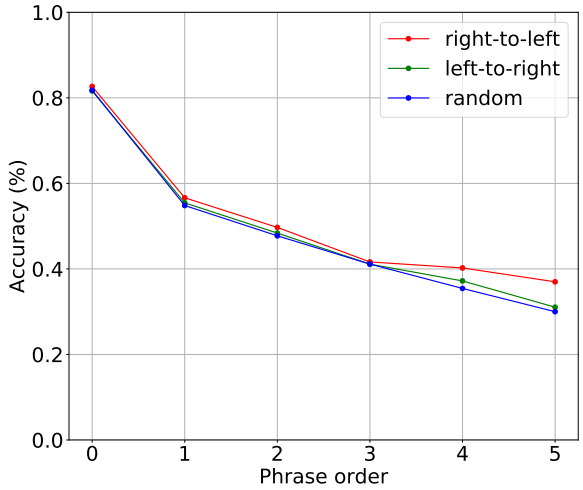


Figure 5.3: *Grounding accuracy versus the ordering of the grounded phrase among the noun phrases of the sentence.*

boxes, we used the same phrase-visual encoders that were pre-trained in Section 5.1.3. For each image-sentence input, we created a similarity matrix for all possible phrase-box pairs using the similarity function 4.2. Using this matrix, the phrases were grounded to the most similar box and boxes for the models *MSB* and *MSBs*, respectively.

Table 4.6 shows the performance of the six ablated models and two baselines on the Flickr30K Entities dataset. All these models perform substantially worse than the complete model of SeqGROUND. This confirms our intuition that knowing the global context for both visual and textual data, in addition to history and future, plays an important role in phrase grounding. We conclude that each stack contributes to our full model's superior performance.

Phrase Ordering. We consider several ways of ordering the phrases of a sentence.

1. Left-to-Right: The network grounds the phrases in lexical order, starting from the first phrase of the sentence.
2. Right-to-Left: The network grounds the phrases in reverse lexical order, starting from the last phrase.
3. Random: We randomly order the phrases for the *phrase stack*, and keep the ordering fixed for all of the training.

At test time, the phrases are ordered in the same order as the corresponding design's training time. The grounding accuracy with respect to the phrase's order among the noun phrases of the sentence is shown in Figure 5.3 for different ordering options. Red, green, and blue plots show the performance

Method	Accuracy
MCB [Fukui et al., 2016]	48.69
SMPL [Wang et al., 2016b]	42.08
NonlinearSP [Wang et al., 2016a]	43.89
GroundeR [Rohrbach et al., 2016]	47.81
RtP [Plummer et al., 2015]	50.89
Similarity Network [Wang et al., 2018]	51.05
IGOP [Yeh et al., 2017]	53.97
SPC+PPC [Plummer et al., 2017]	55.49
CITE [Plummer et al., 2018]	59.27
SeqGROUND	**61.60**

Table 5.2: *Phrase grounding accuracy (in percentage) of the state-of-the-art methods on the Flickr30k Entities dataset.*

when the phrases to the LSTM cell are ordered left-to-right (lexical order), right-to-left (reverse lexical order), and randomly, respectively. For all ordering options, the accuracy for the first phrase is significantly higher than the others. This is due to the fact that the first phrases usually belong to the category of *people* or *animals* which have significantly more samples in the dataset. Moreover, the candidate boxes from RPN are more accurate in proposing boxes for these categories which provides easier detection. The grounding accuracy drops towards the last phrases, which usually belong to the categories that have less samples in the dataset. Ordering the phrases right-to-left boosts the performance slightly for the last phrases of the sentence, since they are the first ones to be grounded. In this way, these hard-to-ground phrases are not a subject of a possible error cumulation in the *history stack*.

Unguided Testing. SeqGROUND does not necessarily need to be given phrases to ground. Due to its sequential nature, it scans through all the phrases in the sentences, selected phrases or not, and makes decisions which of those to ground and where (see Fig. 5.4). This is a more complex scenario than addressed by prior works, which only focus on phrases that implicitly have groundings.

5.2.4 Quantitative Results

We report the performance of SeqGROUND on the Flickr30K Entities dataset, and compare it with the state-of-the-art methods in Table 5.2. SeqGROUND is the top ranked method in the list, improving the overall

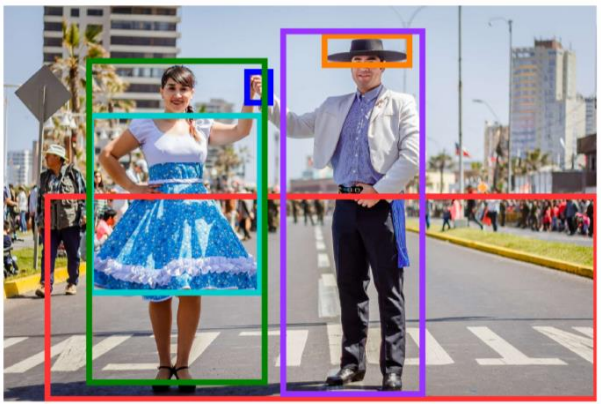
Method	people	clothing	body parts	animals	vehicles	instruments	scene	other
SMPL	57.89	34.61	15.87	55.98	52.25	23.46	34.22	26.23
GroundeR	61.00	38.12	10.33	62.55	68.75	36.42	58.18	29.08
RtP	64.73	46.88	17.21	65.83	68.72	37.65	51.39	31.77
IGOP	68.71	56.83	19.50	70.07	73.72	39.50	60.38	32.45
SPC+PPC	71.69	50.95	25.24	76.23	66.50	35.80	51.51	35.98
CITE	73.20	52.34	30.59	76.25	75.75	48.15	55.64	42.83
SeqGROUND	76.02	56.94	26.18	75.56	66.00	39.36	68.69	40.60

Table 5.3: *Comparison of phrase grounding accuracy (in percentage) over coarse categories on Flickr30K dataset.*

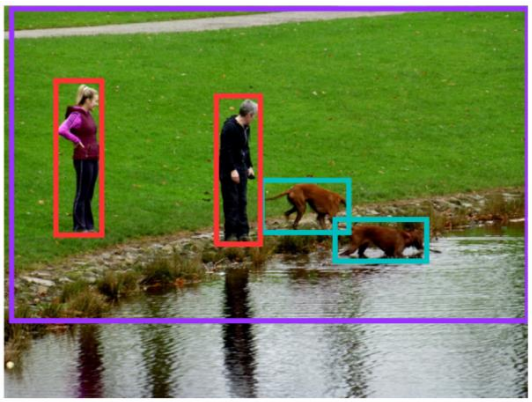
grounding accuracy by 2.33% to 12.91% by performing phrase grounding as a sequential and contextual process, compared to the prior work. For a fair comparison, all these methods use a fixed RPN to obtain the candidate boxes and represent them in features that are not tuned on the Flickr30K Entities dataset. We believe that using an additional conditional embedding unit as in [Plummer et al., 2018], and the integration of a proposal generation network with a spatial regression that is tuned on Flickr30K Entities as in [Chen et al., 2017] should improve the overall result even more. Performance on this task can be further improved by using Flickr30K-tuned features to represent the image regions, with the best result of 61.89% achieved by CITE [Plummer et al., 2018]. Futhermore, the use of an integrated proposal generation network to learn regression over Flickr30K Entities improves the result up to 65.14% as achieved by [Chen et al., 2017]. Table 5.3 shows the phrase grounding performance with respect to the coarse categories in Flickr30K Entitites dataset. Competing results are directly taken from the respective papers, if applicable.

5.2.5 Qualitative Results

We show some qualitative results in Figure 5.4 to highlight the capabilities of our method in challenging scenarios. The colored bounding boxes show the predicted grounding of the phrases in the same color. In (a) and (e), we see a succesful grounding of long sequence of phrases, note the correct grounding of *hands* in (a) despite other *hands* candidates. In (h), the phrase *glasses* is correctly grounded to a single correct box instead of selecting all the glasses, including the glasses of the partially occluded person in the middle even though it was one of the proposed boxes. Similarly in (d), SeqGROUND could distinguish which boxes to ground the phrases *a girl* and *a woman*, suppressing the other candidates despite their similar context. We believe

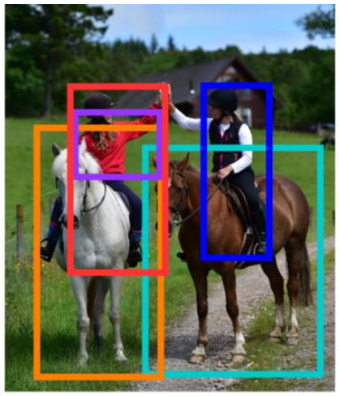


(a) A young lady in blue skirt and a man with a black hat are holding hands in the middle of a road.

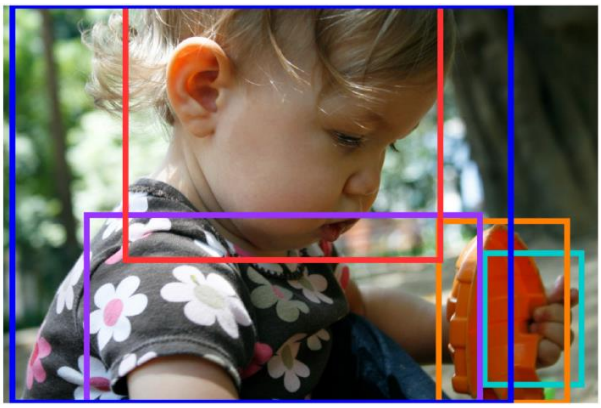


(b) Two people are standing near a lake looking at two brown dogs.

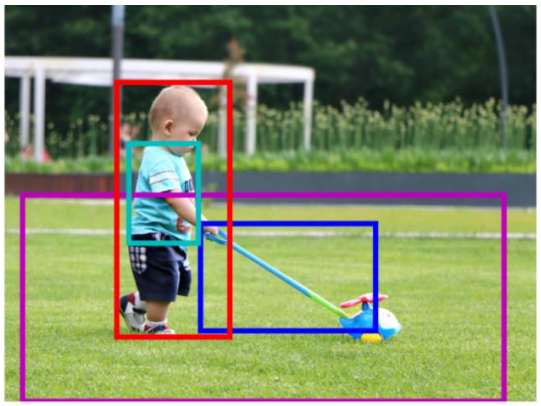
(c) Three people are dancing where the person in the middle wears a wedding gown.



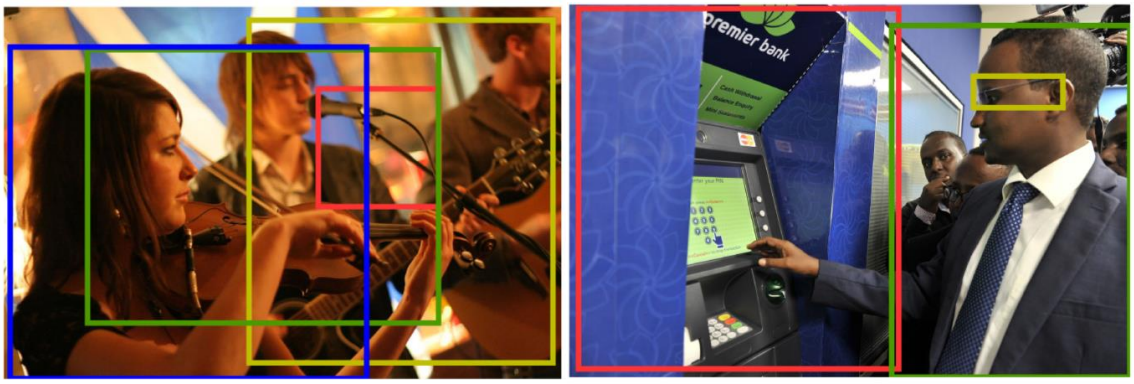
(d) A girl with a red shirt on a white horse and a woman on a dark horse are clapping their hands.



(e) A baby with blond hair in flower patterned shirt holding an orange toy in her hand.



(f) A toddler in a blue shirt is steering his toy on a grass field.



(g) A young woman is playing a violin while a young man is singing to a microphone.

(h) A man with glasses is working on an ATM machine.

Figure 5.4: *Sample phrase grounding results obtained by SeqGROUND.*

this is possibly due to SeqGROUND's ability to perform in a sequential way where it consders the global image and text context. As an intuitive example, the performed grounding starts by matching *a dark horse* to the correct box. Encoding this grounded pair and the overall contextual information, it grounds *a woman* to the correct box, which is just above *a dark horse*, instead of getting confused by the box that has *A girl*. At the decision time for *a woman*, the *phrase stack* encodes the future information, which is *a girl* should have a *red shirt* and should be on *a white horse*. Taking account of this information likely has led SeqGROUND to eliminate the box for *a girl* at the decision time for *a woman*. In (b), phrases are correctly grounded to multiple boxes, instead of one large single box for *two people* which would contain mostly *grass*. Likewise, (c) shows an example where a single box is used to ground multiple phrases, *three people* and *the person* which are positioned far apart. Phrase grounding with many-to-many matching is one of the distinguishing properties of SeqGROUND, which is partially or completely missing in most of the competing methods. All these images, and more in the supplementary material, show state-of-the-art performance of SeqGROUND due to its contextual and sequential nature.

5.2.6 Further Results

More examples are shown in Figures 5.5-5.12.

[1]Due to copyright issues of the images in the Flickr30K Entities dataset, we are not allowed to show images from it. Instead, we created similar content with images that have Creative Commons license.

(a) A Japanese woman poses in a ceremonial clothing with an elaborate headpiece.

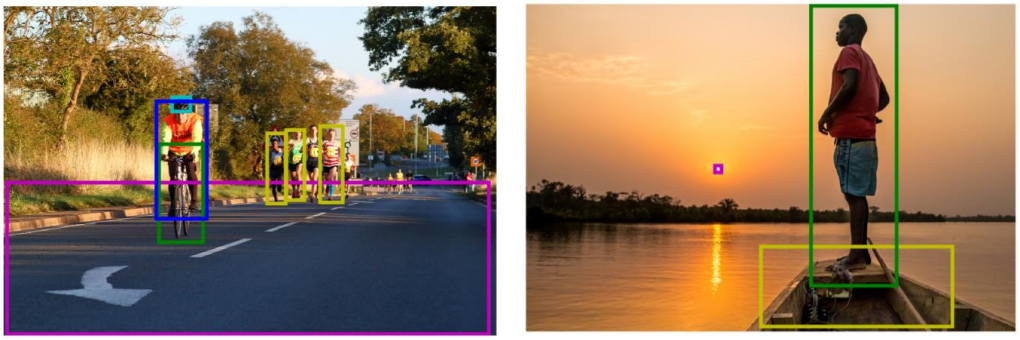
(b) Two men are dancing and holding hands with a bride that has an elegant wedding dress.

Figure 5.5: *Examples of succesful results.*

(a) A white dog is running over the water.

(b) A boy with a dark shirt is running ahead of two men on a paved road.

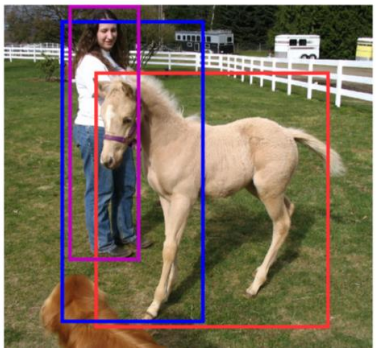
Figure 5.6: *Example results. (a) Succesful grounding. (b) The grounding of two men is partially missing the man at the very back.*



(a) A man with a helmet is riding a bike in front of a group of running men on the road.

(b) A man is standing on a boat as the sun sets.

Figure 5.7: *Examples of succesful results.*

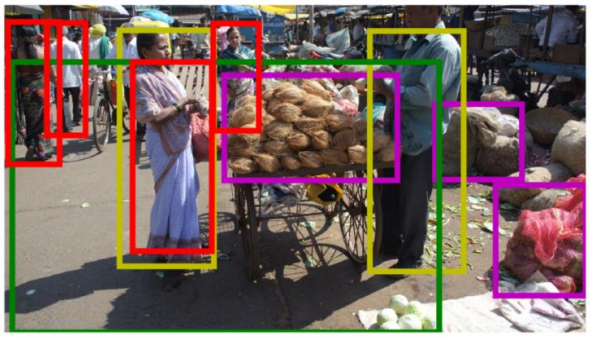


(a) A woman is looking at a young horse that looks at a dog.

(b) A man with a helmet is riding a motorbike in front of a yellow car.

Figure 5.8: *Failure cases. (a) a dog, which is partially visible, is grounded wrongly. (b) a yellow car is grounded to a blue car.*

(a) A man fishes on a calm sea under a nice sky.

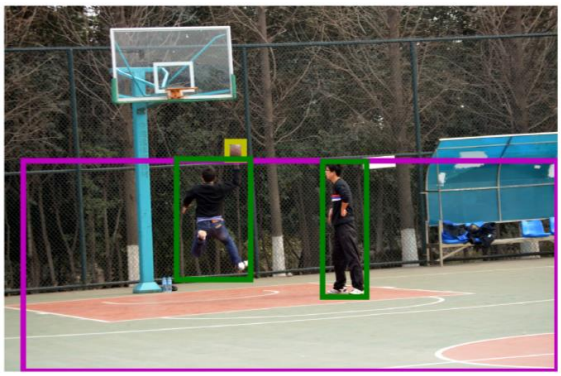


(b) Lots of people on the streets and a vendor selling her goodies.

Figure 5.9: *Failure cases (a) a calm sea is grounded to a much larger area. (b) a vendor is grounded to two people, which are challenging to distinguish.*

5.3 Summary

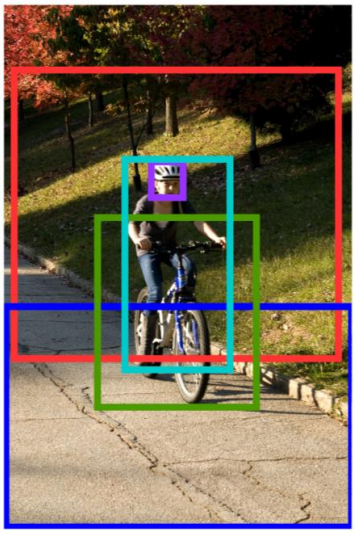
In this chapter, we proposed an end-to-end trainable Sequential Grounding Network (SeqGROUND) that formulates grounding of multiple phrases as a sequential and contextual process. SeqGROUND encodes region proposals, and all phrases into two stacks of LSTM cells along with the partially grounded phrase-region pairs to perform the grounding decision for the next phrase. Results on the Flickr30K Entities benchmark dataset and ablations studies show significant improvements of this model over more traditional grounding approaches.



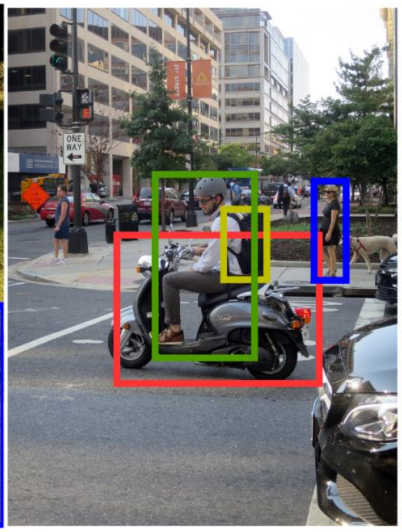
(a) Two boys are playing basketball in an outdoor court.

(b) A man with a hat in dark clothes on his skateboard is performing on an obstacle in front of a tree as an old man is watching him.

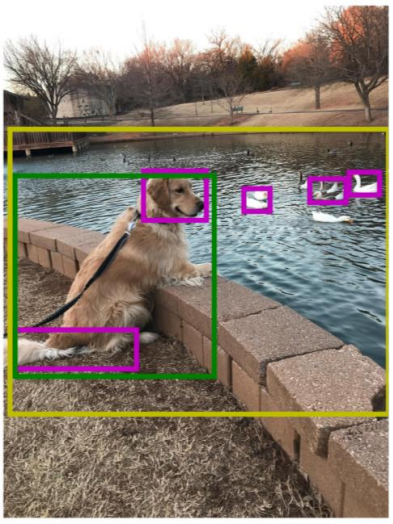
Figure 5.10: *Example results. (a) Succesful grounding. (b) a tree, which is significantly ocluded, is not grounded.*



(a) A girl with a helmet is riding a bike on a paved road near a grass field.

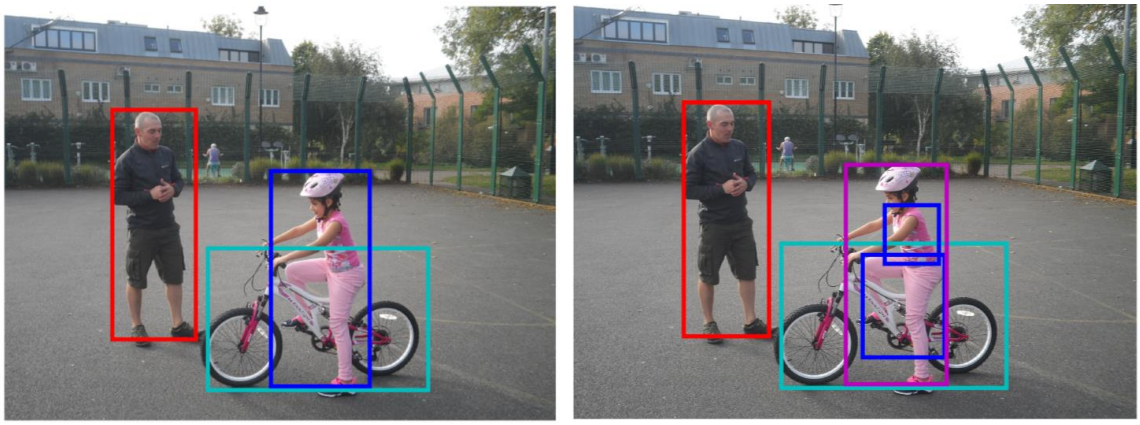


(b) A man with a backpack is on a motorbike and a woman with a dog is walking.



(c) A cute dog is standing near a river and looking around, and there is a group of swans.

Figure 5.11: *Example results. (a) Succesful grounding. (b) a dog is missed. (c) Some parts of a cute dog are assigned to a group of swans.*



(a) A girl in *black clothing* is riding a bike while her father is watching her.

(b) A girl in pink clothing is riding a bike while her father is watching her.

Figure 5.12: *Example results which are showing the efficacy of SeqGROUND. (a) Inaccurate phrase a black clothing is succesfully ignored by SeqGROUND. (b) Succesful grounding for an accurate description.*

CHAPTER 6

Conclusion

In this thesis, we presented novel methods for contextual alignment of visual and textual data, which is a significant step towards joint understanding of multi-modal content with narrative content. Namely, we developed a neural approach to align multi-modal data taking account of the temporal relations and dependencies within the data sequences, as well as a label-based approach to detect finer semantic changes in visual data. Furthermore, we extended our neural architecture in order to have a spatial contextualized representation of visual elements to propose a sequential approach for the phrase grounding task.

In the following, we will review the principle contributions of the thesis and discuss the limitations of the presented work as well as possible directions for future work.

6.1 Review of Principle Contributions

In this thesis, we have shown how contextualized representations can be used for multi-modal data alignment, and how this task can be formulated as a sequential decision classification problem. By building on recently developed neural architectures, we were able to take more global context into account while making alignment decisions, compared to the traditional methods.

In Chapter 3, we have presented a label-based approach to temporally align the video frames with the descriptive sentences using both visual and textual context. Our approach directly works on the raw video without the

need of shot segmentation or threading as a pre-processing step. Our method performs the shot segmentation in an integrated and iterative way, which allows the detection of semantic changes within continuous camera shots.

In Chapter 4, we have shown an end-to-end neural architecture where the alignment actions are implemented as moving data between stacks of LSTM blocks. This novel architecture formulates the alignment problem as a sequence of decision classification where the decisions are performed by taking account of conducive contextual information that is scattered over the visual and textual data sequences. Besides allowing one-to-many alignment of sequences, this flexible architecture supports non-monotonicity and alignment of multiple sequences with extensions.

In Chapter 5, we have presented an expansion on recent temporal alignment networks, and developed a sequential *spatial* phrase grounding network, SeqGROUND, which allows many-to-many grounding decisions. We have proposed the notion of contextual and sequential phrase grounding, where earlier decisions can inform the latter, and formulated this process with an end-to-end learnable neural architecture.

To conclude, we believe that, traditional alignment approaches, which perform on pre-processed data in two stages by defining a similarity metric and applying an optimal alignment technique based on dynamic programming, are disadvantaged by the separation of these stages. Instead, neural architectures that learn a metric directly helping to optimize alignment are beneficial. Furthermore, these architectures are capable of using contextual information and dependencies that are scattered far apart and beyond limited local context. We think using neural methods for multi-modal data alignment bear potential for more interesting research and applications, and hope that our work provides an important step in such a direction.

6.2 Future Work

We have presented some advances on how to align visual and textual data with narrative content, however the methods are still far from perfectly solving the challenges in complex scenarios. While we have discussed the specific technical issues of each method in the corresponding chapters, we will discuss the limitations of the presented work on a broader view as well as possible directions for future work.

Currently the main limitation of video-text alignment is that the alignment granularity remains very coarse for an acceptable accuracy for long length

text and videos. As the length of the sequences increase, dissimilarities and non-monotonicity escalates drastically. For example, consider aligning a book to its movie adaptation. Most of the time, the original story from the book is changed largely in the movie: some events and dialogues are missed or added, the temporal order of the events are changed, the scenes and even the characters are portrayed with different physical proporties. As the length increases, the stories divert even more, where the local information in the sequence elements of fine granularity is not enough by itself for an accurate alignment decision since the global context is scattered far apart. Even with coarse granularity levels, such as chapter to scene alignment, the accuracy numbers stay low. Possible directions of future work include having a more accurate fine level alignment could be learning and performing the alignment in a pyramid representation, where strong anchor points from the coarser level alignment are used for a finer alignment at every iteration. Designing a neural achitecture to learn such a behaviour in general would be interesting.

For the image-text alignment, a significant limitation is the use of pre-trained region proposal networks that pose an upper bound on the detection performance. One direction to overcome this problem is designing a neural network that introduces an integrated region proposal generation network that learns regression on the objects.

Another interesting direction to explore would be combining video-text and image-sentence alignments to improve the overall alignment for the purpose of meta-data extraction. Once a unit of video-sentence pair is aligned, the phrases of the aligned sentence could be localized in the frames of the video. The localized entities could be propagated, and the next video-sentence alignment decision could be performed using this extra information obtained by the propagation. The overall procedure could be designed as a chain of alternation between alignment and propagation through time. For this purpose, optical flow and temporal propagation methods would be required to integrate.

An issue we have not addressed in thesis is the generation of a description sentence or an expression for a selected video part or an image region. Considering the alignment problem as a comprehension task, we can jointly design a model for both comprehension and generation tasks. Being able to generate text will allow us to annotate the elements of the text sequence that do not have correspondance in the visual data. In this way, the whole video can be annotated with the aligned or generated meta-data.

References

[Anderson et al., 2017] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and VQA. *arXiv preprint arXiv:1707.07998*, 2017.

[Antol et al., 2015] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *ICCV*, pages 2425–2433, 2015.

[Apostolidis and Mezaris, 2014] Evlampios Apostolidis and Vasileios Mezaris. Fast shot segmentation combining global and local visual descriptors. In *ICASSP*, 2014.

[Arora et al., 2016] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. 2016.

[aud, 2009] Description key for educational media. *The Described and Captioned Media Program*, 2009.

[Bahdanau et al., 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Barzilay and Lee, 2003] Regina Barzilay and Lillian Lee. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 16–23. Association for Computational Linguistics, 2003.

[Bengio et al., 2003] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[Berndt and Clifford, 1994] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, 1994.

[Bojanowski et al., 2015] Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Weakly-supervised alignment of video with text. In *ICCV*, pages 4462–4470, 2015.

[Bojanowski et al., 2017] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[Bowman et al., 2015] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[Caspi and Irani, 2000] Yaron Caspi and Michal Irani. A step towards sequence-to-sequence alignment. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 682–689. IEEE, 2000.

[Cer et al., 2018] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[Chakravarthi, 1992] Prakash Chakravarthi. The history of communications-from cave drawings to mail messages. *IEEE Aerospace and Electronic Systems Magazine*, 7(4):30–35, 1992.

[Chen and Gupta, 2017] Xinlei Chen and Abhinav Gupta. Spatial memory for context reasoning in object detection. In *ICCV*, 2017.

[Chen and Manning, 2014] Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750, 2014.

[Chen et al., 2017] Kan Chen, Rama Kovvuri, and Ram Nevatia. Query-guided regression network with context policy for phrase grounding. In *ICCV*, 2017.

[Chen et al., 2018a] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[Chen et al., 2018b] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In *CVPR*, 2018.

[Cho et al., 2014] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[Chung et al., 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[Conneau et al., 2017] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[Cour et al., 2008] Timothee Cour, Chris Jordan, Eleni Miltsakaki, and Ben Taskar. Movie/script: Alignment and parsing of video and text transcription. In *European Conference on Computer Vision*, pages 158–171. Springer, 2008.

[Dai et al., 2016] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016.

[Dijkstra, 1959] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1959.

[Diringer, 2013] David Diringer. *The book before printing: ancient, medieval and oriental*. Courier Corporation, 2013.

[Doğan et al., 2015] Pelin Doğan, Tunç Ozan Aydın, Nikolce Stefanoski, and Aljoscha Smolic. Key-frame based spatiotemporal scribble propagation. In *Proceedings of the Eurographics Workshop on Intelligent Cinematography and Editing*, pages 13–20. Eurographics Association, 2015.

[Dogan et al., 2016] Pelin Dogan, Markus Gross, and Jean-Charles Bazin. Label-based automatic alignment of video with narrative sentences. In *European Conference on Computer Vision*, pages 605–620. Springer, 2016.

[Dogan et al., 2018] Pelin Dogan, Boyang Li, Leonid Sigal, and Markus Gross. A neural multi-sequence alignment technique (neumatch). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8749–8758, 2018.

[Dogan et al., 2019] Pelin Dogan, Leonid Sigal, and Markus Gross. Neural sequential phrase grounding (seqground). *arXiv preprint arXiv:1903.07669*, 2019.

[Donahue et al., 2014] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014.

[Donahue et al., 2015] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[Drew et al., 1999] Mark S Drew, Jie Wei, and Ze-Nian Li. Illumination–invariant image retrieval and video segmentation. *Pattern Recognition*, 1999.

[Dyer et al., 2015] Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075*, 2015.

[Everingham et al., 2006] Mark Everingham, Josef Sivic, and Andrew Zisserman. Hello! my name is... buffy–automatic naming of characters in tv video. 2006.

[Farhadi et al., 2010] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*. 2010.

[Felzenszwalb et al., 2010] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[Fukui et al., 2016] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.

[Girshick et al., 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, June 2014.

[Girshick, 2015] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[Goodfellow et al., 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[Hampapur et al., 1995] Arun Hampapur, Ramesh Jain, and Terry E Weymouth. Production model based digital video segmentation. *Multimedia Tools and Applications*, 1995.

[Hazen, 2014] Kirk Hazen. *An Introduction to Language*. Wiley Blackwell, 2014.

[He et al., 2014] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European conference on computer vision*, pages 346–361. Springer, 2014.

[He et al., 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[He et al., 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[Hodosh et al., 2013] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[Honnibal and Johnson, 2015] Matthew Honnibal and Mark Johnson. An improved non-monotonic transition system for dependency parsing. In *EMNLP*, pages 1373–1378, 2015.

[Hu et al., 2016] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *CVPR*, pages 4555–4564, 2016.

[Huang et al., 2016] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, pages 646–661. Springer, 2016.

[Jaccard, 1912] Paul Jaccard. The distribution of the flora in the alpine zone. *New phytologist*, 11(2):37–50, 1912.

[Jia et al., 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, 2014.

[Karpathy and Fei-Fei, 2015] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pages 3128–3137, 2015.

[Karpathy et al., 2014] Andrej Karpathy, Armand Joulin, and Li F Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*, pages 1889–1897, 2014.

[Kazemzadeh et al., 2014] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

[Kiros et al., 2014] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *Transactions of the Association for Computational Linguistics*, 2014.

[Kiros et al., 2015] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *NIPS*, pages 3294–3302, 2015.

[Kong et al., 2014] Chen Kong, Dahua Lin, Mohit Bansal, Raquel Urtasun, and Sanja Fidler. What are you talking about? text-to-image coreference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3558–3565, 2014.

[Krishna et al., 2017] 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. In *IJCV*, 2017.

[Krizhevsky et al., 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[Lancelle et al., 2019] Marcel Lancelle, Pelin Dogan, and Markus Gross. Controlling Motion Blur in Synthetic Long Time Exposures. *Computer Graphics Forum (Proc. Eurographics)*, 38(2), 2019.

[Lankinen and Kämäräinen, 2013] Jukka Lankinen and Joni-Kristian Kämäräinen. Video shot boundary detection using visual bag-of-words. In *VISAPP*, 2013.

[LeCun et al., 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Lee and Ip, 1995] John Chung-Mong Lee and Dixon Man-Ching Ip. A robust approach for camera break detection in color video sequence. *MVA*, 1995.

[Lin et al., 2014] Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *CVPR*, pages 2657–2664, 2014.

[Logeswaran and Lee, 2018] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*, 2018.

[Long et al., 2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[Löytynoja and Goldman, 2005] Ari Löytynoja and Nick Goldman. An algorithm for progressive multiple alignment of sequences with insertions. *Proceedings of the National academy of sciences of the United States of America*, 102(30):10557–10562, 2005.

[Luong et al., 2015] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[Malmaud et al., 2015] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. *arXiv preprint arXiv:1503.01558*, 2015.

[Mao et al., 2014] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint*, 2014.

[Mao et al., 2016] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016.

[Mikolov et al., 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Mikolov et al., 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[Nagaraja et al., 2016] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *European Conference on Computer Vision*, pages 792–807. Springer, 2016.

[Nagasaka and Tanaka, 1992] Akio Nagasaka and Yuzuru Tanaka. Automatic video indexing and full-video search for object appearances. 1992.

[Noh et al., 2015] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[Oquab et al., 2014] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, June 2014.

[Pan et al., 2016] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, pages 4594–4602, 2016.

[Pennington et al., 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Peters et al., 2018] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[Plummer et al., 2015] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, pages 2641–2649, 2015.

[Plummer et al., 2017] Bryan A Plummer, Arun Mallya, Christopher M Cervantes, Julia Hockenmaier, and Svetlana Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. In *ICCV*, 2017.

[Plummer et al., 2018] Bryan A Plummer, Paige Kordas, M Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. Conditional image-text embedding networks. In *ECCV*, 2018.

[Prokić et al., 2009] Jelena Prokić, Martijn Wieling, and John Nerbonne. Multiple sequence alignments in linguistics. In *EACL Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education*, pages 18–25, 2009.

[Qu et al., 2009] Zhiyi Qu, Ying Liu, Liping Ren, Yong Chen, and Ruidong Zheng. A method of shot detection based on color and edge features. In *IEEE Symposium on Web Society (SWS)*, 2009.

[Ranzato et al., 2014] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[Redmon et al., 2016] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, June 2016.

[Ren et al., 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[Rohrbach et al., 2015] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3202–3212, 2015.

[Rohrbach et al., 2016] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV*, pages 817–834. Springer, 2016.

[Rücklé et al., 2018] Andreas Rücklé, Steffen Eger, Maxime Peyrard, and Iryna Gurevych. Concatenated power mean embeddings as universal cross-lingual sentence representations. *arXiv*, 2018.

[Sadeghi et al., 2015] Fereshteh Sadeghi, Santosh K Kumar Divvala, and Ali Farhadi. Viske: Visual knowledge extraction and question answering by visual verification of relation phrases. In *CVPR*, pages 1456–1464, 2015.

[Sakoe and Chiba, 1978] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1978.

[Sankar et al., 2009] Pramod Sankar, CV Jawahar, and Andrew Zisserman. Subtitle-free movie to script alignment. In *BMVC*, 2009.

[Sheinfeld et al., 2016] Shay Sheinfeld, Yotam Gingold, and Ariel Shamir. Video summarization using causality graphs. In *HCOMP Workshop on Human Computation for Image and Video Analysis*, 2016.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Subramanian et al., 2018] Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*, 2018.

[Sung and Poggio, 1998] K-K Sung and Tomaso Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 20(1):39–51, 1998.

[Sutskever et al., 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[Sutton and Barto, 2017] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second (complete draft) edition, 2017.

[Szegedy et al., 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[Szegedy et al., 2016] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[Szegedy et al., 2017] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[Tapaswi et al., 2014] Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Story-based video retrieval in TV series using plot synopses. In *Proceedings of International Conference on Multimedia Retrieval*, 2014.

[Tapaswi et al., 2015] Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. Book2Movie: Aligning video scenes with book chapters. In *CVPR*, 2015.

[Torabi et al., 2016] Atousa Torabi, Niket Tandon, and Leonid Sigal. Learning language-visual embedding for movie understanding with natural-language. *arXiv preprint arXiv:1609.08124*, 2016.

[Tran et al., 2015] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[Van Rijsbergen, 1977] Cornelis Joost Van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of documentation*, 33(2):106–119, 1977.

[Vendrov et al., 2015] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*, 2015.

[Venugopalan et al., 2014] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

[Venugopalan et al., 2015] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to

sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

[Vinyals et al., 2015a] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *NIPS*, pages 2692–2700, 2015.

[Vinyals et al., 2015b] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164, 2015.

[Wang et al., 2016a] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, pages 5005–5013, 2016.

[Wang et al., 2016b] Mingzhe Wang, Mahmoud Azab, Noriyuki Kojima, Rada Mihalcea, and Jia Deng. Structured matching for phrase localization. In *ECCV*, pages 696–711. Springer, 2016.

[Wang et al., 2018] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[Xiao et al., 2017] F. Xiao, L. Sigal, and Y. J. Lee. Weakly-supervised visual grounding of phrases with linguistic structures. In *CVPR*, 2017.

[Xie et al., 2017] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5987–5995. IEEE, 2017.

[Xu and Saenko, 2016] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, pages 451–466, 2016.

[Xu et al., 2015a] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015.

[Xu et al., 2015b] Ran Xu, Caiming Xiong, Wei Chen, and Jason J Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, volume 5, page 6, 2015.

[Xu et al., 2017] B. Xu, Y. Fu, Y. G. Jiang, B. Li, and L. Sigal. Heterogeneous knowledge transfer in video emotion recognition, attribution and summarization. *IEEE Transactions on Affective Computing*, 2017.

[Yang et al., 2018] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *ECCV*, 2018.

[Yao et al., 2015] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515, 2015.

[Yeh et al., 2017] Raymond Yeh, Jinjun Xiong, Wen-Mei Hwu, Minh Do, and Alexander Schwing. Interpretable and globally optimal prediction for textual grounding using image concepts. In *Advances in Neural Information Processing Systems*, pages 1912–1922, 2017.

[Yosinski et al., 2014] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NIPS*, pages 3320–3328, Cambridge, MA, USA, 2014. MIT Press.

[Yu et al., 2016a] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4584–4593, 2016.

[Yu et al., 2016b] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016.

[Yu et al., 2018] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[Zhang et al., 2017] 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.

[Zhou and De la Torre, 2016] Feng Zhou and Fernando De la Torre. Generalized canonical time warping. *PAMI*, 38(2):279–294, 2016.

[Zhou et al., 2016] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[Zhu et al., 2015] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *CVPR*, 2015.

[Zoph et al., 2018] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.