Karlsruher Institut für Technologie

Story Understanding through Semantic Analysis and Automatic Alignment of Text and Video

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)
genehmigte

Dissertation
von

Makarand Murari Tapaswi

aus Goa, Indien

Tag der mündlichen Prüfung:	16. Juni 2016
Hauptreferent:	Prof. Dr.-Ing. Rainer Stiefelhagen Karlsruher Institut für Technologie
Korreferent:	Dr. Cordelia Schmid INRIA, Grenoble

Abstract

Stories are the pinnacle of human creativity, and yet a ubiquitous phenomenon. An important element of human communication consists of telling and listening, reading, and nowadays watching stories enacted on film. Among different means of story-telling, videos (TV series and movies) are a very powerful medium as they have the potential to engage multiple human senses.

Artificial Intelligence (AI) has made large strides in the last decade, through which several advances have been achieved in both language and vision. However, these fields have been primarily studied separately, and only in the last few years do we see joint analysis rising to prominence. We argue that the ability to model, analyze, understand, and create new stories is a stepping stone for strong AI – a machine that could perform any intellectual task that a human can. Towards this grander goal of story understanding, we seek to build upon a wave of research in the joint study of vision and language.

TV series and movies are perfect candidates for such a study, as they are videos produced for the specific purpose of story-telling. In this thesis, we define machine understanding of stories as the ability to perform human-like tasks upon those stories, such as indexing and searching for story events in large collections, summarizing the stories, and answering questions about them. We address the problem of story understanding from three vantage points. First, we introduce the use of novel sources of natural language text that allow to better learn the content of the videos. Next, we propose a visualization technique to obtain a big picture overview of the story conveyed in a video. Finally, we provide a means to examine machine understanding of stories by using question-answering as a surrogate task.

Subtitles and transcripts have been an excellent source of low-level information for video understanding, however, they are inadequate to understand the story plot. We introduce the use of two diverse forms of natural language text that focus on the story: *plot synopses* and *books*. Plot synopses are concise descriptions of the story in the episodes or movies and are obtained easily through crowd sourcing. On the contrary, books, from which the videos are adapted, are large texts that describe the events (characters, scenes, and interactions) in rich detail. Unlike transcripts, the potential of these text sources needs to be unlocked by first aligning the text units with the video. We propose similarity metrics to bridge the gap between the text and video modalities. Using them, we align plot synopsis sentences with individual video shots, and book chapters with video scenes. To this end, we develop several alignment models that attempt to maximize joint similarity while respecting story progression constraints. We test these approaches on two sets of videos for both plots and books and obtain promising alignment performance. The alignment gives rise to applications such as describing video clips using plot sentences or book paragraphs, story-based video retrieval using plots as intermediaries, and even the ability to predict whether a scene from the video adaptation was present in the original book.

Our second approach towards improving story understanding is through visualization. We automatically generate StoryGraphs – charts that depict character interactions in an episode and augment them with information about key events. The graph layout is treated as an optimization problem that trades off functionality with aesthetics. We conduct a user experiment and show that such graphs can aid humans in speeding up the search for story-events in a video.

Our third important contribution is in the field of assessing machine understanding. Here, we create a large scale question-answering (QA) data set based on movie stories. The data set not only covers simple visual aspects such as “Who”, “What”, and “Where”, but also requires long-range temporal reasoning to answer “Why” and “How” questions. A unique aspect of our QA data set is that answering can be performed using text sources (*e.g.* plots, subtitles) or video clips. The data set is made available as part of a benchmark challenge. Further, we analyze data set bias, explore the quality of our multiple-choice questions, and propose several techniques for answering.

In addition to the primary contributions, we also work on analyzing and creating better meta-data for the videos. In particular, we propose new techniques for scene boundary detection, and improve person identification in TV series.

Kurzzusammenfassung

Geschichten sind ein Höhepunkt menschlicher Kreativität und noch immer ein universelles Phänomen. Ein wesentlicher Teil der menschlichen Kommunikation besteht aus dem Lesen, Erzählen und Anhören von Geschichten. Heutzutage geschieht dies meist in einer modernen Form, als Fernsehserien und -filme. Von den verschiedenen Möglichkeiten eine Geschichte zu erzählen, sind diese ein sehr leistungsfähiges Medium, da sie mehrere menschliche Sinne beteiligen können.

Im letzten Jahrzehnt hat die Entwicklung der Künstlichen Intelligenz (KI) große Fortschritte gemacht, wodurch mehrere Entwicklungen in der Sprach- und Bildverarbeitung möglich wurden. Dennoch wurden diese Fachgebiete bisher hauptsächlich getrennt voneinander untersucht, erst in den letzten Jahren änderte sich dies. Wir betrachten die Fähigkeit, Geschichten zu analysieren, zu verstehen und neu zu erschaffen, als einen wichtigen Schritt auf dem Weg zur Entwicklung starker KI – das heißt einer Maschine, die jegliche intellektuelle Aufgabe wie ein Mensch lösen kann. Für die Analyse und das Verständnis von Geschichten wollen wir dabei auf den großen Fortschritten im Bereich der automatischen Sprach- und Bildanalyse aufbauen.

In dieser Dissertation definieren wir Maschinelles Verständnis von Geschichten als die Fähigkeit einer Maschine, wie ein Mensch mit Geschichten umzugehen, das heißt zum Beispiel, die Fähigkeit bestimmte Handlungen und Ereignisse zu erkennen und wieder finden zu können, sowie die Fähigkeit Geschichten zusammen zu fassen und Fragen über die Geschichte beantworten zu können. Als Anwendungsbeispiele betrachten wir dabei Fernsehserien und Spielfilme.

Wir betrachten das Problem des Verständnisses von Geschichten aus drei verschiedenen Blickwinkeln. Zuerst führen wir die Nutzung neuartiger Bezugsquellen von Video-

Beschreibungen in natürlicher Sprache ein, welche uns ein besseres automatisches Verständnis des Videoinhaltes ermöglichen. Dann schlagen wir eine Visualisierungstechnik vor, um einen Überblick der Geschichte zu bekommen. Abschließend stellen wir die automatische Beantwortung von Fragen zu Filmen als eine Methode vor, mit deren Hilfe das maschinelle Verständnis von Geschichten bewertet werden kann.

Texte, wie z.B. Untertitel und Filmskripte, sind eine exzellente Quelle an ergänzenden Informationen um Videos zu verstehen. Wir stellen zwei unterschiedliche natürlichsprachige Textquellen vor: Synopsen (plot synopses) und Bücher. Synopsen sind kurze Zusammenfassungen von Serien oder Filmen, sie sind für viele Filme und Serien erhältlich. Im Gegensatz dazu sind Bücher lange Texte, die Ereignisse (Charaktere, Szenen und Interaktionen) mit zahlreichen Details beschreiben. Um ihr Potenzial zur inhaltlichen Analyse der Videos zu entfalten, müssen die einzelnen Textabschnitte dieser Textquellen dabei zunächst den Szenen eines Videos zugeordnet werden. Wir erstellen Ähnlichkeitsfunktionen, um diese Lücke zwischen Text und Video schließen. Hiermit ordnen wir die Sätze der Synopsen den einzelnen Video-Einstellungen (shots), sowie einzelne Buchkapitel den Videoszenen, automatisch zu. Wir entwickeln hierzu verschiedene Alignment-Modelle, welche unsere Ähnlichkeitsfunktionen maximieren, dabei aber den Ablauf der Geschichte nicht außer Acht lassen. Wir testen diese Ansätze an zwei Datenquellen, sowohl für Synopsen als auch für Bücher und können dabei vielversprechende Ergebnisse für das Alignment erzielen. Dies ermöglicht eine Fülle von Anwendungen, wie die Beschreibung von Video-Clips durch Teile der Synopse oder dem Buch, das Auffinden von Ereignissen in Videos unter Nutzung der Synopse als Zwischenschritt und auch die Fähigkeit fest zu stellen, ob eine Szene einer Videoadaptierung im Buch vorhanden ist.

Unser zweiter Ansatz zur Verbesserung des Verständnisses einer Geschichte erfolgt durch Visualisierung. Wir generieren so gennante *StoryGraphs*, Diagramme, welche die Interaktionen zwischen Personen in einer Folge darstellen und ergänzen diese mit Informationen zu wichtigen Ereignissen. Die Anordnung der Grafik wird dabei als ein Optimierungsproblem betrachtet, welches funktionale und ästhetische Aspekte abwägt. Wir führen eine Benutzerstudie durch und zeigen, dass derartige Grafiken dem Menschen dabei helfen, gesuchte Ereignisse im Video schneller zu finden.

Unser dritter wichtiger Beitrag ist im Bereich der Bewertung von Maschinellem Verständnis von Geschichten. Zu diesem Zweck erstellen wir eine große Frage-Antwort (Question-Answering) Datenbank, basierend auf Filmgeschichten. Die Datenbank beinhaltet nicht nur einfache visuelle “Wer”, “Was” und “Wo” Aspekte, sondern sie erfordert

auch Schlussfolgerungen über lange Zeiträume hinweg, um Fragen über das “Warum” und “Wie” beantworten zu können. Ein besonderer Aspekt unserer Frage-Antwort Datenbank ist, dass die Beantwortung der Fragen mit Hilfe von Textquellen (z.B. Synopsen, Untertiteln) oder Video Clips erfolgen kann. Die Datenbank wird als Teil eines Benchmarks veröffentlicht. Wir analysieren weiterhin verschiedene Merkmale und Aspekte der Datenbank (data set bias), untersuchen die Qualität unserer Multiple-Choice Fragen und schlagen diverse Techniken zur automatischen Beantwortung der Fragen vor.

Zusätzlich zu unseren Hauptbeiträgen arbeiten wir auch an der Analyse und Erstellung von besseren Meta-Daten für Videos. Insbesondere schlagen wir neue Techniken zur Detektion von Szenenschnitten vor und verbessern die Identifikation von Personen in Fernsehserien.

Acknowledgments

The journey of a doctoral thesis is a story in itself with many players. I take this opportunity to thank them all for motivating and guiding me and making it a great experience.

Firstly, I thank Rainer Stiefelhagen for providing an open work environment where I was free to choose a topic and aim towards higher semantics. I also thank him for the numerous learning opportunities presented to me in creating lectures, writing proposals, presenting demos, and participating in other lab-wide activities. I also thank Cordelia Schmid for kindly agreeing to be a reviewer and improving the quality of this thesis.

Starting from my Master's thesis, I found a great guide, brainstorming partner, and friend in Martin Bäuml. Thank you so much for all your help and patience and nurturing me throughout this time. It has, in part, come to fruition through this thesis.

I thank all other members at our lab including Hazım, Boris, Ziad, Tobias, Manel, Daniel, Monica, Lukas, Arne, and Manuel for making these five years of my life a memorable time. Special thanks to Corinna for her untiring help with the administrative things and making me feel welcome. I was happy to work with Çağrı, Esam, and Monica on their theses and thank them for teaching me how to teach!

I thank Rainer again for encouraging me to go on internships. I was lucky to spend three summer months of 2013 at the Visual Geometry Group in Oxford, and thank Andrew Zisserman for providing this opportunity and taking a lot of time to advise me. I thank Omkar for the fun we had thinking about our face clustering problem together (in Marathi!), and Minh, Eric, Yusuf, and Relja for a nice stay. In the fall of 2015, I had another opportunity to visit Sanja Fidler and Raquel Urtasun's lab at the University of Toronto. This fantastic collaboration resulted in the MovieQA data set. I thank

Sanja and Raquel for the chance and guiding me through every step of the way, Yukun Zhu for his amazing help, and Antonio Torralba for guidance with the dataset. I found knowledgeable labmates in Elman and Shikhar and thank them for solving numerous Theano doubts, and Lluis, Kaustav, Namdar, Ivan, and Alex for making it a memorable stay. Finally, I am excited to visit UofT as a post-doctoral fellow.

Most importantly, I thank my parents and family for inculcating great values, believing in me, and fostering an open environment. Passion and dedication for one's work, while remaining calm and composed are key aspects I learned from them. Their support through this long journey was invaluable. I thank Divya for her love and support and look forward to the future!

श्रीकृष्णार्पणमस्तु

Contents

Chapter 1

Introduction

Understanding and discussing *stories* encompasses a phenomenally large time among us humans. We engage each other in telling or listening to stories; reading about them; or nowadays, watching them enacted on film (as movies and TV series). Thus, stories are one of the most important features of human engagement. Stories are ubiquitous and appear not only in fictional forms (*e.g.* films, comic strips), but also in our daily newspapers or even history books. Stories form an essential part of child development imparting moral values and inculcating creativity and imagination among children.

So common is story-telling that often the reason for indulging in this activity is not thought about. Fictional stories basically stem from our imagination, the process of perceiving images of things that are not actually in front of our eyes, and the ability to then create interactions between the players of the hypothesized world. The simple mention of "fire breathing dragon" triggers a mental chain reaction where we possibly imagine a dragon somewhere high in the mountains. We may even form a story around it with some villagers harassed by the dragon going to their queen and pleading with her to defend them (see Fig. 1.1). One of the surprising components of story-telling is that stories told all over the world (even prior to globalization) seem to have a remarkably similar patterns (Booker, 2005).

Among techniques for story-telling, *films* activate and engage multiple human senses and are thus the more interesting form of stories to analyze. As Artificial Intelligence (AI) develops and learns to deal with relatively simpler tasks of classifying image content, or representing language in vector space, the pinnacle would be to understand existing stories, learn from them, and develop the ability to craft new stories. With the growing

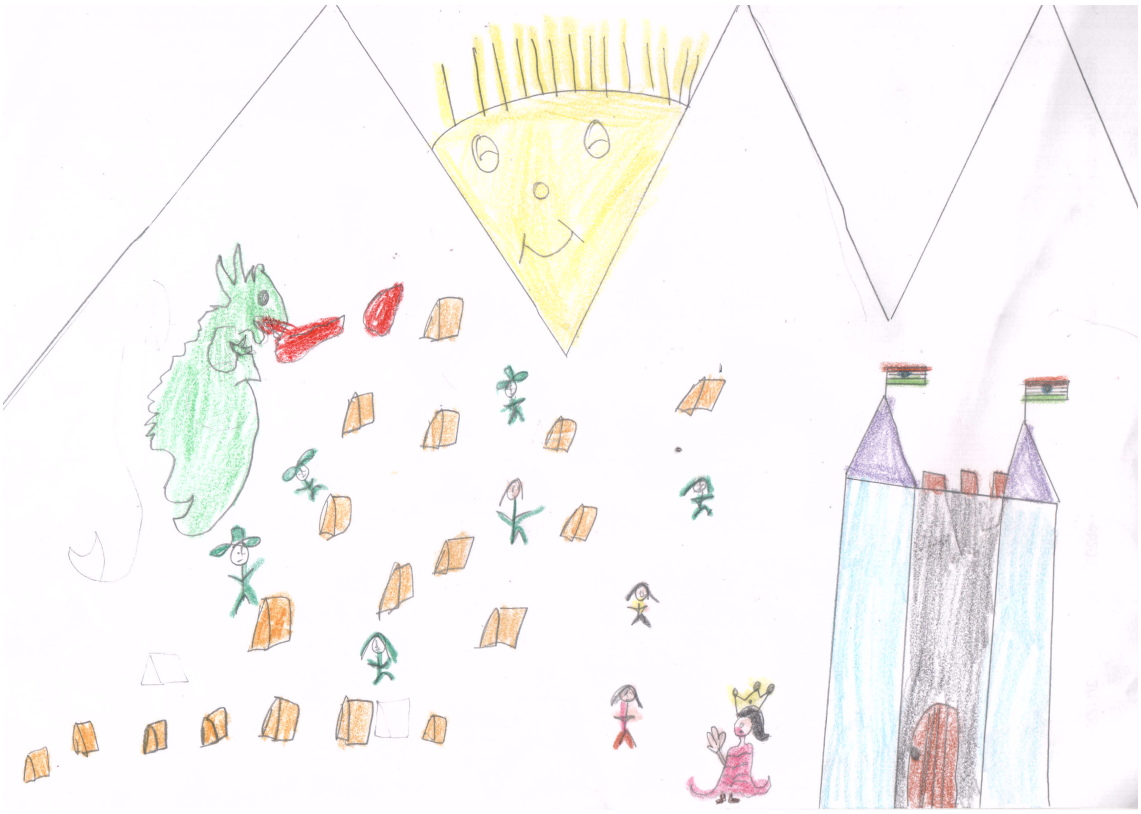


Figure 1.1: Stories are a time tested means to teach children about our world and are often associated with morals and creative thinking. On the contrary, current AI systems operate in restricted domains that do not entail a world view. Learning to understand stories is a step towards this broader direction. Here, we see a portrayal of the "fire breathing dragon" discussed in the introduction by a 6 year old. Being able to draw pictures from a textual description, or describing the content of images highlights the fascinating interplay between human visual and lingual processing centers, which are responsible for story comprehension. Image courtesy: Ananya Tapaswi.

popularity and emergence of interesting research through the joint study of vision and language, this thesis makes several contributions towards automatic story understanding.

AI and Vision. Over the past decade, artificial intelligence has made rapid strides in understanding the visual world as we see it (LeCun et al., 2015). Image-based recognition has seen a dramatic shift from hand-crafted features such as Scale-Invariant Feature Transform (SIFT) (Lowe, 2004) and Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2005) towards end-to-end trainable models for image representation and classification using deep Convolutional Neural Networks (CNNs) (Krizhevsky et al., 2012). On the arduous path of image understanding, we acknowledge several benchmark

data sets that have helped create a common platform to improve various tasks in Computer Vision (CV) such as scene recognition (*e.g.* SUN (Xiao et al., 2010), Places (Zhou et al., 2014)), object detection and recognition (*e.g.* Pascal VOC (Everingham et al., 2015), ImageNet (Russakovsky et al., 2015), Microsoft COCO (Lin et al., 2014b)), action recognition (*e.g.* Hollywood (Laptev et al., 2008), Youtube 1M Sports (Karpathy et al., 2014)), person identification or verification (*e.g.* LFW (Huang et al., 2007), KIT-TV (Bäuml et al., 2013)), and many other niche challenges.

As the above tasks become easier and reach saturation, the community is veering away from these classical vision problems and actively addressing more complex problems such as image and video description (Guadarrama et al., 2013), text-to-video alignment (Zhu et al., 2015), and question-answering (Antol et al., 2015; Weston et al., 2015a) which require a deeper understanding and reasoning about the image and language content.

AI and Language. As in human development, vision and language are natural allies and often complement each other. While children learn the alphabet and first few words by relating them with real-world physical objects, more abstract concepts and words (*e.g.* "abstract" itself, or "thought") stem from a deeper understanding of language. Similarly for AI, analyzing vision hand-in-hand with language, Natural Language Processing (NLP) is a natural step forward in improving understanding.

Re-emergence of Recurrent Neural Network (RNN) units such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) or development of new ones (*e.g.* Gated Recurrent Unit (GRU) (Cho et al., 2014)) have helped train character- or word-level models to predict the next character or word in a sequence. Intelligent modifications are being made to the classical forms of these networks, to, for example, model attention (Bahdanau et al., 2015). Simultaneously, semantic vector representations for words (Mikolov et al., 2013) and sentences (Kiros et al., 2015b) are becoming commonplace and are used in various applications.

Videos, unlike static images or text engage multiple human faculties at once (such as auditory, visual, and emotional) and are a very powerful medium of conveying stories. However, addition of the temporal aspect to images and presenting the story with shots and scenes makes videos exceedingly complicated. Automatically understanding the story conveyed by videos and making them accessible for search and summarization is the next big frontier for machine understanding of language and vision. While user video upload

statistics set new records year after year (*e.g.* YouTube[1]), most user-generated video is often quite chaotic, unstructured, of low quality, and may not necessarily contain a clear narrative. As this is among the first attempts towards story understanding, we narrow our domain to structured videos produced primarily for the purpose of story-telling, namely *TV series* and *films*.

1.1 Many-faced stories

Machine *understanding* is often used vaguely, as it doesn't entail a specific machine learning task such as classification. In this thesis, *machine understanding of stories* conveys the ability of a machine to "watch" or "read" a story and be able to perform human-like tasks. For example, we humans are very good at telling the story in a shorter form (summarization), answering questions about it, or retrieving specific parts from the story. People, being innate pattern recognizers, are also very good at relating stories with each other (even when the actors of the events, both human and non-human are different) and creating such constructs as genre, or classifying stories based on their end result (*e.g.* tragedy).

Complete understanding of stories is indeed a lofty and ambitious goal, however, we are at a point today where we see various manifestations of the same story. For example, a large number of books (especially in the fiction genre) are converted to TV series or movies. This provides us with not only an original source book, but a complete professionally made video resembling key parts from the book. The video additionally comes with associated meta information such as dialogs in the form of subtitles and filming scripts. In addition to these, fans of most TV series or movies collaborate to create short descriptions (*plot synopses*) of the story and upload them freely to internet fan sites or Wikipedia. Thus, the core concept of the story is available in various forms and we show in this thesis that a joint analysis of multiple facets of stories provides a unique opportunity to alleviate some of the difficulties in understanding them.

A different approach to story-telling is through static visual depictions in the form of comic strips or timeline charts. A popular example is the chart depicting Napolean's losses during the Russian campaign of 1812[2]. Another example, which is more applicable

[1] YouTube Statistics: https://www.youtube.com/yt/press/statistics.html
[2] https://upload.wikimedia.org/wikipedia/commons/2/29/Minard.png

to our domain is the XKCD comic strip[3]. Such a visualization presents a nice overview of the interactions between characters of the story, and when annotated with information about events or locations can help viewers recall story events faster. Motivated by such examples, the thesis also proposes an automatic method to generate such timeline charts that attempt to provide people with a big picture overview of story. Such charts are demonstrated to be useful in sifting through an episode of video data.

1.2 Vision and language

While humans develop a keen sense of the world through vision, language is a strong requisite for well-functioning society. Similarly, as machines learn to make sense of their visual world by detecting objects or identifying people, language is the best non-invasive human-machine interface. As fundamental techniques for analyzing texts and images develop in NLP and CV respectively, their joint analysis and synthesis (text/image generation) becomes the next natural step.

Recent interest in tasks such as describing the content of images (Chen and Zitnick, 2015; Karpathy and Fei-Fei, 2015; Kiros et al., 2015a; Vinyals et al., 2015) and videos (Guadarrama et al., 2013; Rohrbach et al., 2015) is evidence to possibilities of end-to-end image and text models for generating textual descriptions for visual data. In fact, many of the above methods treat captioning as a problem of machine translation – a classical and popular task in NLP.

On the other hand, the arguably harder task of generating images has also garnered interest. Denton et al. (2015) work on generating images starting from noise and a class name, while Mansimov et al. (2015) demonstrate image generation capabilities which can handle simple input sentences.

While these techniques are far from being able to generate text describing the story of a video, or generating high quality images, an intermediate and critical step is the creation of large data sets addressing such a problem. Our work dealing with alignment of plot synopses with TV episodes has the potential to generate concise single sentence descriptions for video clips ranging from 1 to 5 minutes. On the other hand, aligning text from books with their video adaptations provides data to describe video clips with multiple paragraph of text.

[3] http://xkcd.com/657

Matching data from different modalities is a popular task, and has been extensively studied and discussed for our case of vision (videos) and text (plots and books). We further discuss work related to joint analysis of vision and language in Sec. 2.1.

1.3 Contributions and outline

We analyze various aspects of story understanding in this thesis. Automatically abstracting the story from a TV episode or movie is an incredibly hard challenge. Nevertheless, we take a first stab at this problem and propose to use plot synopses to help us do this. Additionally, we also propose to use books for which there exist film adaptations. In sharp contrast to plots, books provide very rich, and detailed descriptions of characters and events. Understanding the story is often about getting a big picture overview of the entire timeline. To this end, and inspired by a web comic, we propose an automatic visualization technique to show character interactions interspersed with event information. Finally, question-answering is emerging as an easy-to-evaluate and robust means to test machine understanding of data. We create a benchmark data set for question-answering on story-related aspects of movies. Here, the questions are collected using plot synopses.

The thesis is structured as follows:

Chapter 2: Background and related work. The first part of this chapter starts by presenting some examples of the growing interest in joint analysis of vision and language, followed by a discussion on specific instances of work related to the contributions. In particular, we analyze previous work in the areas of our contributions: text-to-video alignment, video retrieval, story visualization, and the emerging field of question-answering.

Chapter 3: Preprocessing. Analyzing large amounts of data in the form of video and text requires some basic preprocessing steps to make the data usable. We present methods used to process the videos, from grouping video frames into shots to semantic grouping of shots into *scenes*. We also analyze the various sources of text that are used in the thesis and present an overview on existing methods to parse the text. The latter half of the chapter presents the methods used to identify people in videos. We highlight our contributions in the area of improving person identification, namely, face track clustering and improving the quality of automatically obtained weak labels to train character-specific models. The former, face track clustering was performed in collaboration with University of Oxford.

Chapter 4: Aligning TV series and films with plot synopses and books. A major contribution of this thesis is the introduction of alternative forms of text to help improve the process in which machines understand and analyze videos. We are the first to use plot synopses to analyze TV series and show direct applications in story-based video retrieval. We are also the first to jointly study books and their adaptations in the form of TV series or films. We propose methods to compute the similarity between very different visual and textual domains and present several methods to align the two story representations. While there are numerous applications of the text-to-video alignment, in this thesis, we focus on three particular areas: (i) rich descriptions for video shots; (ii) story-based search within large video collections; and (iii) finding differences between book and video adaptations.

Chapter 5: StoryGraphs. While plot synopses and books provide supporting information to a machine, we propose a method to provide users with a big picture overview of the story. The visualization of a story has direct applications in improving video retrieval. In fact, our method presents the entire timeline at a glimpse and displays interactions between characters along with special event labels obtained automatically from plot synopses. Our visualization methodology is generic and can be easily extended to analyze other scenarios involving people-people interactions such as meeting rooms.

Chapter 6: Question-answering in stories about movies. A tangential approach to analyze machine understanding of stories is to ask the machine to answer questions about a story. This is a recent and emerging field of study and is a promising direction of research. We create a benchmark data set *MovieQA*, which consists of 15,000 questions collected from over 400 movies. Each question consists of 5 multiple choice options, only one of which is correct. All our answer options are highly plausible choices when a human or machine does not know the story. We present various baselines to answer the questions while analyzing data set bias. We also discuss techniques based on word matching, propose a neural architecture for answering multiple-choice questions, and discuss important modifications to memory network architectures that were specifically built for question-answering. This work was performed in collaboration with University of Toronto.

Chapter 7: Conclusion. We provide a summary of the thesis and outline the main contributions made in the field of improving story understanding through joint analysis of vision and language. We also present multiple directions for future research, both low-hanging fruit and a potential direction in which the area might be headed.

1.4 Published contributions

Contributions presented in this thesis have been published at various venues. Proposed modifications to person identification were presented in (Tapaswi et al., 2014c, 2015b). The plot synopsis to video alignment along with story-based video retrieval was introduced at Intl. Conference on Multimedia Retrieval (ICMR) (Tapaswi et al., 2014a) and an extended version was published in (Tapaswi et al., 2015c). Alignment of videos to books was presented at the Conference on Computer Vision and Pattern Recognition (CVPR) (Tapaswi et al., 2015a). StoryGraphs, the visualization of character interactions as a timeline was presented at CVPR (Tapaswi et al., 2014b). Finally, the *MovieQA* benchmark data set has been made publicly available in late March 2016, and is seeing enthusiastic response from the community. The data set along with several answering baselines will be presented at CVPR 2016 (Tapaswi et al., 2016).

A full list of my publications, including work not contained in this thesis can be found in Appendix 7.2.

Chapter 2

Background and Related Work

While the past few years have seen a resurgence in joint analysis of vision and language, this area has witnessed some impressive early work setting the trends in this field. In this chapter, we lay the foundations and present related work in all the contributions of this thesis. We first present an overview of the relevant literature in the broader area of vision and language, jointly analyzing images/videos along side words/sentences (Sec. 2.1). While most previous work relied on template-based techniques, more recent approaches have proposed the use of end-to-end trainable models. This is followed by an survey of work related with text-to-video alignment (Sec. 2.2) including a short motivation for our work with semantic forms of text. As applications of video analysis, retrieval and summarization are popular problems. We discuss a few previous approaches that attempt to solve them (Sec. 2.3), and discuss a crucial shortcoming that need to be surpassed.

One part of the contributions of this thesis is a fully automatic method to present a big picture visualization for the story. We discuss and review work explicitly working in the general area of story visualization (Sec. 2.4). Finally, as AI gets smarter, the community is looking towards tasks which require high-level reasoning such as question-answering. We present a discussion of growing interest in question-answering both in the textual and visual domain (Sec. 2.5), and briefly compare previous data sets and answering approaches to ours.

2.1 Modeling vision and language

As presented in the introduction for vision, data sets and benchmarks are often critical to effectively measure the progress of a particular task. We open this section by a discussion of few data sets that start this multimodal journey, and navigate through the years discussing technical progress, especially seen in the fields of joint vision and language modeling and captioning.

Among the first works in joint vision and language understanding is the task of *image tagging*. Like modern photo-sharing websites (*e.g.* Flickr), the goal of this task is to automatically assign concept or semantic tags to images. The Corel5k data set (Duygulu et al., 2002) is among the first such data sets and consists of 5000 images tagged with 371 words. The availability of words and tags spawned a variety of approaches ranging from analyzing the joint distribution of image regions and words (Barnard et al., 2003), applying sparse coding for automatic annotation (Wang et al., 2009), indexing pictures using tags for retrieval (Li and Wang, 2003), combining text analysis texchniques (bag-of-words) with SIFT-like region descriptors (Sivic et al., 2005a), to encouraging collection of more data through well designed computer games (von Ahn and Dabbish, 2004). Seeing the interest in this area, Carneiro et al. (2007) created Corel30k, an extended version of the Corel5k data set, consisting of over 30,000 images and 950 words.

The next move was towards increasing the complexity of language interactions. Images were annotated with short descriptive captions of their content resulting in a flurry of data sets: Pascal 2008 (Everingham et al., 2015), Flickr8k (Hodosh et al., 2013), Flickr30k (Young et al., 2014), SBU corpus (Ordonez et al., 2011), and more recently MS COCO (Lin et al., 2014b). Flickr8k, presented by Hodosh et al. (2013) presented an alternative perspective to look at the subjective task of image captioning. They framed the problem interchangeably as ranking textual descriptions given an image, or ranking images using textual descriptions.

Increasing the complexity on the vision side was the next natural step. Here, images were replaced by short video clips obtained from user-generated content or production-quality movies. While the YouTube2Text data set was collected to analyze paraphrasing (Chen and Dolan, 2011), it also served as data for video description (Guadarrama et al., 2013). Domain specific videos related to cooking activities were addressed as the next problem, resulting in creation of the YouCook (Das et al., 2013), TACoS (Regneri et al., 2013; Rohrbach et al., 2013) and TACoS Multi-Level (Rohrbach et al., 2014) data sets. Related

to this thesis, the video captioning community also collected video clips from movies aligned with corresponding Descriptive Video Services (DVS) – an audio track specifically included for visually impaired people that describes the visual aspects of the video. These clips are considerably longer than YouTube2Text (Chen and Dolan, 2011) and have much more "wild" content as compared to the cooking data sets. Both data sets, MPII-DVS (Rohrbach et al., 2015) and M-VAD (Torabi et al., 2015) have been introduced very recently and are a testimony to the growing interest in semantic analysis of video content.

2.1.1 Multimodal modeling

Modeling language has greatly benefited several vision problems. One among these is through the use of Bayesian networks to model not only nouns, but "prepositions" and "adjectives" to express their relationships (Gupta and Davis, 2008). A different approach investigates the use of WordNet (Miller, 1995) subgraphs to build object hierarchies which improve recognition performance (Marszalek and Schmid, 2007). In the case of sports videos, Gupta et al. (2009) propose a technique to combine action recognition and model events using AND-OR graphs to learn a storyline from annotated video.

There are plenty of works in the area of image tagging, some of which are discussed below. Huiskes et al. (2010) use captions, or tags, with low-level image features to improve multimodal modeling. Similarly, Guillaumin et al. (2010) learn strong Multiple Kernel Learning classifiers using both image content and keywords to annotate unlabeled images. Ngiam et al. (2011) and Srivastava and Salakhutdinov (2012) present Robust Boltzmann Machines as generative models to jointly capture properties of the image and text data. These models not only predict tags for an image, but are also able to use tags to perform image retrieval. Kiros et al. (2014) introduce multimodal neural language models that jointly learn word representations and image features through convolutional networks.

Zero-shot learning and attributes. Zero-shot Learning (ZSL) is another field that can also be thought of as dependent on joint image and text analysis. In this paradigm, classifiers are learned for *unseen* classes based on characteristic properties borrowed (transferred) from *seen* classes. Attributes (*e.g.* colors, patterns, shapes, living/non-living characteristics) act as intermediary knowledge sources and see popular use in zero-shot learning (Farhadi et al., 2009; Ferrari and Zisserman, 2008; Lampert et al., 2009; Parikh and Grauman, 2011) .

With the advent of deep CNNs for images (Krizhevsky et al., 2012), and word vector representations (Mikolov et al., 2013), learning joint embeddings for image and text samples has grown in popularity. Frome et al. (2013) present deep visual and semantic (word) embeddings, which can also be used for zero-shot learning. Ba et al. (2015); Elhoseiny et al. (2013) use Wikipedia articles (large documents of text) to analyze similarity between unseen and seen classes and learn classifiers for unseen classes using most related seen classes.

While not a part of this thesis, we have work in this direction where we propose to solve a critical problem with attribute-based ZSL. Class-attribute associations needed for knowledge transfer are typically manually defined, and this severely limits the scalability of such approaches to large number of classes (*e.g.* ImageNet (Russakovsky et al., 2015)). In Al-Halah et al. (2016), we use word embeddings for classes and attributes and learn associations through semantic relationships, resulting in dramatic improvements in ZSL performance.

2.1.2 Automatic image and video description

Image and video captioning is another instance of joint vision and language understanding. This challenging task is often seen as a Statistical Machine Translation (SMT) problem, where the input language is the image (video) and the output language is the corresponding textual description. This is evident when especially considering the use of popular SMT evaluation metrics – the BLEU (Papineni et al., 2002), ROUGE (Lin and Hovy, 2003) or METEOR (Banerjee and Lavie, 2005) scores – to evaluate captioning.

Image and video captioning can both be partitioned into two research directions. The former is based on language modeling and involves detecting the objects (and maybe actions) in the image, through which a sentence is generated using language constraints (templates or based on n-grams). The other direction uses sequence generators (such as RNNs) and involves training of end-to-end models to learn a joint embedding for image and caption space. Below, we discuss each approach in more detail.

Template and grammar based models. In language modeling approaches, sentence generation is usually performed with syntactic constraints. Early in the domain of street surveillance Nagel (2004) provides an overview of the efforts to understand what happens in a street scene. The car detections and tracks are modeled using a hand-crafted Situation

Graph Tree, and a description is generated by filling in appropriate template sentences such as "Obj2 entered the lane. Later obj6 entered the lane. The vehicles formed a pair.". More recently, image description was attempted for open-ended images (*e.g.* those obtained from Flickr). Farhadi et al. (2010) perform object and action recognition to infer Subject-Verb-Object triplets which are converted to sentences using predefined templates. Ordonez et al. (2011) create a large data set of 1 million images and captions and argue that any new image can be described by leveraging the caption of an existing, similar-looking image. Kulkarni et al. (2013) present a system to generate detailed descriptions that learn from vision recognition algorithms (*e.g.* by listing all visible objects in an image). However, unlike real image captions which focus only on the important objects of an image, the Babytalk system (Kulkarni et al., 2013) details every object. Yang et al. (2011) produce descriptions using HMMs as sentence generators which take in estimates from visual detectors.

Similar template, or data-driven approaches can be seen for videos. Barbu et al. (2012) present a system that attempts to describe "who did what to whom, where and how". They use a fixed vocabulary consisting of a verbs, nouns, adjectives, and other parts of speech to generate such descriptions. Much earlier, Kojima et al. (2002) operate in a simplistic environment, however propose a procedure to generate sentences from videos, a very complicated task for their time. They first estimate human actions and identify objects, followed by a conceptual hierarchical description of body parts. The next step involves generating a whole body expression which is transformed into natural language. Das et al. (2013); Rohrbach et al. (2013) generate a semantic representation of visual content by training a Conditional Random Field (CRF) to model relationships between various visual inputs. The second stage involves a rule-based procedure to transform the semantic representation into language.

End-to-end models. The second direction involves end-to-end trainable models which are *not* explicitly provided with a list of objects present in the image. Several examples (Chen and Zitnick, 2015; Fang et al., 2015; Karpathy and Fei-Fei, 2015; Mao et al., 2015; Vinyals et al., 2015; Xu et al., 2015a) of this type have been presented in the last year. Notably, Karpathy and Fei-Fei (2015) present a framework to learn correspondences between parts of an image caption with objects in the image. The authors chain a region-CNN (Girshick et al., 2014) framework (selective search (van de Sande et al., 2011) and CNN features from each object proposal) with a bi-directional RNN to generate a

score between parts of the image and the sentence. At test time, this can be also used to generate captions. Mao et al. (2015) decompose image captioning into a three-step approach: detecting words by Multiple Instance Learning, generating sentences using a language model, and re-ranking the sentences using a deep embedding. Xu et al. (2015a) incorporate the concept of attention (popularly used for machine translation (Bahdanau et al., 2015)) on top of previous work (Vinyals et al., 2015). The attentive model looks only at salient image sub-regions and is thus more likely to describe important objects in the scene.

In the case of videos, the end-to-end model is closer to the sequence-to-sequence task (Sutskever et al., 2014) of machine translation. For the SMT problem, an encoder RNN looks at a sentence of the input language, and a decoder RNN generates the output sentence in the desired language. Venugopalan et al. (2015b) generate a mean pooled image representation of all frames in the video and use image-captioning techniques to generate descriptions. In contrast, Donahue et al. (2015); Venugopalan et al. (2015a) use CNNs to learn single frame representations which are fed as inputs to the encoder RNN. Similar to the spatial attention of Xu et al. (2015a), Yao et al. (2015) introduce attention in the temporal dimension of the video to generate captions, while using spatio-temporal convolutional networks (Ji et al., 2013) to obtain local action features.

Our contribution. Clearly, there is encouraging recent work in the domain of joint image and text modeling and image and video description. Our thesis makes use of some notions of image-text modeling, however does *not* focus on any of the above classical tasks – tagging or description. While the current section gives a big picture of the current progress in joint vision and language modeling, in the next few sections, we discuss literature more closely related to the problems addressed in this thesis.

2.2 Text-to-video alignment

As presented above, there has been a large amount of work on joint analysis of text and video. Given a video (TV series episode or film) and a related text document (subtitle, script, *etc*.) another form of joint analysis is to align parts of the video with relevant parts of the document. For a detailed description of the types of text documents analyzed in this thesis, we refer the reader to Sec. 3.2. A key point to note is that subtitles (not much different from closed captions) are a unique form of text document that not only contain

the dialogs, but also the corresponding timestamps when they are displayed. Thus, by nature, subtitles are already aligned to videos, and this fact is heavily exploited for various applications and approaches as we will see soon.

In their seminal work, Everingham et al. (2006) proposed to leverage subtitles along with fan transcripts to automatically obtain training data for character-specific face models. As both sources – subtitles and transcripts – contain dialogs, aligning them with each other is a fairly easy task. Aligning subtitles that come with timestamps with transcripts that provide speaker names allows to obtain cues about who is speaking when in the video. This information was exploited to attach weak (not necessarily correct) labels to (speaking) face tracks, which were in turn used to train character-specific models allowing fully automatic person identification in videos.

So much was the impact of Everingham et al. (2006), that it spawned a large number of papers (*e.g.* (Cour et al., 2009; Köstinger et al., 2011; Sivic et al., 2009)) addressing person identification through the use of subtitles and transcripts. We too did work in this area, proposing to use weakly labeled as well as unlabeled face tracks to train character models (Bäuml et al., 2013). We also worked on improving the assignment of weak labels, which directly impacts the trained face models (Tapaswi et al., 2015b) (presented in Sec. 3.5.2).

Subtitle and transcript alignment opened up avenues for multiple other tasks, such as shot threading and scene detection (Cour et al., 2008). While the transcripts used by Everingham et al. (2006) minimally require names and dialogs, they often contain information about what goes on between dialogs (scene descriptions). Such descriptions were used to create realistic action recognition data sets based on Hollywood movies (Laptev et al., 2008; Marszalek et al., 2009).

A slightly different approach was justly proposed by Cour et al. (2010). While transcripts are very helpful in furthering video analysis, they are often difficult to obtain. In addition, when we humans watch a TV series or movie, we do not need to be told about how characters look. This knowledge is gained by listening to the dialogs between characters. Cour et al. (2010), through intelligent use of various information sources proposed a means for transcript-free person identification. We too worked in the area of using only subtitles to perform person identification (Haurilet et al., 2016). We model the problem in a Multiple Instance Learning (MIL) framework, propose methods to automatically create bags using very sparse labels from subtitles and analyze and compare several MIL approaches for obtaining character-specific training data.

Nevertheless, over the past few years, transcripts have held their stay and are also seeing use in joint analyses. Liang et al. (2013) propose a different application where they use transcripts to create a video archive with containing characters, place and time information. Given a new script, they choose appropriate data from the archive, and after post-production are able to automatically generate new videos. Ramanathan et al. (2014) present a method to jointly address the problem of co-reference resolution in text and person identification in videos. Bojanowski et al. (2013) combine the principles of Everingham et al. (2006); Laptev et al. (2008) and learn joint person and action classifiers. Later, Bojanowski et al. (2014) introduce action ordering constraints that further help improve action recognition. Bojanowski et al. (2015) move away from the symbolic labels defined earlier (Bojanowski et al., 2014) and attempt to directly align larger chunks of text with video clips. While this is evaluated on a data set of cooking activities (Regneri et al., 2013), this is in essence similar to our attempts of aligning natural language text with videos. The alignment problem is formulated as a quadratic program and solved using an efficient conditional gradient algorithm.

The problem of aligning text documents with videos is also applicable in the context of different types of videos. For example, in the domain of sports videos, (Xu et al., 2008) uses webcast text to perform event detection. However, this is typically easier than aligning documents with TV episodes as sports videos typically contain a timer indicating the time progressed in the match. In the domain of autonomous driving videos, Lin et al. (2014a) parse textual queries as a semantic graph and use bipartite graph matching to bridge the text-video gap, ultimately performing video retrieval.

Coming back to the videos of concern in this thesis, namely TV series and films, Roy et al. (2014) present a data set benchmark for enabling reproducibility among researchers. The data set comes with multiple aligned sources of information (subtitles, transcripts, episode summaries and outlines) and other meta data such as (speakers, shot and scene boundaries). Two additional data sets in this domain are based on transcribed text from Described Video Service (DVS), a narration service for the visually impaired that along with the standard audio track recounts the visual aspects of the video (Rohrbach et al., 2015; Torabi et al., 2015). These data sets come as clips along with their descriptions, and are being used to train video/text embeddings (Zhu et al., 2015) or as a real-world video description benchmark.

A few months after our work on aligning books with videos, Zhu et al. (2015) presented an approach for a similar task. In contrast to our work, they propose to obtain only

fine-grained matches between video shots and book sentences aiming for high precision at the cost of recall. The similarity between book sentences and video shots is computed by training a joint embedding using the DVS data set (Rohrbach et al., 2015) and a global movie/book alignment is obtained through a chained Conditional Random Field (CRF).

Sankar et al. (2009) consider a different problem of aligning videos with transcripts in the absence of subtitles. Their alignment model is based on dynamic programming and acts as a baseline for one of our methods. Using visual features to obtain location and person information, and along with speech, the paper proposes an alignment that combines the multiple features into a single cost function. We will elaborate on the differences between the methods in Sec. 4.4.

Our contribution. Aligning text documents with video has provided a large impetus for analyzing complex story-telling videos (TV series and films). Not only have the alignments provided a large source of realistic data, this has often come at little-to-no additional manual annotation cost. We wish to raise the semantic content of such text documents and with this desire introduce two new and diametrically opposite forms of text sources – *plot synopses* and *books*. We propose and evaluate several methods to align them with their corresponding videos (see Chapter 4). These two text sources afford unique applications at the highest semantic level – the *story* – that were unachievable before. In Sec. 4.6 we discuss some applications that arise from such an alignment.

2.3 Video retrieval and summarization

Searching and summarizing video content has been a major challenge for the past decade. This has been promoted largely by the TRECVID evaluation campaign (Smeaton et al., 2006) featuring several tasks ranging from semantic indexing (concept tagging for media) to instance search (finding particular people) to multimedia event detection (event tagging for videos). Covering all works in this area is virtually impossible, nevertheless, we attempt to provide a broad overview of various approaches taken to address this problem.

Video retrieval. A major breakthrough in the area of image and video search was the paradigm shift of focus from low-level image based retrieval (*e.g.* color, shape) to concepts (*e.g.* scenes, people, events) (Snoek and Worring, 2007). Hauptmann et al. (2007) evaluate whether such concepts can help bridge the semantic gap between concepts and video

retrieval. Their study of video retrieval in broadcast news concludes that even though the concept detection performance is low (10% accuracy), when provided with a large number of concepts (5000), good results can be achieved.

From the computer vision perspective the content of videos (*e.g.* objects, people, actions) also helps improve retrieval. Early works in this area is by Sivic and Zisserman (2003), who propose a method to retrieve objects in videos through SIFT matches and rank them efficiently using text-retrieval approaches (TF-IDF). Soon after, Sivic et al. (2005b) develop an approach for analyzing face sets (tracks) in videos and retrieving shots based on characters. This influential work presented face tracking in TV series and triggered a series of papers on automatic person identification in TV series (Bäuml et al., 2013; Cour et al., 2009; Everingham et al., 2006; Sivic et al., 2009).

Speech (dialog) is a critical factor in retrieving videos in large audiovisual collections. Huurnink and de Rijke (2007) analyze the impact of speech in multiple TRECVid test sets and show a significant boost in performance. Additionally, they show that content-based analysis can also help in improving retrieval in archives (manually transcribed video collections) (Huurnink et al., 2012).

While never-ending learners (Chen et al., 2013; Divvala et al., 2014) and learning concepts or event detectors are a popular video analysis task (Bhattacharya et al., 2014; Dehghan et al., 2014; Gan et al., 2015; Sun and Nevatia, 2014; Xu et al., 2015b), there are instances of other data sets for large-scale video retrieval. Revaud et al. (2013) present a data set (Event Video – EVVE) and approach to search for other similar clips in large-scale video collections. The EVVE data set consists of about 3000 videos related to the queries and an additional 100k videos which act as distractors.

Perhaps closest sounding to our contribution is the work by Peng and Xiao (2010) where they perform video retrieval in news vides. However, there are several points of difference between the two works. First, and most importantly, the "stories" used by Peng and Xiao (2010) are obtained from news videos. We wish to assert that films and TV series are not only larger videos (thus making retrieval within the video harder), but are also more complex from the story-telling standpoint. Secondly, the paper proposes retrieval-by-example, *i.e.* clip-based retrieval in contrast to our use of natural language queries. Finally, Peng and Xiao (2010) use a combination of concept- and content-based features to represent video clips and perform ranking using a classifier. As we will see in Sec. 4.3, this is vastly different from our proposed approach.

Video summarization. Typical video summarization involves generating keyframes or compiling moving image summaries from larger videos. Other types of outputs include building mosaics (sports, parking lots), video browsing frameworks, and generation of trailers (movies). We see early approaches attempting to generate keyframe-based summaries (DeMenthon et al., 1998; Gong and Liu, 2000) or mosaics from the video shot (Aner and Kender, 2002). Over the years, several papers have approached video summarization through different lenses. Li et al. (2006) present a method for skimming through large videos (films), lucidly explaining the concepts of shots and sub-stories. An alternative approach to summarization is generating movie trailers. Irie et al. (2010) propose a system which mimics trailers by extracting audio-visual segments from movie using affective content analysis.

Closer to our work with stories, Chen et al. (2009) propose a structural video content browsing framework along with entities such as who, what, where and when. The framework allows users to not only browse the video efficiently, but also allows to focus on content-specific parts.

Characters play a very important role in any film and story. Tsoneva et al. (2007) present an approach to generate moving-image summaries of storylines. They extract textual features (*e.g.* keywords) from subtitles and movie scripts and along with character names create an importance function to identify important moments of the storyline. Comprehending the importance of characters, Sang and Xu (2010) also propose a method to score movie shots and scenes based on character interactions. Important shots are then used to create summaries.

Summarization and browsing is also an active area of research for *web videos*. For example, Khosla et al. (2013) present an approach to summarize user-generated videos using image priors as maximally informative frames of a video. They also propose an automatic scheme to evaluate keyframe based video summarization converting the nature of video summarization evaluation from subjective (requiring human raters) to objective. More recently, Kim et al. (2014) present an approach to jointly summarize Flickr images and YouTube videos. The idea uses images to reduce noise and redundancy in videos and videos to glue fragmented images.

Our contribution. As is evident, video retrieval and summarization have developed gradually over the many years. However, rarely has anyone attempted to search within the story line of the video, or presented a summary considering the big picture overview

of the story. Given the set of impressive vision and language analysis tools available to us now, we believe that we are at a juncture which can enable analysis at the level of the story for videos.

We propose story-based retrieval and hypothesize a scheme for summarization as direct applications of aligning videos to high-level descriptions in the form of plot synopses. Story-based video retrieval refers to the task of searching for story events (*e.g. Buffy stakes Dracula*) in a video. To the best of our knowledge, we are the first to work on enabling search for story events. We hypothesize a scheme to perform video summarization using the plot as a guideline. In addition, visualization of the story as a chart (Chapter 5) can also be seen as a pictorial form of video summarization.

Note that "shots containing X and Y" is not a query related to the story, while "X and Y play a game of chess" is a valid query. As we will see in Sec. 4.6.2 our queries are fairly complicated and are obtained partially from a fan-website.

2.4 Story visualization

Visualization can play a critical role in depicting the events of a story. Tufte (2001) presents sound theories and good practices in the design of data graphics and includes a wealth of amazing illustrations in the world. Among them, we see the example by Charles Minard, who accurately portrays a cartographic depiction of Napolean's losses during the Russian campaign of 1812[1]. Such type of band graphs illustrating flow are called Sankey diagrams. Visualizations have the power to present the key interactions at a glimpse. Another example, closer to this thesis is the depiction of character interactions in movies. Illustrated by Randall Munroe in `XKCD:657`[2] the graphic presents multiple visualizations for fan-favorite movies.

The visualization community contains a lot of work on guidelines and practices for good visualization (Battista et al., 1998; Byron and Wattenberg, 2008; Graham and Kennedy, 2003; Tamassia et al., 1988). In particular, metro maps are a famous example where aesthetically pleasing, legible, and topologically correct graphics are desired (Nesbitt, 2004; Nollenburg and Wolff, 2011). While we discussed a few examples of hand-drawn graphics, and best practices, there are only a few works which attempt to visualize stories (especially for films) automatically.

[1] https://upload.wikimedia.org/wikipedia/commons/2/29/Minard.png
[2] http://xkcd.com/657/

Ercolessi et al. (2012a) present StoViz, a story visualization tool for TV series which provides users with episode overviews by separating different storylines. Relying on speech analysis and simple color histograms, they first detect scene boundaries and perform plot de-interlacing (Ercolessi et al., 2012b). The sub-plots of an episode are separated and tabulated in the visualization tool.

Closest to our StoryGraphs is the work by Tanahashi and Ma (2012). Like our goal, they too aim to generate timeline charts depicting temporal dynamics of interactions in films (similar to the examples in `XKCD:657`). However, one of the critical drawbacks is that the meta information (who appears when) essential to generate the graphs stems from clean manual annotations. Tanahashi and Ma (2012) treat graph layout as an optimization problem, and solve it using genetic algorithm. In particular, each genome goes through 3 steps of computation: laying interaction sessions, re-arranging lines, and removing white spaces. The algorithm iterates until it minimizes a fitness function consisting of the number of wiggles and crossovers. They also present post-processing schemes to relax line wiggles and deemphasize lines by modifying their width.

Shrestha et al. (2013) attempt to incorporate spatial and temporal information in the generated depictions. They generate maps for war logs based on army movements, and plot latitude-longitude information across time. The geometric information (latitude-longitude) is transformed and plotted as a straight line between the two coordinates (on either side of the plot) while temporal events on this geometrical line are highlighted as data points. Here too, the meta-data is assumed available, and the only problem is plotting.

Our contribution. We propose a method for drawing StoryGraphs for TV episodes and do not assume that the visualization meta-data is provided. Using the alignment to plot synopses, we obtain event labels; and using transcripts, we identify all characters in the episode. Armed with scene boundary detection, and knowledge of which character appears when, we formulate graph generation as an optimization problem. We also perform experiments showing that such graphs can help reduce the time required to search for story-related events. For more details please refer to Chapter 5.

2.5 Question answering

A great way to test one's understanding about a particular topic is to check whether he/she is able to answer questions about it. Recently, the AI community has demonstrated that this is not limited to humans, but can be easily extended to machines. In most Question-Answering (QA) systems, AI attempts to answer questions based on a story which takes the form of text, image, video or a combination thereof. The answering itself can be done in several ways and typically consists of proposing the correct word (or set of words), or choosing from a set of multiple choice answers. The latter scheme is limited to a given set of choices and can be considered close-ended answering. Truly open-ended answering, where the AI is expected to generate sentence- or paragraph-long answers is still an open problem, not least due to the complexity of automatically evaluating such answers.

QA is a perfect example of vision and language research coming together. In the premise of our thesis, this is a great challenge to test machine understanding of stories.

Question-answering based on textual knowledge representations has seen a lot of work (Bao et al., 2014; Clark et al., 1999; Ferrucci et al., 2010; Hovy et al., 2001; Kiyota et al., 2002), however, we are interested in QA systems that work with stories (textual and visual). Knowledge-based QA systems are often called factual QA, and while certainly relevant for search engine applications (Agichtein et al., 2001), they are not the focus of this thesis.

In the following, we describe story related QA data sets and approaches. For simplicity, we categorize them depending on the modality of the story (textual and visual).

2.5.1 Text-based QA

Text-based QA deals with a passage of text containing the story about which several questions are framed. Among the early works, Richardson et al. (2013) presented a data set for machine comprehension of short stories. The complexity of stories (and questions) is limited to that of a 7-year-old child's vocabulary. Using crowd-sourcing, the workers create 150-300 word stories around a wide range of children topics (*e.g.* vacations, school, fairy tales, spaceships). Associated with each of these 660 stories are 4 multiple-choice questions consisting of four answer options each, only one of which is correct.

In contrast, Hermann et al. (2015) propose a QA data set whose stories stem from a decade of news articles from the CNN and Daily Mail. The news articles are available along with their summary in the form of a few bullet points. The data set is constructed by replacing key entities from such summaries and converting them into Cloze style questions (Taylor, 1953) (similar to fill-in-the-blanks). While this yields a massive data set (1 million questions and stories), the question complexity is limited due to the fill-in-the-blank style.

Recently, Weston et al. (2015a) present a QA data set explicitly categorizing the questions based on their complexity. For example, some of the simpler sets include single/two/three supporting fact(s), argument relations, yes/no questions, counting, and negations. The questions also attempt to cover various aspects of reasoning which require to understand the flow of time, deduction, induction, co-reference, path finding, *etc*. The answer to these questions is one of the words from the vocabulary. A strong aspect of the data set is the split into 20 tasks that aim to identify where machines face the most difficulty. However, due to the synthetic nature of the questionnaires, each task has very limited vocabulary (20-40 words) and consists of short stories (median of 9 sentences, each of 5.8 words on average) and questions (5.3 words on averages with single word answers.

Answering approaches. Several text-based answering techniques have been developed to address the QA problem. They range from simple lexical approaches (*e.g.* bag of words (Richardson et al., 2013)) to various RNN architectures (Hermann et al., 2015), and even special kinds of neural architectures designed specifically for answering questions: Memory Networks (Sukhbaatar et al., 2015; Weston et al., 2015b).

Examples of RNN architectures to solve QA is seen in the work by Hermann et al. (2015). They present multiple approaches building upon the Long Short-Term Memory (LSTM) to solve the fill-in-the-blanks problem. Their models incorporate attention while reading the story (Attentive Reader), and even while words of the question become visible (Impatient Reader) which yield performance improvements.

The first memory network was proposed by Weston et al. (2015b). The network consisted of a scheme to evaluate the similarities between various parts of the story and the question. Many hand-crafted additions were made to the basic similarity function to allow for word sequences and time, and to handle exact matches and previously unseen words. The model was refined and simplified by Sukhbaatar et al. (2015) enabling end-to-end training of the network (from learning word embeddings to picking the correct answer). This end-to-end memory network (MemN2N) learned embeddings for the story sentences.

Given a question, it used a form soft-attention to pick relevant story sentences, and produce single-word responses from the vocabulary. Multiple layers of the attention module helped answer long-range dependency questions (*e.g.* the "three supporting facts" bAbI task).

Several other works have attempted solving the textual QA problem. Smith et al. (2015) and Wang et al. (2015) propose stronger lexical matching methods incorporating contextual windows and coreference resolution for MCTest. Kumar et al. (2015) present a Dynamic Memory Network which introduces the concept of episodic memories that can be used to retrieve facts conditioned on the question. Li et al. (2015) start with Graph Neural Networks (Scarselli et al., 2009) and modify it to include gated recurrent units. Such a network is shown to have favorable performance as compared to RNNs or LSTMs on the bAbI tasks.

2.5.2 Vision-based QA

While QA developed initially for text sources, with the rising performance of vision tasks such as object detection and action recognition, the vision community started working towards image-based question answering. In this area, Malinowski and Fritz (2014) created a QA data set – DAQUAR – by framing questions on images taken from the NYU-RGBD data set (Silberman et al., 2012). The questions test machines for spatial reasoning, counting, analyzing shapes, colors and locations of objects. The answers are typically single words, but can also consist of word-lists (*e.g.* "What is on the desk?").

While the DAQUAR data set consisted of about 12,500 question-answer pairs obtained from 1449 RGBD images, Antol et al. (2015) present a large scale data set and benchmark for visual question answering. The first release of Visual Question Answering (VQA) contains 250k images obtained from 2 data sources – 200k real-world images from MS COCO (Lin et al., 2014b), and 50k synthesized images from Abstract Scenes (Zitnick and Parikh, 2013) – resulting in over 750k question-answer pairs. All questions are collected by crowd-sourcing and the paper presents a detailed analysis about question types based on the first few words in the questions. Answering is either open-ended (but single word) or one from a large list of multiple choices (18 candidates) created by considering the correct answer, few plausible answers (obtained by humans) and the 10 most popular answers. Typical questions include analysis of object color, type, shape, and location or yes/no questions and counting.

A third data set in this rapidly growing area is Visual Madlibs (Yu et al., 2015). The data set is crowd-sourced by fill-in-the-blank templates, and is also built on the MS COCO (Lin et al., 2014b) image data set. There are 12 types of templates designed to gather information about people, objects, their interactions, activities and general scene context. While not a typical question-answering data set, the fill-in-the-blanks is presented as a multiple-choice complete the sentence task. Particularly interesting are tasks which ask the machine to predict what may have happened before and after the picture was taken.

Answering approaches. Along with the DAQUAR benchmark, Malinowski and Fritz (2014) propose a probabilistic framework to combine natural language input with visual scene analysis. Antol et al. (2015) analyze the images with pre-trained CNN models (Krizhevsky et al., 2012) and answer the question, optionally with the help of image captions from the COCO data set. Malinowski et al. (2015) present and end-to-end framework for visual QA. They represent the image as a CNN, and feed such a representation to an LSTM along with words from the question. The LSTM is trained so as to provide the correct answers when sampled after the end of the question. Following image captioning techniques, Yu et al. (2015) present CNN+LSTM baselines for question answering on the Visual Madlibs data set. Inspired by machine attention mechanisms in image captioning (Xu et al., 2015a), there are works which attempt to answer questions by allowing the machine to attend to specific parts of the image (Xu and Saenko, 2015). Zhang et al. (2015) focus in particular on binary VQA by converting the question to a relation tuple and classifying them as yes/no. Very recently, Andreas et al. (2015) present a modular approach to building custom neural architectures for visual question answering. Motivated by the superior performance of neural networks, they decompose the question into a set of instructions each of which can be answered using a stack of neural modules.

Our contribution. Our MovieQA data set allows for both textual and visual QA. Built upon complex stories (movies), this is also the first data set to contain videos. The presence of temporal aspect, questions of type "why" and "how" make the data set unique as compared with previous work. See Chapter 6 for a detailed comparison, and description of the proposed data set. Our technical contributions include a novel convolutional network for question-answering in addition to significant modifications to the MemN2N to make it suitable for large vocabulary and multiple-choice QA.

Chapter 3

Preprocessing

We present the toolchain used to prepare both text and video data, *i.e.* convert it from its raw state to a structured form with meta information. We talk about the various text sources used in this thesis and some steps taken to process them. For videos, we briefly discuss their general structure building up from shots to scenes, and end with a discussion on identifying characters in them.

While all components of the toolchain are not novel, the key contributions are:

- A short introduction to novel forms of text – plot synopses (Sec. 3.2.4) and books (Sec. 3.2.5) – that form a core component of the thesis and are the main topic of Chapter 4.
- A scene detection algorithm which leverages shot threads and provides optimal scene boundaries with respect to a shot grouping criterion (Sec. 3.4.3).
- Two improvements in the field of person identification: (i) face track clustering (Sec. 3.5.1); and (ii) improving the quality of automatic labels used to train character models (Sec. 3.5.2).

3.1 Data sets

Research on TV series and movies has often been done with different data sets, as there is no common benchmark in the community. In fact, the data set proposed in our previous work on identifying characters in TV series (Bäuml et al., 2013; Tapaswi et al., 2012) has been downloaded and used by teams around the world.

Here, we describe the different video series used throughout this thesis. While discussing the evaluation of methods, we will revisit the data set section, list the series and episodes used and propose justifications for our choice.

The Big Bang Theory (BBT) is a traditional situational comedy (sitcom) featuring a small (∼10 characters) recurring cast list, many of whom appear together in scenes. We work on season 1 of this TV series. The episodes are 20 minutes long, and are shot in well-lit typically indoor scenes, and a season is loosely connected by story continuity.

Scrubs (SCR) is another sitcom where the story of the protagonist takes place primarily at a hospital and in his apartment. While the primary cast list is quite small (∼10), the hospital setting provides a large number of unknown background characters (patients, nursing staff), and includes long takes with significant camera motion. We work with multiple 20 minute long episodes from season 1.

Buffy The Vampire Slayer (BF) is a supernatural drama series consisting of a protagonist fighting off forces of darkness. Each episode is about 40 minutes in length, includes a self-contained story, and is part of a coherent storyline through the season culminating in a final episode with a face off against a "boss" demon. Thus, each season is a perfect test bed to study story progression and analyze machine understanding of stories. We work with season 5 of this series, where the number of recurring characters is ∼20.

Game of Thrones (GOT) is a fantasy drama set in a medieval world with a large number of characters (∼50) and families vying for power and control of the kingdoms. Episodes last for about 1 hour, and cover multiple story lines for different characters, often located in far away countries. This high budget TV series is based on the book series *A Song of Ice and Fire* and unlike standard television, presents very high quality cinematography much similar to a movie. However, each episode is not self-contained, and in fact, since the story corresponds to several books, a season too does not necessarily have a closed story. Nevertheless the number of parallel story lines in the series present an interesting case study for automatic story understanding.

Harry Potter series (HP) is a fantasy book and film series where the protagonist learns to use magic to defeat an evil arch-enemy. As compared against previous data sets, the

series consists of multiple films, each on average about 150 minutes in duration. While the primary cast list is short (∼30), there are a large number of background characters. Similar to GOT, the series is based on books and thus provides an additional means to approach the story.

Assorted movies. In Chapter 6 of this thesis, we use a collection of over 400 movies based on which we gather a story question-answering benchmark data set. The movies cover all genres, filming styles and span a period of 70 years. Stories from the movies are analyzed through both text and video modalities.

3.2 Sources of text

As discussed in the introduction (Sec. 1.1), the same story is available today in a variety of forms. We use various forms of text to help analyze the different levels of story abstraction of the video. The text sources contain large variations in their semantic content. For example, on one hand subtitles provide only the dialogs between characters, while plot synopses on the other extreme summarize the story of an episode without mentioning any dialogs.

Note that text, in particular subtitles and transcripts have been used successfully to identify characters in the video (Everingham et al., 2006) and the actions they perform (Laptev et al., 2008). Our contribution in this thesis is to explore two additional high-level and diverse forms of text: story summaries - *plot synopses*; and the original story presented with a lot of details - *books*.

3.2.1 Subtitles

A text which contains timestamped dialogs from the video qualifies as subtitles. Often, they contain non-verbal audio information such as `(engine roars)` and are called *closed captions*. Note that subtitles are similar to the output of a near-perfect automatic speech recognition system that can reproduce the spoken text and timestamps, but is agnostic with respect to speaker identities.

Subtitles for a TV episode or film are easiest source of text to obtain. They are not only available on all DVD or Blu-ray versions of the film, but are also easy to find online.

We present a typical example of subtitles below. Note how each dialog is timestamped to a certain duration. This timestamp will help us localize texts with video. Another interesting aspect of subtitles is when the dialogs in a short timespan are uttered by different characters. In such a case a hyphen is inserted at the beginning of the dialogs.

```
00:04:27,733 -> 00:04:30,008
Just call me the Computer Whisperer.

00:04:30,093 -> 00:04:32,527
Let's get scanning.
I wanna see this puppy go.

00:04:32,613 -> 00:04:36,97
- Start with those.
- Start?  Where is finish?
```

3.2.2 Scripts and transcripts

A text which (minimally) contains the name and dialog of the speaking characters is a transcript. Note that in most cases real production scripts[1] provided to the actors are hard to obtain (*e.g.*, due to copyright restrictions). In fact, after post-production, the scripts may not even follow the released video. However, fans often provide various levels of transcription of the dialogs on the internet[2].

While the original scripts contain extra information about the scene setting (INT/EXT - interior/exterior, time and day, location, actions, *etc*.), fan transcripts often do not. We restrict ourselves to the information necessary for person identification, namely character names and dialogs.

Here, we present an example from a transcript[3] corresponding to the subtitles displayed presented above. While the transcripts group character and dialog information, they may also add information about simple actions (*e.g.*, "Giles puts a pile of old books on her outstretched arm").

[1] https://sites.google.com/site/tvwriting/ or http://www.imsdb.com/
[2] *e.g.*, http://bigbangtrans.wordpress.com/
[3] http://www.buffyworld.com/buffy/transcripts/079_tran.html

GILES: Thank you, Willow. Obstinate bloody machine simply refused to work for me. (Walks off)
WILLOW: Just call me the computer whisperer. (Stands up, putting something in the scanner) Let's get scannin'. I want to see this puppy go.
Giles puts a pile of old books on her outstretched arms.
GILES: Start with those.

Subtitles and transcripts are easily aligned using the common dialog information and are a rich source of information about "*who* speaks *what* and *when*" and "*who* is performing *what actions*". Subtitles and transcripts are aligned using Dynamic Time Warping (DTW) algorithm as proposed by Everingham et al. (2006). Throughout this thesis, we use such a scheme to identify characters in video. The details are discussed in Sec. 3.5.

3.2.3 Descriptive Video Service

Commercial productions such as movies and large budget TV series contain an audio track describing the video to enable visually impaired people to watch and understand the video. Recently, Rohrbach et al. (2015) and Torabi et al. (2015) built large scale data sets to analyze complex video content. They provide video clips and the corresponding video descriptions obtained from the Descriptive Video Service (DVS). The DVS text can be thought to be a proxy for perfect visual analysis, and is in some ways what a machine should "see" in the video. A drawback however is that DVS data is especially hard to obtain and shows concerns with respect to scalability as they are not created by many low-budget films and TV shows.

DVS: They rush out onto the street.

Script: Valjean and Javert hurry out across the factory yard and down the muddy track beyond to discover -

A man is trapped under a cart.

A heavily laden cart has toppled onto the cart driver.

Valjean is crouched down beside him.

Valjean, *Javert and Javert's assistant* all hurry to help, but they can't get a proper purchase in the spongy ground.

An example of DVS is shown above for a few shots of the video (taken from Rohrbach et al. (2015)). Errors in the script are highlighted in red.

The data sets based on DVS are introduced very recently, and we will present an example usage of DVS to compute joint video and language embeddings and answer questions about the story of movies in Chapter 6 of this thesis.

3.2.4 Plot synopses

A major contribution of this thesis is the introduction of plot synopses to better aid story understanding of videos. Plot synopses, unlike scripts or DVS, are a concise description of the story of the movie or TV episode and are easily obtained from Wikipedia. They rarely contain the actual dialog (may occur only when quoting something very important) that appears in the videos.

Wikipedia contains a nice article which contains guidelines on how plot summaries should be written[4].

> "The description should be thorough enough for the reader to get a sense of what happens and to fully understand the impact of the work and the context of the commentary about it. [...] Plot summaries that are too long and too detailed can be hard to read and are as unhelpful as those that are too short."

We present a few plot synopsis sentences below and highlight the sentence that spans the story excerpts corresponding to the examples for subtitles and transcripts.

> *Giles has Willow start scanning books into a computer so that they can be resources for the gang to use.* He then tells her that he's going back to England because it seems he's no longer needed by Buffy or the Scoobies.

In Chapter 4 we will show how such plot synopses can be aligned to the video. We will also present story-based video retrieval as an application that arises naturally from such an alignment.

[4] https://en.wikipedia.org/wiki/Wikipedia:How_to_write_a_plot_summary
Retrieved 4 April 2016

3.2.5 Books

In the recent decades, TV series and films are often adapted from books. Popular examples in the fantasy genre include *The Lord of the Rings* trilogy, the TV series *Game of Thrones*, and the *Harry Potter* movie series. There are also examples in many other genres such as The *Bourne* series (action), *Hunger Games* (adventure), *House of Cards* (political drama), *etc*.

While sticking to the same overall storyline, books present the information in a very different way. Typically, book chapters are interspersed with narrative paragraphs and dialog. Book dialog often forms the base of the video dialog, while most narrative content is conveyed visually. We believe that books and videos are a massive untapped resource for future analysis. In fact, soon after our work, there were key contributions in the field of sentence representations using a large book corpus (Kiros et al., 2015b) and also aligning books with movies (Zhu et al., 2015).

Below is a example from a book chapter[5] presenting the discussed variation in narration followed by dialog between characters situated at the location.

> And one day fifteen years ago, this second father had become a brother as well, as he and Ned stood together in the sept at Riverrun to wed two sisters, the daughters of Lord Hoster Tully.
>
> "Jon ..." he said. "Is this news certain?"

We present techniques to align books to parts of the video adaptation in Chapter 4. In particular we develop an approach that explicitly handles the situation where the video adaptation may not follow the book. As an application, we discuss using narrative paragraphs from the book for video description.

3.3 Text processing

As presented above, this thesis deals with both structured (subtitles and transcripts) and natural language (plot synopses and books) forms of text. We discuss some common text processing tools which are used throughout this thesis. In particular, *plots* and *books* are processed with several additional steps. We use the Stanford CoreNLP toolkit (Manning

[5] George R. R. Martin, Chapter 2. *A Game of Thrones*, Bantam Publishers, 1996.

et al., 2014) to perform tokenization (breaking sentences into words) and part-of-speech (POS) tagging (labeling each word with its type, *e.g.* noun, verb, adjective, *etc.*) on all text forms. As we will see further, our primary cues involve character names and are found using POS tags corresponding to proper nouns (NNP) and pronouns (PRP). We have also experimented with word stemming, and when performed word stems are calculated using Porter (Porter, 1980) or Snowball (Porter, 2001) stemmers from the Python Natural Language Toolkit (NLTK) (Loper et al., 2009).

Plot synopses are processed with anaphora resolution (Lee et al., 2011) to associate pronouns (*e.g.* he, she) with their corresponding names.

Books, obtained as electronic versions are first divided into chapters. Each chapter is divided into paragraphs, which are classified as narrative or dialog depending on whether they contain a dialog. For simplicity, only the direct form of dialogs (*i.e.* within " ... " quotes) are considered.

More details about individual processing steps for plots, books or other forms of text will be discussed when used.

3.4 Elements of a video: shots, threads, and scenes

A video is a complex data source. While being represented by pixels, it not only has all the intricacies of an image (objects, scene, people), but also has an additional dimension of time that allows to depict action and change. From a broader perspective, thousands of such individual actions can be looked at together to convey a story.

Like most other video processing methods, we segment our videos into *shots* and *scenes*. Here we discuss the approach taken to detect shots and scenes in this thesis.

3.4.1 Shot boundary detection

A shot consists of a set of continuous video frames taken from the same viewpoint or camera. Cutting and Candan (2015) present an interesting analysis of video shots as a historical perspective for a large number of movies. A clear take home message is that the number of shots have increased steadily with the rise of digital technologies.

Shot detection is performed by computing *Displaced Frame Differences* (DFD) (Yusoff et al., 1998) between consecutive motion compensated frames. The feature value spikes

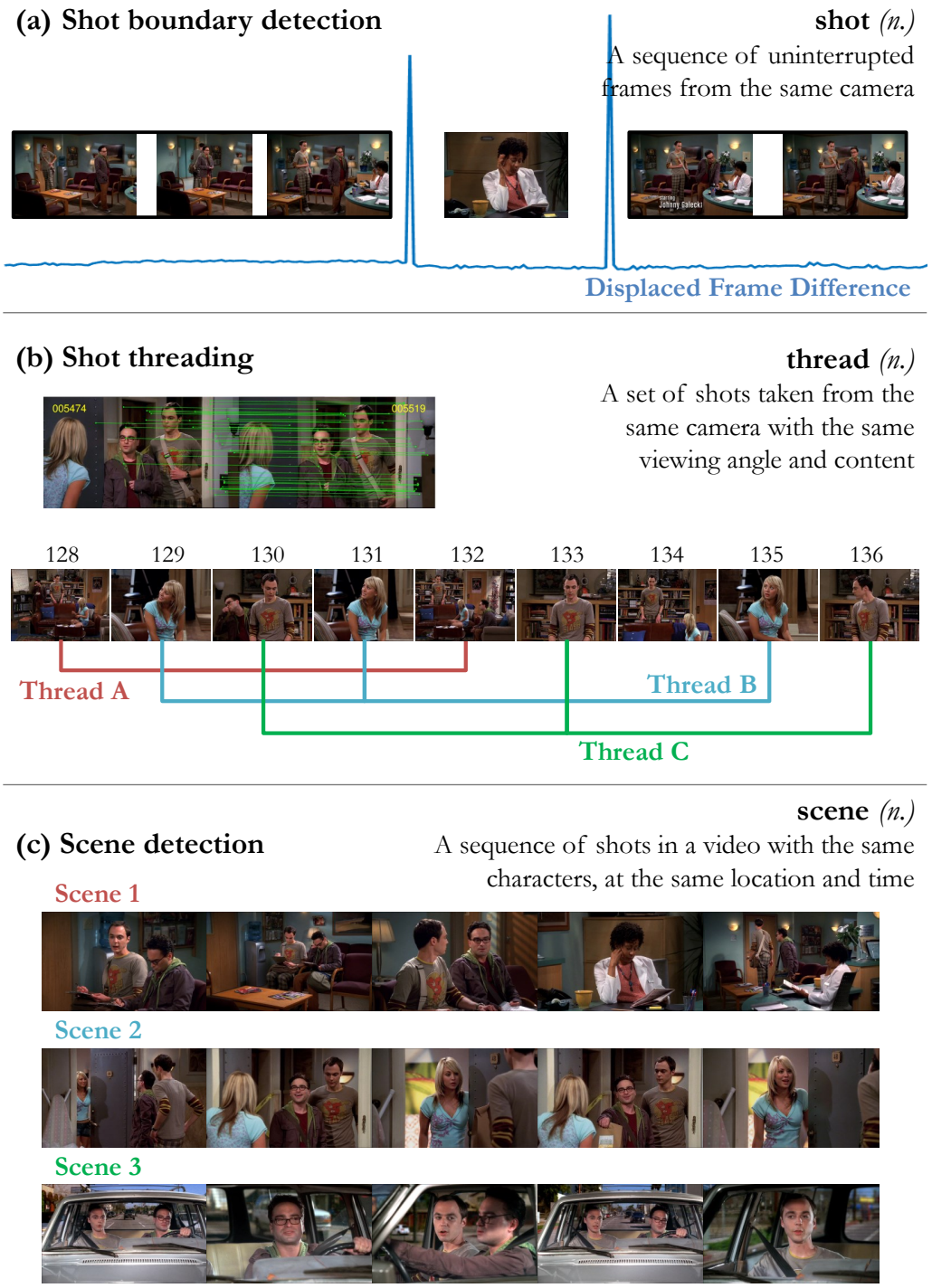


Figure 3.1: Video editing structure of a TV series episode or film. Individual frames are grouped into shots, whose boundaries are detected using Displaced Frame Difference (Sec. 3.4.1). Shots arising from the same camera viewing angle are grouped together to form a shot thread (Sec. 3.4.2). Such threads along with color information from shots are used to determine scene boundaries (Sec. 3.4.3). Threads also help in improving face clustering and identification (Sec. 3.5).

at the location of a shot change where the frame t looks very different from frame $t-1$. An advantage of DFD compared to simple color-histogram based methods is evident especially in video sequences with large camera motion (*e.g.*, action clips).

The image is first divided into macro-blocks (similar to MPEG coding) of 16×16 pixels and the DFD is computed between each frame t and its neighbor $t-1$ as

$$DFD(t) = \sum_{x,y} ||I_{16\times16}\big((x,y),t\big) - I_{16\times16}\big((x,y) - o(x,y,t), t-1\big)||, \tag{3.1}$$

where $I_{16\times16}((x,y),t)$ is the image macro-block at position (x,y) and frame t. $o(x,y,t)$ denotes the motion offset for the image block of frame t from the previous frame $t-1$, computed using a simple block matching approach.

The corresponding DFD feature is filtered using a series of opening and closing morphological operations to reduce the noise due to non-compensable large motions. After filtering, applying a threshold provides very good shot boundary detection performance.

Evaluation on the first 6 episodes of a sitcom (*The Big Bang Theory*) provides 1981 correct detections, with 2 misses and 8 false positives. We see similar performance when analyzing other data sets qualitatively. The impact of errors in shot detection on scene detection, face tracking or future text-to-video alignment are negligible and ignored.

Fig. 3.1(a) shows examples of shot boundary detection in a video along with the corresponding DFD signal.

3.4.2 Shot threading

As the number of shot changes increase, the same viewpoint is often presented as alternating shots. Shots which are taken from the same viewpoint and contain the same characters and background are referred to as belonging to a *shot thread*. A classical example is over-the-shoulder shots in a dialog between two characters. Such shots exhibit an A-B-A-B pattern, *i.e.* shots A and B are captured over the shoulders of the characters.

In fact, our analysis on two TV series (BF, SCR) showed that a surprisingly large number of shots, about 60-70%, are part of a shot thread. This number even holds true for high quality productions (*e.g.* GOT), where we see that 65% shots are part of a thread.

Shot threads have been proposed before by Cour et al. (2008), however, the method proposed to model the threading is extremely complicated. In contrast, drawing from insights in geometry, we first compute a homography between a pair of images using SIFT keypoints (Lowe, 2004) and apply RANSAC (Fischler and Bolles, 1981) filtering to build shot threads. To compute threads in the video, we compare only the last frame of a selected shot and the first frame of $R = 25$ subsequent shots. We then apply transitivity to obtain unique shot threads.

Fig. 3.1(b) presents an example of the SIFT matches procedure and resulting shot threads.

3.4.3 Scene boundary detection

This work is a part of our paper on generating StoryGraphs (Tapaswi et al., 2014b). We thank Prof. Andrew Zisserman for the insightful discussion about scene detection.

We define a *scene* as a set of shots in the video telling a sub-story. This typically means that the shots depict the same location (indoor or outdoor), a restricted group of characters, and present an event in a short time span. Scenes, unlike shots are not well defined, making scene boundary detection a fuzzy process, where two human annotators need not always agree upon all the locations of the scene boundaries in an episode. Note that, by the above definition, scene boundaries always occur at shot boundaries.

As aptly put by Han and Wu (2011), the problem of scene boundary detection involves: (i) measuring the coherence within a scene, *i.e.* the similarity between shots of a scene; and (ii) searching for appropriate boundary positions.

Related work. Scene detection in TV series and movies is a well studied problem. Notable examples include work by Rasheed and Shah (2005) which uses color and motion-based shot similarity cues and recursively apply normalized cuts on a shot affinity graph. Other work by Chasanis et al. (2009) first creates groups of shots using spectral clustering followed by a sequence alignment method. The idea is to declare a scene change when the difference between overlapping windows of shots is larger than a threshold. There is also work on scene detection with the help of supporting scripts. Liang et al. (2009) use an Hidden Markov Model (HMM) to map the scene structure of the script onto movies.

Most related to our proposed method is the work by Han and Wu (2011). Here, shot similarity is evaluated using normalized cut scores and dynamic programming (DP) is used to optimize placement of three boundaries (two scenes) at a time.

Proposed scene boundary detection. Our contribution to scene boundary detection is the incorporation of shot threads obtained via SIFT matches. We also propose a DP based model to find an optimal set of shots. In contrast to Han and Wu (2011) our method does not require to look at multiple scenes. In addition, it provides a simple way to automatically determine the number of scene boundaries and does not involve a hierarchical splitting approach as proposed by Rasheed and Shah (2005).

Our first cue to place scene boundaries involves analyzing the color-based shot similarity within a scene. The similarity score between a shot s and a set of shots P is computed based on the average distance between shot histograms

$$C_{s,P} = \phi\left(\frac{1}{|P|}\sum_{p\in P}||h_s - h_p||^2\right), \tag{3.2}$$

where h_s is the mean pooled $6 \times 6 \times 6$ RGB histogram of all frames in shot s, and $\phi(\cdot) \in [0,1]$ normalizes the distance to yield a shot-similarity score.

Our second cue is based on shot threading. Here, we compute a binary similarity value as

$$T_{s,P} = \begin{cases} 1, & \text{iff } s \text{ is part of a shot thread with any shot in } P \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

Intuitively, we wish to find the optimal grouping of shots to maximize within-scene color-based similarity, and encourage shots within a thread to belong to the same scene.

We formulate the problem of finding scene boundaries as grouping all video shots, N_{shot}, into sets S_i to form N_{scene} scenes. The set of all scenes taken together, $\cup_i S_i$ corresponds to all the shots in the video and each scene consists of a non-overlapping set of shots, *i.e.* $S_i \cap S_j = \varnothing$. The optimal grouping of scenes is obtained by attempting to place a shot

s in a scene S_i that consists of a set of shots P subject to the above constraints, while maximizing the color and threading similarities.

$$S^* = \underset{S}{\operatorname{argmax}} \sum_{i=1}^{N_{scene}} \sum_{(s,P)\in S_i} \alpha(|P|) \cdot (C_{s,P} + T_{s,P}). \tag{3.4}$$

We also introduce a regularization term that prevents scenes from growing too large, $\alpha(n) = 1 - 0.5(n/N_l)^2$. N_l is the maximum number of shots that may be assigned to one scene.

To solve Eq. 3.4 we perform an exhaustive search through all possible shot groupings. This is done efficiently using dynamic programming, and the solution corresponds to finding the highest scoring path through a cube of size $N_{scene} \times N_{shot} \times N_l$. The decay factor is influenced by N_l, which represents the maximum number of shots that can be assigned to a scene. While theoretically this is all the shots from the video N_{shot}, in practice, we see that using $N_l = N_{shot}/5$ provides similar results and a large speed-up. The third dimension of the cube is necessary to "count" the number of shots that have been assigned to the current scene, thus allowing us to regularize the similarity through $\alpha(\cdot)$.

Our scene detection model was inspired by the dynamic programming model used for aligning plot synopses sentences with video shots. More details of the model including the transition paths in the forward computation pass and the backtracking are presented in Sec. 4.3.3 (DTW3).

Unlike many other methods, our approach does not need to know the number of scenes a priori. In fact, the forward pass on the cube computes all possible combinations of shots into scenes ranging from one scene taking all shots, to each shot being its own scene. The optimal number of scenes is determined by finding the elbow point on the curve of scores in the last column of the cube. Note that the scores in a DP cube are monotonically increasing (from top to bottom), however, they reach saturation as creating new scene boundaries does not yield additional advantages (*e.g.* due to broken shot threads).

Evaluation. We compare our proposed method with Rasheed and Shah (2005), which uses hierarchical splitting using normalized cuts to reach a desired number of scenes. We evaluate our method (DP) on 2 episodes each from 3 TV series – *The Big Bang Theory* (BBT), *Buffy the Vampire Slayer* (BF), and *Game of Thrones* (GOT). These series are chosen as they cover a wide variety of genres: situational comedy, drama, and fantasy; and have

		BBT-1	BBT-2	BF-1	BF-2	GOT-1	GOT-2
$\mu(s)$	DP (ours)	**18.46**	**7.95**	**13.41**	**12.00**	**12.61**	**17.05**
	(Rasheed and Shah, 2005)	21.38	16.15	19.52	16.34	21.63	29.52
R_{30}	DP (ours)	**75.00**	**90.91**	**84.85**	**96.67**	**88.57**	**87.10**
	(Rasheed and Shah, 2005)	62.50	81.82	72.73	86.67	80.00	77.42

Table 3.1: Scene detection performance when the number of predicted scenes is set to be the number of minutes in the episode.

		BBT-1	BBT-2	BF-1	BF-2	GOT-1	GOT-2
$\mu(s)$	DP (ours)	**24.91**	**27.79**	**15.04**	**16.34**	**18.55**	**33.36**
	(Rasheed and Shah, 2005)	25.85	39.25	25.62	22.23	42.84	50.74
R_{30}	DP (ours)	**62.50**	**63.64**	**81.82**	**90.00**	**74.29**	**77.42**
	(Rasheed and Shah, 2005)	56.25	54.55	63.64	76.67	57.14	64.52

Table 3.2: Scene detection performance when the number of predicted scenes is exactly the same as number of annotated scenes.

different episode lengths, 20, 40, and 60 minute respectively causing variation in the number of scenes in each episode.

We measure the quality of scene detection using two criteria: (i) $\mu(s)$: the mean absolute deviation in seconds from the annotated scene boundary location to the closest predicted boundary (lower is better); and (ii) R_{30} (Rasheed and Shah, 2005): recall at 30 seconds is the percentage of predicted scene boundaries that lie within 30 seconds of a ground truth annotated boundary (higher is better).

Table 3.1 presents the results when we force the methods to generate one scene per minute of the episode. Such a scene detection scheme is used to generate StoryGraphs in Chapter 5. Table 3.2 shows the results where the number of automatically detected scenes is equal to the number of manually annotated scenes. While both our metrics essentially evaluate recall, creating the same number of scenes as the ground truth annotation implicitly evaluates precision. Note that we outperform a strong baseline with multiple cues and the use of normalized cuts (Rasheed and Shah, 2005) in both scenarios.

Fig. 3.1(c) displays example shots from scenes created using our proposed approach.

3.5 Person identification in videos

Most stories revolve around characters and their interactions. A crucial part of story understanding is being able to identify "who did what with/to whom". Thus, knowing who the characters are is a first and important step towards this problem. Character identities are an important cue in our work with text-video alignment. We primarily rely on our previous work (Bäuml et al., 2013) to identify on-screen characters. In addition, we also make some contributions in this field, that prove to be quite necessary to obtain good performance on the harder data sets (*e.g.*, *Game of Thrones*, *Harry Potter*).

Face detection and tracking. Faces are the most important cue to identify characters in video. Before classifying faces, we need to first detect and track them in the video. We employ a cascade classifier built on the Modified Census Transform (a Local Binary Pattern) descriptor to detect faces (Fröba and Ernst, 2004; Küblbeck and Ernst, 2006) in the video. A dense sliding-window scan is performed on the first frame of each shot to initiate face tracks and subsequently at every 5th frame to look for new characters or restore occluded tracks. Between the detections, faces are tracked via the *tracking-by-detection* paradigm using a particle filter (Bäuml et al., 2010). For a detailed performance analysis of the face detector and tracker, we refer interested readers to an earlier dissertation from our lab (Bäuml, 2014). For this thesis, we assume videos and their corresponding face tracks are available using the above method.

3.5.1 Face clustering

This work is the main topic of Tapaswi et al. (2014c) and was accomplished during an internship at Prof. Andrew Zisserman's lab at the University of Oxford. In particular, Omkar M. Parkhi helped with feature extraction, and Eric Sommerlade provided face tracks. More details about the evaluation can be found in the paper.

Face clustering in videos is a popular vision task (Guillaumin et al., 2009; Khoury et al., 2013; Ramanan et al., 2007; See and Eswaran, 2011; Wu et al., 2013a,b), and is often a precursor to actual identification. However, most above methods attempt to cluster face tracks down to the number of characters. Even with latest advances in (shallow) face descriptors (Parkhi et al., 2014) this has proven to be a very difficult task. Above mentioned methods achieve a purity around 60-70% as the number of clusters reduce.

We postulate that this is an unusable purity, especially when such a clustering is used for further applications. In this spirit, we aim to perform aggressive clustering (reduce the number of tracks as much as possible) while maintaining a very high purity above 99%.

Three-stage clustering approach. We propose to leverage the video-editing structure commonly found in TV series and movies (see Sec. 3.4 shots, threads and scenes) in a stage-wise approach.

(A.) In the first step, we obtain pairs of *must-not-link* tracks using temporal co-occurrence and threading patterns. While tracks that co-occur have been used as negative pair influences before, we obtain a large number of new pairs from shot threads.

(B.) The second step consists of *within-scene clustering*. Within a shot thread, track pairs that appear in the same spatial region are very likely to be the same person. For a pair of tracks $\{t_i, t_j\}$, we compute the Euclidean distance between track feature representations $d(t_i, t_j)$ and a region intersection-over-union score $\gamma(t_i, t_j)$. We train Support Vector Machine (SVM) classifiers that learn a threshold θ_{thr}. A pair of tracks is said to belong to the same person when

$$-w_d \cdot d(t_i, t_j) + w_\gamma \cdot \gamma(t_i, t_j) > \theta_{thr}\,. \tag{3.5}$$

Track pairs that are not part of any thread are compared directly based on Euclidean distance and merged using a threshold θ_{sc}. We observe that θ_{sc} is stricter than θ_{thr}, indicating that the threading allows to obtain more merges than would have been otherwise possible. Also, as we operate within the regime of one scene, we typically observe fewer number of characters leading to simpler clustering. We apply transitivity rules and create larger clusters by merging all track pairs.

(C.) The third step in our approach is clustering at the episode level. We first recompute the Fisher Vector descriptors for each face cluster obtained so far (from step B.) as they are good at aggregating representations (Perronnin and Dance, 2007). By propagating the negative track pairs obtained in A. through the within-scene clustering B., we obtain several negative instances for every track (cluster). These are used to train exemplar-SVMs (Malisiewicz et al., 2011) for each cluster, and the SVM prediction scores are in turn used to merge clusters at the episode level.

Through the use of the video-editing structure, we are able to train discriminative classifiers at both stages (B. and C.) without the need for manual labels.

Evaluation. We evaluate the proposed approach on 6 episodes each of two diverse TV series: *Scrubs* (SCR) and *Buffy the Vampire Slayer* (BF). We first present a detailed overview on some key numbers of the data set (see Table 3.3). Here, we highlight a few key aspects from these numbers:

- The number of shots and number of tracks for named characters is proportional to the video length (BF episodes are about twice as long as SCR).
- A large fraction (about 60-70%) of the shots are part of a threading pattern. This directly translates to more than 68% of tracks seen in shots which belong to threads.
- As compared to within-shot negatives obtained via temporal co-occurrence (214 average for SCR), threads provide a much larger number of negative pairs (851 on average for SCR) with a very high accuracy (over 98%).

As we are concerned with having pure clusters, we measure the clustering performance using the Weighted Clustering Purity (WCP). For a given clustering C, the metric weights the purity of each cluster by the number of elements.

$$\text{WCP} = \frac{1}{N} \sum_{c \in C} n_c \cdot \text{purity}_c \,. \tag{3.6}$$

Purity is measured as a fraction of the largest number of data samples that belong to one class to all the data samples in the cluster n_c. N is the total number of tracks. Note that $\text{WCP} \in [0, 1]$. In our original paper (Tapaswi et al., 2014c), we include additional metrics that count the number of clusters and the number of clicks required to fix errors in the clustering (Operator Clicks Index (Guillaumin et al., 2009)).

We compare our proposed approach against an aggressive hierarchical agglomerative clustering (HAC) baseline. Like our method, HAC is performed at two stages, within scenes and at the full episode level. A threshold is learned at each level, and all track pairs whose distance is lower than this threshold are merged.

Table 3.4 presents a detailed overview of the clustering performance for both TV series. For each method (and each episode) we display two numbers – the number of clusters, and the weighted clustering purity. Note that, we maintain the purity for our clusters close to 1.0, while reducing the number of clusters as compared to the baseline agglomerative clustering approach. We are able to halve the number of tracks for *Scrubs*, and significantly reduce the number of tracks in *Buffy*, both while creating highly pure clusters.

		SCR-1	SCR-2	SCR-3	SCR-4	SCR-5	SCR-23	BF-1	BF-2	BF-3	BF-4	BF-5	BF-6
I	shots	450	370	379	319	360	315	678	616	820	714	675	745
	threads	91	70	79	53	73	69	121	114	165	120	105	151
	scenes	27	21	24	25	23	21	37	35	37	48	39	37
II	named char.	17	12	15	17	16	19	11	15	13	15	18	18
	tracks for named	495	413	376	365	404	419	630	779	973	668	646	842
III	shots in thread	295	242	260	223	264	242	419	383	550	413	363	494
	tracks in thread	340	271	254	230	279	309	449	552	668	425	408	592
IV	negs in-shot	232	192	116	212	222	310	282	620	752	216	192	582
	negs A-B-A-B thread	136	104	114	88	174	316	176	240	172	58	170	164
	negs complex thread	1146	432	370	552	1078	596	1742	1672	1916	212	712	4114

Table 3.3: Various information about the face track data sets *Scrubs* (SCR) and *Buffy* (BF) along with cues from their video-editing structure. The first section (I) presents the number of shots, threads and scenes in each episode. This is followed by the number of named characters, and tracks associated with them (II). Section (III) presents an analysis of the number of shots and tracks that are part of a thread. Note how more than 60% tracks are part of a thread, which helps our approach obtain confident positive and negative pairs. The final section (IV) presents the number of negative pairs (negs) that can be obtained from within a shot (temporally co-occurring), within a A-B-A-B threading pattern, and other complex threading patterns. Within A-B-A-B threads, negative pairs are created by linking all tracks in thread A with tracks of thread B. For more complex patterns (*e.g.* A-B-C-A-C-A-B-C) we verify negative pairs by an additional threshold on the distance between the feature representations.

	SCR-1		**SCR-2**		**SCR-3**		**SCR-4**		**SCR-5**		**SCR-23***	
#tracks (#ideal)	495 (17)		413 (12)		376 (15)		365 (17)		404 (16)		419 (19)	
Within scene clustering												
Baseline (HAC)	319	0.996	233	1.000	222	0.997	200	1.000	251	1.000	232	1.000
Proposed part-1	293	1.000	202	0.994	205	0.998	181	1.000	217	1.000	212	1.000
Full episode clustering												
Baseline (HAC)	287	1.000	208	0.998	196	0.995	182	1.000	246	1.000	208	1.000
Proposed part-2	244	0.992	185	0.992	179	0.992	147	0.998	198	0.998	173	1.000

	BF-1		**BF-2**		**BF-3**		**BF-4***		**BF-5**		**BF-6**	
#tracks (#ideal)	630 (11)		779 (15)		974 (13)		668 (15)		646 (18)		843 (18)	
Within scene clustering												
Baseline (HAC)	537	1.000	697	1.000	852	1.000	567	1.000	584	0.999	761	1.000
Proposed part-1	501	1.000	655	1.000	814	1.000	524	1.000	550	0.999	717	1.000
Full episode clustering												
Baseline (HAC)	534	1.000	688	1.000	852	1.000	566	1.000	575	0.999	751	1.000
Proposed part-2	466	1.000	598	1.000	730	1.000	494	1.000	507	0.998	643	1.000

Table 3.4: Results of face clustering on *Scrubs* (SCR, top) and *Buffy the Vampire Slayer* (BF, bottom). Episode SCR-23 and BF-4 are used for training clustering parameters and thresholds. Below each episode, we show the number of tracks and the ideal number of clusters. For each method, we present the number of clusters and weighted clustering purity (WCP).

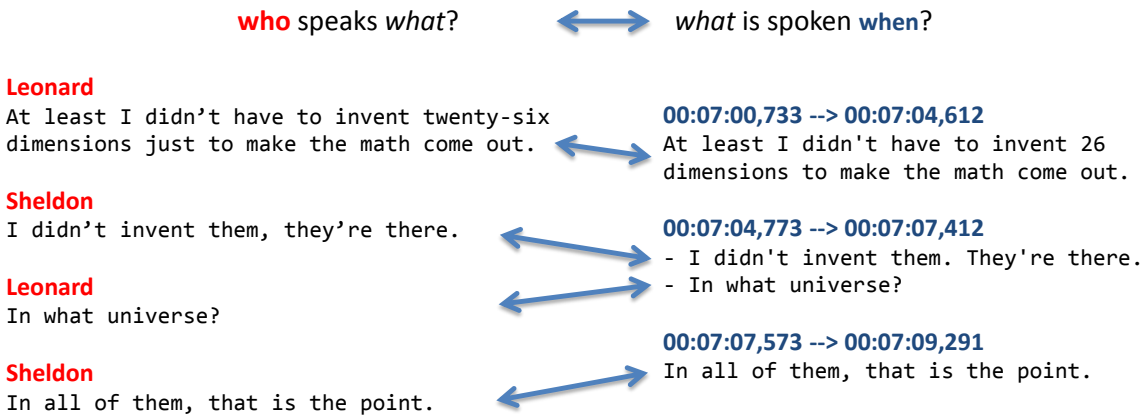


Figure 3.2: Subtitles are aligned with transcripts to obtain information about *who* speaks *when*. Aligning the two texts is fairly straightforward since most words in the dialog are a perfect match (Bäuml, 2014).

As face track clustering is only a minor contribution in the thesis, we request interested readers to refer to Tapaswi et al. (2014c) for more details.

3.5.2 Face-based identification using weak labels

This work was presented at the Automatic Face and Gesture Recognition (FG 2015) conference. More details, especially evaluation can be found in Tapaswi et al. (2015b).

As examined in related work, since Everingham et al. (2006), most work around person identification in TV series is performed automatically by first aligning subtitles with transcripts to obtain *who* is speaking *when*. Fig. 3.2 presents an example of such a subtitle-transcript matching. The names obtained via matching are associated with corresponding face tracks (*e.g.* based on lip motion), and such weakly labeled tracks are used to train character-based face models.

A critical component of such an approach is the quality of weak labels obtained to train person models. In a typical TV series, the subtitle-transcript matching assigns labels to 10-20% of all tracks with a precision of 80-90%. However, a major drawback of the above approach is that it assigns weak labels to tracks independently, without considering the video-editing structure or neighboring tracks that may influence this decision. We revisit the problem of weak label assignment (interchangeably called "speaking face assignment"), keeping in mind the resources available for joint labeling through the use of negative pairs and face track clusters (see Sec. 3.5.1).

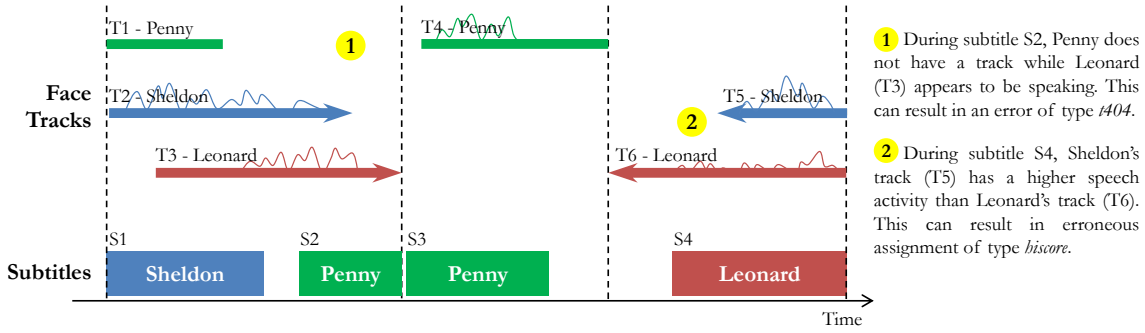


Figure 3.3: Overview of the weak label assignment of Everingham et al. (2006). Subtitles, their associated names and timespan are shown at the bottom of the timeline, while face tracks are shown on the top. Each character is represented with a unique color. The lip activity associated with a face track is shown above each track. The track threading is indicated by arrow heads for the track edges. We highlight the two scenarios of errors and describe them in the neighboring text.

Fig. 3.3 presents an overview of the weak label association depicting two common problems with the naïve approach. The first occurs when the speaking face track is not visible (either due to tracker failure or actor not speaking to the camera). We see an example where S2 (*Penny's* speech segment) is assigned to T3 (*Leonard*), the only available track since *Penny* is not tracked while she speaks. We call such errors *t404*, or track not found. The second is due to failures in quantifying lip movement. We see S4 (*Leonard's* speech segment) being assigned to T5 (*Sheldon*) as it appears that *Sheldon* is moving his mouth. Such errors are denoted by *hiscore*. Both these cases can be solved by looking at more than one track at a time, and we present a model to incorporate several of such cues and improve the quality of weak labels.

Joint weak labeling. We make several modifications to improve the quality (precision) and quantity (fraction assigned) of face tracks which are associated with weak labels.

Consider a set of N tracks $\mathcal{T} = (t_i, \mathbf{s}_i, y_i), i = 1 \ldots N$. Each track t_i is associated with a score measuring the amount of lip movement $\mathbf{s}_i$ and a ground truth identity y_i. Let weak labels be associated with M tracks with identities $\hat{y}_k, k = 1 \ldots M$. Through this discussion, we will use two primary metrics

$$\text{speaker assignment precision} = \frac{1}{M} \sum_{k=1}^{M} \mathbb{1}\{y_k = \hat{y}_k\}, \tag{3.7}$$

$$\text{speaker assignment fraction} = \frac{M}{N}. \tag{3.8}$$

We first discuss the process of scoring the lip movement for face tracks. While Everingham et al. (2006) relied on primitive facial landmark recognition, landmark localization has progressed dramatically. Unless severely occluded or rotated, landmark detection algorithms proposed by Ren et al. (2014); Xiong and Torre (2013) provide a very good estimate of mouth points. This enables us to simply calculate the distance between the upper lip center and lower lip center and use that as a measure of lip motion. We represent the speaking score for track t_i as $\mathbf{s}_i \in \mathbb{R}^C$, where C is the number of characters. We accumulate the lip opening for the track, ψ_i^u, in the duration when it has an overlap with subtitle u whose speaker is determined (via transcript matching) to be c and obtain

$$\mathbf{s}_i = \sum_u \hat{\mathbf{e}}_c \cdot \psi_u^c . \tag{3.9}$$

Here $\hat{\mathbf{e}}_c$ is a unit vector with zeros at all places except at dimension c.

The second part of our joint track association employs pair-wise information about tracks. We obtain negative pairs $\mathcal{N} = \{(t_i, t_j)\}$ of tracks as before using tracks that co-occur and are part of a thread. Positive track pairs $\mathcal{P} = \{(t_i, t_j)\}$ are also obtained by grouping tracks which are part of a thread (see discussion on face clustering Sec. 3.5.1).

We perform speaking face assignment, *i.e.* associate names with tracks using an energy minimization framework. For each track t_i, we define a random variable $\mathbf{x}_i \in \mathbb{R}^C$ that holds the probability of assigning one of the C names to the track t_i. We incorporate four energy terms:

1. *Speaking score term* ($\uparrow$): incorporates the similarity between $\mathbf{x}_i$ and $\mathbf{s}_i$. Note that, by definition, $\mathbf{s}_i$ is usually a sparse vector with a single non-zero entry at the dimension of the character obtained via subtitle-transcript matching. To assign track t_i to the speaker, we wish to maximize $\mathbf{x}_i^T \mathbf{s}_i$ subject to $\sum_c \mathbf{x}_i^c = 1$ constraint. In the absence of the other energy terms, the ideal value of $\mathbf{x}_i$ is a vector of zeros with a single 1 at the location of non-zero $\mathbf{s}_i$.

2. *Uniqueness term* ($\downarrow$): is a pairwise energy term that is calculated as $\mathbf{x}_i^T \mathbf{x}_j$ and applies to negative pairs of tracks $(t_i, t_j) \in \mathcal{N}$. Minimizing this term promotes assignment of different labels $\hat{y}_i \neq \hat{y}_j$ to the track pair.

3. *Threading term* ($\uparrow$): in contrast to the negative pairs, this term is maximized in the energy function formulation and applies to all positive track pairs $(t_i, t_j) \in \mathcal{P}$.

	BBT		BF	
	Baseline	M+U+T	Baseline	M+U+T
speaker assignment precision	89.8	94.3	85.9	93.5
speaker assignment fraction	17.9	18.0	18.7	18.8
errors: total	72	40	155	72
errors: *t404*	13	7	46	11
errors: *hiscore*	59	27	109	50
errors: *other*	0	6	0	11

Table 3.5: Error analysis of the weak label association. If we keep the fraction of assigned tracks constant, we see that the erroneous assignments are almost halved.

4. *Regularization term* (↓): our final term regularizes the values of $\mathbf{x}_i$ and makes sure that a small speaking score does not have a large effect on the values of $\mathbf{x}_i$. As $\mathbf{x}_i$ is constrained to sum to 1, the regularization essentially pushes $\mathbf{x}_i$ closer to $1/C$ (uniform prior). This term is calculated for all tracks as $\mathbf{x}_i^T \mathbf{x}_i, \forall t_i$ and is minimized.

The optimal values for the assignment random variables are obtained by minimizing the energy of the unary and pair-wise terms discussed above.

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \; -w_S \sum_i \mathbf{x}_i^T s_i + w_U \sum_{(t_i,t_j)\in\mathcal{N}} \mathbf{x}_i^T \mathbf{x}_j - w_T \sum_{(t_i,t_j)\in\mathcal{P}} \mathbf{x}_i^T \mathbf{x}_j + w_R \sum_i \mathbf{x}_i^T \mathbf{x}_i \quad ,$$
$$\text{subject to } \sum_{c=1}^{C} x_{ic} = 1 \quad . \tag{3.10}$$

We solve the above formulation using MATLAB's constrained function minimization routine `fmincon`, and assign weak labels for each track t_i as

$$\hat{y}_i = \underset{c}{\operatorname{argmax}}\, x_{ic}^* \, . \tag{3.11}$$

Note that through the use of a confidence threshold (based on $\mathbf{x}_i$), we control the assignment process and do not need to label all tracks at this time.

Evaluation The evaluation is carried out on the updated KIT TV data set (Bäuml et al., 2013) consisting of 6 episodes each of two TV series: (i) *The Big Bang Theory* (BBT) and (ii) *Buffy the Vampire Slayer* (BF). To make comparison to previous work easier, we use the same face tracks, and follow the same identification protocol (Discrete Cosine Transform (DCT) face representations and 1-vs-all polynomial kernel SVMs).

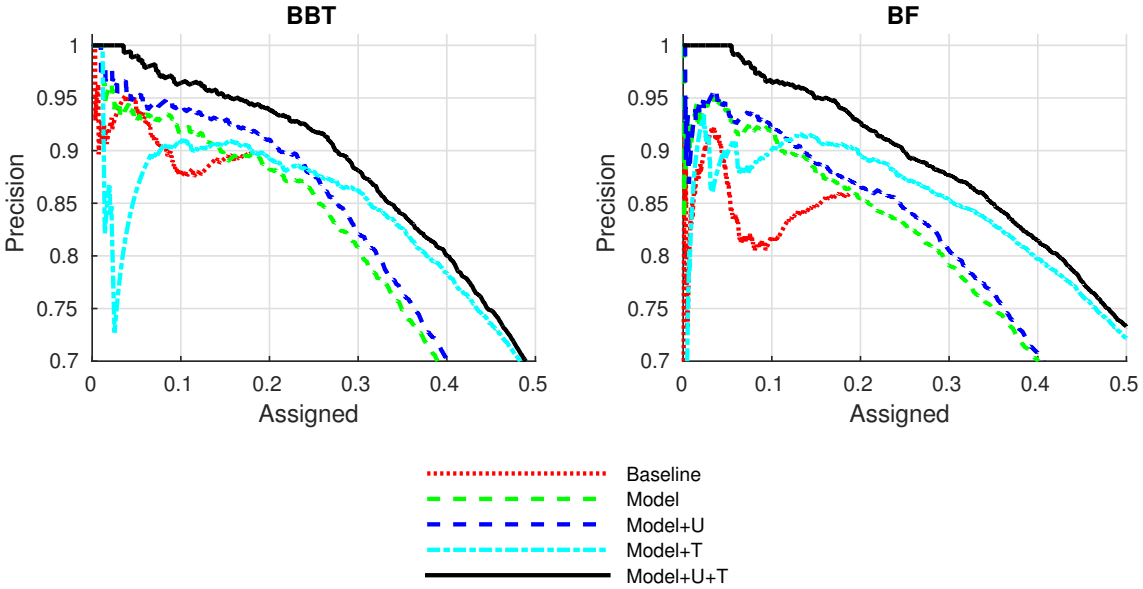


Figure 3.4: Speaker assigment precision vs. the fraction of tracks assigned a name for the two TV series - BBT and BF. Our model with all four terms ("Model + U + T") surpasses the baseline ("Baseline") by a large margin improving upon both precision and the number of assigned tracks.

The first part of our evaluation analyzes the weak labeling performance. Fig. 3.4 presents PR curves in the form of speaker assignment precision against the fraction of tracks which are assigned a label (and used to train the model). The performance of "Baseline" (Everingham et al., 2006) is very erratic and assigns labels to very few tracks. Using an energy minimization framework (only regularization and the baseline lip movement score "Model") we already see improved performance. Adding the uniqueness ("Model + U") and threading terms ("Model + T") improves performance significantly, while combining all terms ("Model + U + T") presents the best results. The figure shows performance while using the legacy lip movement descriptor for a fair comparison.

Table 3.5 shows the error reduction observed by using our model. At an operating point where the fraction of assigned tracks is maintained level, we are able to eliminate about half the errors. Both types of errors (*t404*: track not found; and *hiscore*: wrong track has higher lip movement score) are corrected by the proposed energy minimization framework which jointly considers the assignments to tracks.

Our constrained optimization approach is not only effective, but also very fast. As all tracks are not part of pair-wise connections, small cliques of 5 - 20 tracks are formed. This enables fast processing, as different cliques are independent of each other.

Using these tracks as training data, we train person-specific, polynomial kernel SVM classifiers for each character. We then classify and apply the label of the character whose model scores highest to each track. This results in a consistent improvement of about 1% in track identification accuracy (number of tracks labeled with the correct name). We do not detail the impact of the new lip movement descriptor, however, this further boosts both speaker assignment precision and final person identification accuracies by 2%.

The proposed improvements in weak labeling are necessary, especially in case of harder TV series such as *Game of Thrones*, where the number of tracks assigned using the baseline model is lower than 10%. In the interest of brevity, and as this is a minor contribution, we request the reader to refer to our original work (Tapaswi et al., 2015b) for a longer discussion and evaluation.

Chapter 4

Aligning Videos with Plot Synopses and Books

This chapter is a combination of work based on aligning videos with novel text sources. One part is derived from aligning plot synopses with videos (Tapaswi et al., 2014a, 2015c), while the second is based on the alignment of novels (books) with videos (Tapaswi et al., 2015a).

Comprehending the story of videos is a very difficult task. We are motivated by a large amount of previous work which has successfully leveraged text sources (primarily subtitles and transcripts) to improve video understanding. A thorough literature review about fantastic use of such text sources was presented as part of the related work analysis in Sec. 2.2. Most notable among these are the breakthroughs that were achieved in the fields of person identification (Everingham et al., 2006) and action recognition (Laptev et al., 2008) in videos.

In this chapter, we propose and investigate the use of two additional sources of text which aid to understand the video while focusing on aspects of the story. Recall that throughout this thesis we use *videos* to primarily mean videos produced to convey a story: *TV series* and *movies*. We first explore and motivate the two new forms of text in Sec. 4.1. Prior to using the text sources for semantic applications, we need to *align* chunks of text with parts of the video. Building upon preprocessing techniques discussed in Chapter 3, we bridge the gap between text and video and propose joint similarity functions to compare

the data from the two modalities (Sec. 4.2). We propose and evaluate numerous models that use the above similarity to obtain a suitable text-to-video alignment (Sec. 4.3). Finally, we present a thorough evaluation of these models in Sec. 4.5 and end the chapter with a discussion about the applications that arise from the alignment (Sec. 4.6).

4.1 Text sources

Sec. 3.2 presented a short overview about the myriad types of text sources used in this thesis. Here, we motivate, analyze and discuss the two new forms of text in more depth. Texts such as subtitles and transcripts come with tags (*e.g.* timestamps, scene information, speaker names) and can be seen as structured and meta sources of information. However, the two following types of texts contain complex narrative structures of stories and are purely composed in natural language.

4.1.1 Plot synopses

A short and complete textual summary of the story of a TV episode or a movie is commonly found on Wikipedia or other fan-sites. These summaries, typically consist of 30-40 complex sentences consisting of multiple clauses that provide a high-level overview of the important aspects of the story. On average, a sentence from the plot synopsis has over 20 words and represents about 1-2 minutes of video from TV episodes. In the case of movies, the video duration (2 hours) is far longer than standard TV episodes (40 minutes), while the number of plot synopses sentences is similar, thus giving them a more condensed feel.

Note that such plot synopses do not describe low-level events (*e.g.* door opens, *Jim* walks in), rarely contain dialogs (except when quoting an exceedingly important event) and are a complete spoilers-included synopsis of the story. That said, plot synopses often skip unimportant portions of the video. While this makes aligning parts of the plot with the video a fairly difficult problem, once resolved, many advantages can be gained through the use of such crisp story descriptions.

A direct application arising from the alignment between plot synopses and videos is *story-based video search*. Searching for queries in the form of story events directly in the video is transformed into an arguably simpler task of searching within the plot synopsis, and using the alignment to present the corresponding video clip. Another application of

the alignment is *video summarization*. Here, an extractive text summarization technique (manual or automatic) can be first applied to the plot, automatically highlighting the most important segments of the video. These segments can be passed through standard video summarization methods, for further processing, but come with the added benefit of being grounded in the story and not an arbitrary feature (*e.g.* shot color). Finally, and especially with respect to the rise in popularity to train joint video/text embeddings and video captioning, plot synopses can act as a large and highly semantic data resource.

We will address the problem of story-based video retrieval in Sec. 4.6.2. StoryGraphs, presented in Chapter 5, also leverage plot synopses to obtain labels about key events in the episode. Plot synopses are also the data source which we use to gather questions and answers for the MovieQA data set presented in Chapter 6.

4.1.2 Books

The recent decade of television and film has seen a large number of books being adapted into film. This presents a very interesting opportunity to jointly analyze large natural language stories along with videos. There are numerous instances where film adaptations have encouraged consumption of both the literary and audio-visual material[1]. In fact, it has been noted that almost half of the highest grossing screenplays in the last 20 years are based on some form of literary adaptation[2].

Books are diametrically opposite to the summarizing nature of plot synopses. In sharp contrast to plots, a good book prods, develops and exposes its characters and their environments to severe detail. Books are very rich texts (in terms of vocabulary) conveying visual details about character appearance, interaction, their clothing, and surrounding scenes, while simultaneously maintaining a high-level story plot often presented through changing physical or mental states of characters. Similar to the case of plot synopses, an important first task to jointly localize parts of the book with segments of the video. Such visually grounded descriptions from the book can be used to *explain* and *understand* stories as compared merely to generating simple *captions* for visual content.

Aligning books with their video adaptations provides interesting opportunities for second-screen applications. While the video adaptation may not faithfully reproduce the details

[1] http://www.theatlantic.com/entertainment/archive/2014/09/kids-actually-read-the-books-that-movies-are-based-on/380395/
[2] https://stephenfollows.com/highest-grossing-movie-adaptations/

of the book, the core story is usually portrayed without much modification. As more and more platforms promote or sell books and videos (*e.g.* Google Play, Amazon), the alignment can enable a user to browse through the video clips while reading a book, or read well worded narrations or famous quotes from parts of the book corresponding to video scenes.

We present several examples of video captioning that can be achieved through the book-video alignment (Sec. 4.6.1) and also address the problem of finding differences between the book and its video adaptation at the scene-level (Sec. 4.6.3).

4.2 Similarity between video and text

Finding an alignment between two modalities – video segments with parts of text – requires bridging the gap between their data representations. While, there has been an explosion of work focused on learning joint visual and textual embeddings recently *e.g.* Frome et al. (2013); Karpathy and Fei-Fei (2015); Kiros et al. (2014); Zhu et al. (2015), our approach focuses on alignment by finding characteristics of the data that can be found in both modalities. In particular, and in contrast to most other methods, we leverage the notion that stories are strongly character-centric. Thus, detecting and resolving character names in the text, and identifying characters in the video are a crucial component to a good alignment. In fact, in preliminary experiments with recently proposed sentence embeddings (Kiros et al., 2015b), we observed reduced performance as character names are not well captured in such vector representations.

In this section, we define a joint similarity $\phi(t, v)$ between each video segment in the video, $v \in \mathcal{V}$, and chunks of text in the document, $t \in T$ computed at every unit of the alignment granularity. The alignment problem is thus transformed into an optimization problem that maximizes the similarity between the two modalities while respecting some story progression constraints.

For plot synopses, we propose a fine-grained approach where we seek to align individual shots of the video with each sentence from the plot synopsis. Dividing either of the modalities further is a fairly cumbersome task. Note that, the above can also be thought of as a "label assignment" problem, where the shots are individual data points and the plot synopsis sentence is the (quite long and descriptive) label. Indeed this will form the basis of our alignment evaluation metric.

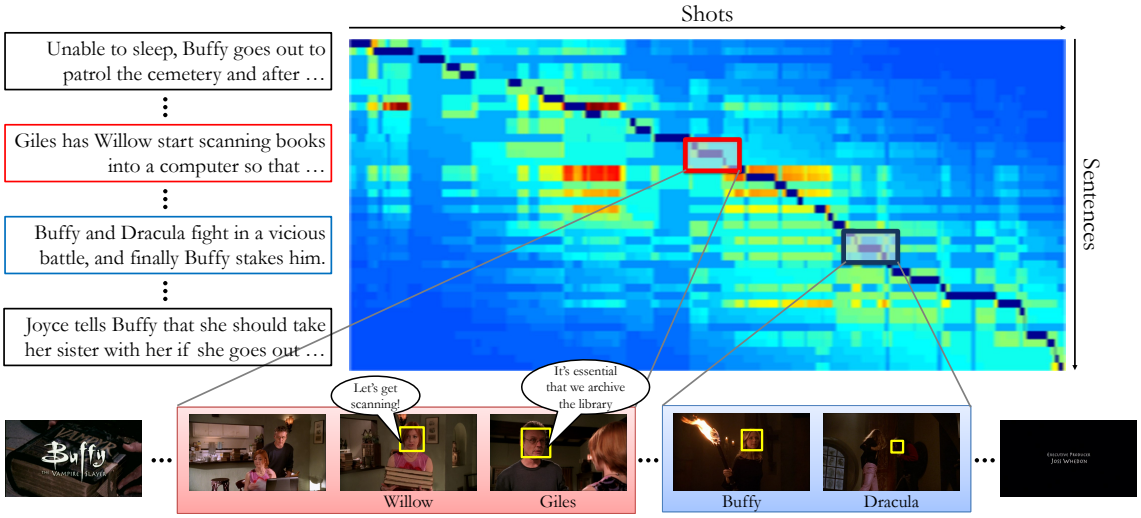


Figure 4.1: Each sentence of the plot synopsis is aligned to a set of shots from the video. We visualize the matrix obtained by evaluating the similarity function $\phi(t, v)$ at all points and overlay it with the ground-truth alignment (in dark blue). Each row of the similarity matrix corresponds to a plot synopsis sentence, some of which are presented to the left. Columns of the matrix are shots of the video displayed below. We highlight two shots-sentence pairs showing how a high similarity can arise from character identities and/or matching dialogs.

In case of books, the larger text corpus, we set our first goal to obtain a coarse grained alignment. We desire to find the video scenes which correspond to particular chapters. As noted before, video adaptations from books need not be completely faithfully and may stray a little from the original story. One of the common reasons for changes in a storyline is the larger screen time afforded to good actors, which requires their sub-story to be extended. At the level of our granularity, we are able to predict whether a video scene was part of the book.

Note that for both data sources, the alignment is a many-to-one problem, *i.e.* many video segments may be associated with a chunk of text, but not vice-versa. This will be an important factor while considering alignment algorithms. Below, we define the similarity functions that are applicable for the two text sources. While presented separately for simplicity, they share common ideas spanning *character identities* and *dialog matching*.

4.2.1 Plots: Shot-sentence similarity

We first present the similarity function used for plot synopsis alignment. The plot is a document consisting of N_T individual sentences $\mathcal{T} = \{t_1, \ldots, t_i, \ldots, t_{N_T}\}$. Similarly, the video

is also represented as a set of N_V shots $\mathcal{V} = \{v_1, \ldots, v_j, \ldots, v_{N_V}\}$. Aligning video shots with a plot sentence t_i can be restated as finding sets of shots $\mathcal{V}_i^1 = \{v_{i1}^1, \ldots v_{ik}^1\}$, $\mathcal{V}_i^2 = \{v_{i1}^2, \ldots v_{il}^2\}$, ... that best depict it.

To find the best match, we formulate a similarity function $\phi(t_i, v_j)$ composed of character identity and dialog matching cues, that measures the likelihood that sentence t_i may describe shot v_j. Fig. 4.1 presents an overview of the approach used to compute this similarity function, along with a glimpse of the ground truth alignment.

Character identities. Characters and their interactions play a crucial role in any story. For the alignment, a reference to a character indicates a high likelihood of seeing him/her in the video shots. Note that as the plot is summarizing in nature, the reverse need not hold true, *i.e.* characters may appear in the video and need not always be listed in the plot.

We obtain the list of characters in every episode or movie via IMDb or Wikipedia. Finding exact word matches for the names (typically first names) in the plot document reveals information cues for sentences. In addition, we resolve pronouns and character references (*e.g.* sister, father) using co-reference resolution (Lee et al., 2011). The automatic co-reference resolution is augmented by a simple, yet surprisingly effective technique that looks back in text to find the closest antecedent that agrees in gender. For example,

> *Buffy awakens to find Dracula in her bedroom. She is helpless against his powers and is unable to stop him from biting her.*

establishes the two character interaction in the first sentence. In the subsequent sentence, we can easily resolve "She" and "her" to "Buffy", and "his" and "him" to "Dracula". An additional trick is to ignore names that appear before a preposition. For example, in the sentence Riley asks Spike about Dracula ..., it is reasonable to assume that "Dracula" is not actually visible in the scene. We create for each sentence t_i the list of characters C_i^t that are mentioned in it.

Simultaneously, we detect, track and identify characters using the methods proposed in Chapter 3. Similar to the text domain, we obtain for each shot v_j, the set of characters C_j^v that appear in it. Standard video editing practices in TV series and movies affect the number of characters that are visible at a given time. For example, during a conversation between two characters, the popular "over-the-shoulder" shot only shows the face of the

speaker. In such a case, it is misleading to assume that both characters are not present for the alignment. We reason that most plot synopses sentences correspond to a span of several tens of shots, and thus spread the appearance of a character in shot v_j to a small neighborhood r around it.

Characters in a TV series appear with varying amounts of screen time. We introduce *inverse character frequency*, a measure inspired by the Inverse Document Frequency in text processing (Manning et al., 2008). Characters that appear in a short localized period are very useful to find "hot spots" as their presence in a small part of the video and plot indicate that those parts are very likely to be associated. We compute the inverse character frequency for each character c in the cast list C as

$$w_c = \frac{\log \max_{z \in C} N_z^f}{\log(N_c^f + 1)}, \tag{4.1}$$

where N_c^f is the number of face tracks tagged as character c.

The identity-based similarity function for sentence t_i and shot v_j is

$$\phi^{\mathrm{id}}(t_i, v_j) = \frac{1}{2r+1} \sum_{k=j-r}^{j+r} \gamma(C_i^t, C_k^v), \tag{4.2}$$

where $\gamma(\cdot,\cdot)$ computes a character-weighted intersection between the two character lists

$$\gamma(C_i^t, C_j^v) = \sum_{m \in C_i^t} \sum_{n \in C_j^v} \mathbb{1}(m = n) \cdot w_n . \tag{4.3}$$

Keywords. Our second similarity cue stems from text matching. While plot synopses (unlike transcripts) rarely refer to dialogs verbatim, there are sparse keywords that may appear in the conversation and the plot. We first bin the dialogs (through the use of subtitles) into shots, and perform chunking to get a list of uttered words in every shot W_j^v. A similar procedure provides the list of words in every sentence of the plot W_i^t. We perform stop-word removal (Loper et al., 2009) and additionally obtain the root words by using the Porter Stemmer (Porter, 1980).

The dialog based similarity function depends on the number of matching words in the two lists

$$\phi^{\mathrm{dlg}}(t_i, v_j) = |W_i^t \cap W_j^v| . \tag{4.4}$$

A weighted combination of the identity- (Eq. 4.2) and dialog- (Eq. 4.4) based similarity functions is used to find the best alignment. We show in the evaluation that both cues provide complementary information that helps improve alignment performance. Note that the similarity can be pre-computed for every possible combination to yield a matrix $\phi \in \mathbb{R}^{N_T \times N_V}$ of similarity scores. Fig. 4.1 displays an example of such a similarity matrix across shots and sentences.

Other vision cues. Early in this work, we made attempts to use *object*, *place/scene* and *action* recognition in addition to character identities to inform the similarity function. While place recognition showed most promise and usable vision performance, places are mentioned quite sparsely (typically 3-5 instances in 30 sentences) and their mentions in the plot need to be interpreted (*e.g.* *apartment*, *home* resolve to *indoor*). An additional and significant problem is the difference between the two domains. Classically, images used to train place classifiers (*e.g.* Places 205 (Zhou et al., 2014)) are almost entirely devoid of people, and on the other hand, finding video shots without people is quite challenging. Action recognition in open-ended video clips is a hard problem, which is made more challenging as plot synopses typically refer to complex composite actions (*e.g.* scan books, patrol, magically ignite) for which it is hard to obtain training data. While object recognition works pretty well as image classification, localizing small instances (in the frame) of varied object classes (*e.g.* books, computer, stake) in video (in time) is a hard challenge. Even with recent advances in object detection using Region-CNNs (Girshick et al., 2014), the number of correct matches is far lower than the number of false positives, making it unusable for alignment.

4.2.2 Books: Scene-chapter similarity

Similar to the alignment between plot synopses and videos, we now present the similarity functions for aligning books with videos. For notational simplicity, we keep the same representations. Note that the models proposed in the next section can be applied to align both plot synopses and books with videos. Most notably, we use different granularity for the alignment, and the text unit element t_i now represents book chapters, N_T is the number of chapters, v_j is a video scene, and N_V is the total number of video scenes. The similarity function constitutes cues from character identities and matching dialogs between every book chapter and video scene $\phi(t_i, v_j)$. Fig. 4.2 presents an example alignment between chapter 2 and scene 15 of the *Game of Thrones* book/series.

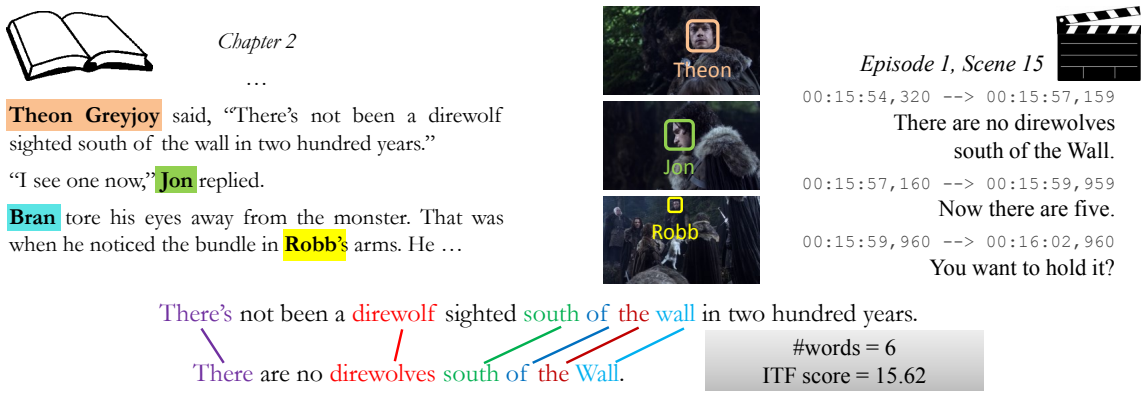


Figure 4.2: We show the alignment cues used to bridge the gap between books and their video adaptations. Our first cue is character identities: names appearing in the book are matched with characters that appear in the video. The second cue is obtained from dialogs between books and video, and while they may undergo some changes, we often see that the keywords remain same. We find the longest common subsequence between dialogs and score them based on the number and rarity of words.

Character identities. Character identities from videos are obtained in a similar fashion to the ones used for plot synopses. We detect, track and identify faces automatically using the methods discussed earlier. Notably, book adaptations are usually movies which exhibit a higher production quality creating difficulties in obtaining automatic weak labels following Everingham et al. (2006). In contrast to traditional TV or sitcoms, movie filming styles often use wider filming angles and the speaking character tends to appear less frequently in the video. To address this problem, and to improve the quality of weak labels, we proposed a joint weak label gathering scheme which incorporates improved speaking score measures and pairwise constraints (Tapaswi et al., 2015b). Details and a short evaluation on this topic were presented in Sec. 3.5.2.

Names in plots can be detected by trivial full word matching techniques. However, books (especially those in the fantasy genre) often address characters by multiple names. A peculiar example is the character *Eddard Stark*, who may be addressed by his first name *Eddard*, family name *Stark*, title *Lord Stark*, or even an alias or nickname *Ned*. Addressing based on family name causes confusion between several characters from the same family, while titles are quite tricky as they may be passed on (*e.g.* from father to son). We thus weight the importance of name references appearing in the book (ordered from high to low) as follows: (i) full name; (ii) only first name; (iii) alias or title; (iv) only last name.

For every scene v_j and chapter t_i we count the number of occurrences or mentions of each character, and accumulate them as "histograms", a vector in the dimension of the

number of characters N_C. We weight all named mentions in the book chapters to obtain $\mathbf{C}^t \in \mathbb{R}^{N_T \times N_C}$ and count the number of face tracks for each character in the video scenes $\mathbf{C}^v \in \mathbb{R}^{N_V \times N_C}$. We further normalize these matrices such that each row (the occurrence of characters in one scene or chapter) has a unit norm. The character identity based similarity function is defined as

$$\phi^{\text{id}}(t_i, v_j) = \sqrt{2} - ||\mathbf{c}_i^t - \mathbf{c}_j^v||^2, \tag{4.5}$$

where $\mathbf{c}_i^t$ corresponds to the i^{th} row of the name occurrences in the book $\mathbf{C}^t$, and $\mathbf{c}_j^v$, the j^{th} row in the video $\mathbf{C}^v$.

Dialogs. Matching dialogs between books and their film adaptation are very strong anchors for aligning them. This is in sharp contrast with plot synopses, where dialogs appear only very rarely, and keyword matches are more prominent. However, the matching process needs to account for dialogs which are adapted to make them more suitable for the screen.

For example, in one of our data sources (*Game of Thrones*), we have 12,992 dialogs in the novel and 6,842 dialogs in the video. Even with such a large number of dialogs, we are only able to find a perfect match between 308 pairs with 5 or more words. Note that small dialogs (*e.g.* "Your Grace") that match too frequently are are not useful for aligning the two modalities. In contrast to the naïve matching scheme, our proposed technique based on leveraging the Term Frequency (Manning et al., 2008) and the Longest Common Subsequence (Chvátal and Sankoff, 1975) finds over 1,358 pairs with a high confidence.

Let $\mathscr{D}^t$ and $\mathscr{D}^v$ be the set of dialogs in the book and video respectively. We extract the longest common subsequence between every pair of dialogs (d_i^t, d_j^v) and score the pair using the set of matching words W_{ij}^{tv}

$$\psi(d_i^t, d_j^v) = -\sum_{w \in W_{ij}^{tv}} \frac{1}{2} \left(\log n^t(w) + \log n^v(w)\right), \tag{4.6}$$

where $n^t(w)$ and $n^v(w)$ are term frequencies (fraction of the number of occurrences of w to the total number of words) for the word w in the text $\mathscr{D}^t$ and video $\mathscr{D}^v$ dialogs respectively. Incorporating inverse term frequency (through negative log) has the effect of automatically diminishing the influence of matching stop-words (here, words that occur frequently in the dialogs), while emphasizing rare words in the dialog pair.

We associate the book dialogs with its chapters and split video dialogs into scenes. The similarity function between a book chapter t_i and video scene v_j is the accumulation of all pair-wise dialog matching scores within them

$$\phi^{\text{dlg}}(t_i, v_j) = \sum_{d_i^t} \sum_{d_j^v} \psi(d_i^t, d_j^v). \tag{4.7}$$

As with plot synopses, the alignment models use a weighted combination of the identity- (Eq. 4.5) and dialog- (Eq. 4.7) based similarity functions. In contrast to plot synopsis alignment, we will see that the matching dialogs have a significant influence in the book-to-video alignment performance.

4.3 Alignment models

Throughout this section on alignment models, we may refer to only one of the two alignment problems (shots with plot sentences, or scenes with book chapters), however, note that they can be used interchangeably. We propose several models which compute an alignment between video shots (scenes) and text sentences (chapters). Depending on the kind of information used to perform the alignment, the final objective, and imposed constraints, the different approaches can be classified into 4 major types:

1. Structure: We propose two alignment methods which use the structure and common sense reasoning about the video and text source. These are treated as priors and do not use the text-video similarity functions computed earlier.

2. Maximize similarity: This method aims to maximize similarity and treats neighboring data points (*e.g.* shots) independent from one another. In essence, this hasty method does not care about imposing any story constraints.

3. Constrained alignment: As both modalities – text and video – come from the same story, we can make an assumption about their structure. In particular, we argue in favor of temporal ordering, *i.e.* if shot v_j is aligned with sentence t_i, shot v_{j+1} is likely to be aligned with the same sentence t_i or the very next sentence t_{i+1}.

4. Freedom of movement: We soften the strict restrictions imposed in the previous "constrained alignment" approach, and propose a method that requires evidence to align a video segment with a chunk of text. Without sufficient evidence, this

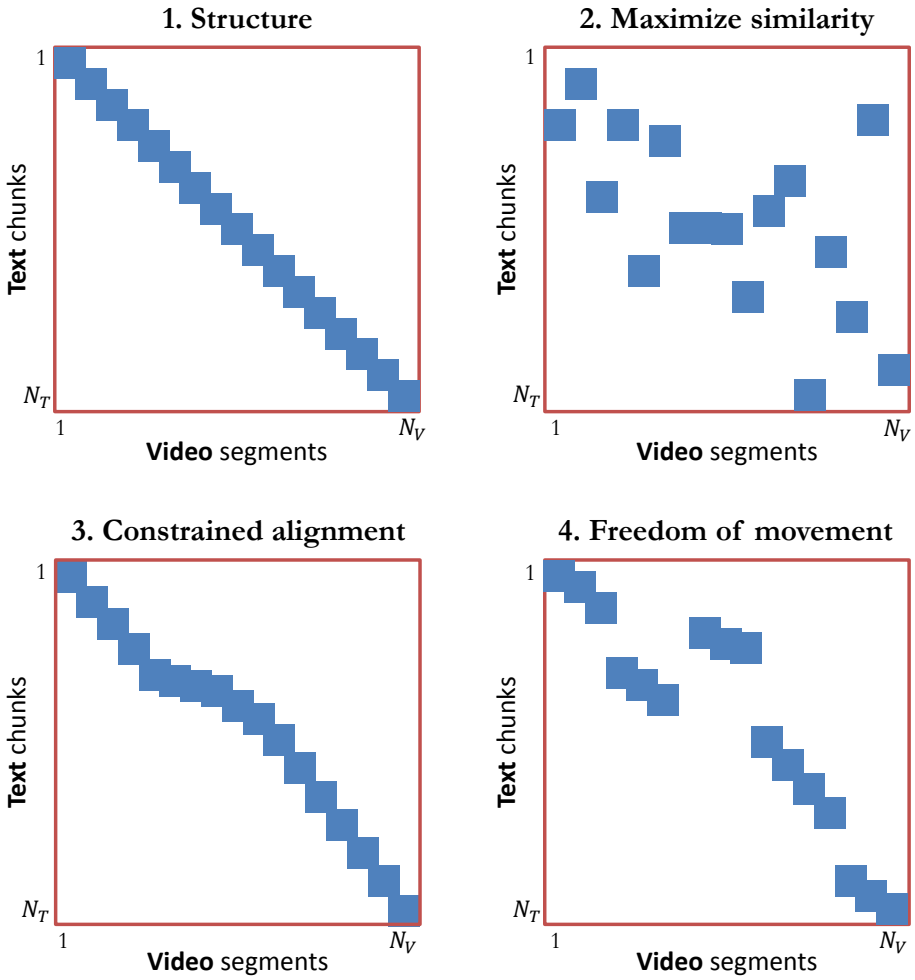


Figure 4.3: The four different types of alignment models. For each model type, we plot the chunks of text as rows, and video segments as columns. The blue boxes highlight the set of video segments and text chunks that are aligned. Note for now how different methods result in different characteristic shapes. We will present alignments and discuss their implications on real data in a similar fashion.

method prefers to deem that the video segment was not discussed (not part of the plot synopsis sentences, or not part of the book chapters) in the text.

Fig. 4.3 provides a big picture overview of the four types of alignment methods. Recall that as all the dialogs in the video do not appear in both sources, plots and books, we are unable to compute an alignment between the plots (or books) and video through subtitles as was done in the case of transcripts (Everingham et al., 2006; Laptev et al., 2008).

4.3.1 Structural alignment

Story-telling is strongly related with a timeline of events and even though different modalities (text vs. video) may differ slightly, the principal chain of events stays true. We propose two alignment techniques based on simple structural cues (*e.g.* depending on the number of sentences or shots) that explore the hidden biases in story telling.

Diagonal alignment (DIAG). As the number of video units (shots or scenes) is higher than their text counterparts (plot sentences or book chapters), we explore assignment of multiple video units to each text unit. A simple approach is to assign equal number of video segments to each text chunk. Given N_T text and N_V video segments, shot v_j is aligned with sentence t_i as

$$i = \lceil j \cdot N_T / N_V \rceil. \tag{4.8}$$

When the alignment is plotted (see Fig. 4.3(1)), this results in a "diagonal" (equal ratio) assignment of video segments to text chunks.

Diagonal alignment is motivated by the reasoning that former (latter) parts of the video are very likely to be described earlier (later) in the text. This principal of diagonal alignment is quite effective, and in fact it is beneficial to incorporate it in other methods that use the similarity function. To include the impact of such a "diagonal prior", we update the similarity function through a Gaussian prior, that emphasizes the bulk of similarity along the main diagonal

$$\phi_g(t_i, v_j) = \phi(t_i, v_j) \cdot \exp\left(-\frac{(j-\mu_i)^2}{2\sigma^2}\right), \tag{4.9}$$

where $\mu_i = (i - 0.5) \cdot N_V / N_T$ is the center of the Gaussian weight update applied to every row (text unit) of the similarity matrix. We empirically set $\sigma = N_V / N_T$. Fig. 4.1 shows the similarity matrix after the diagonal prior has been applied. Note how the similarity on the off-diagonal elements (bottom-left and top-right corners) is reduced, while the values on the leading diagonal are left untouched.

Bow-shaped alignment (BOW). While the diagonal prior performs surprisingly well for aligning plot synopses with videos of some episodes (see Sec. 4.5.2), we observe that climactic story-telling influences the alignment pattern especially in the case of plot synopses. Authors of plots tend to dedicate more space (sentences) to describe the

climactic parts of the story that appear towards the end of the video, while speeding through the introductory segments. This nudges the alignment pattern into the shape of an arced bow off the leading diagonal.

We propose to model the shape of this bow using a *rational quadratic Bézier curve* (Rogers and Adams, 1990) that is used to generate parametric conic sections given a fixed start and end control point. The third point of the quadratic curve is parameterized by $\gamma \in [0,1]$, and is obtained as

$$P_{\text{start}} = [1,1], \tag{4.10}$$

$$P_{\text{end}} = [N_V, N_T], \tag{4.11}$$

$$P_{\text{bow}} = [\gamma N_V, (1-\gamma) N_T]. \tag{4.12}$$

While we present the above equations using 1-based indexing, we use 0-based indexing to plot the Bézier curve. Note that the diagonal alignment presented above is a special case of the bow-shaped prior which is a straight line when $\gamma = 0.5$. To model the alignment while considering climax, we wish to nudge the diagonal in the top-right direction. This is achieved by using $\gamma > 0.5$.

4.3.2 Maximize similarity (MAX)

Given the similarity between text and video units, a straightforward approach to perform alignment is to align each video segment with the text chunk that maximizes similarity. For example, shot v_j is aligned with sentence t_i as

$$i = \underset{r}{\text{argmax}}\, \phi_g(t_r, v_j). \tag{4.13}$$

While this method guarantees maximum similarity over the entire alignment, the burden of good performance is dumped onto the quality of the similarity functions, and knowledge about the structure of stories is left out. The method results in a "broken" alignment where even neighboring video segments may not be assigned to the same text unit. This is a highly unlikely scenario, especially in the case of plot synopsis alignment, where our video units, shots, are often quite small (1-2 seconds). Fig. 4.3(2) presents the expected results from such an alignment. Nevertheless, the use of the diagonal constrained similarity $\phi_g(t_i, v_j)$ in place of the original $\phi(t_i, v_j)$ attempts to coerce the alignment to the leading diagonal.

4.3.3 Constrained alignments

As in the previous methods, we restrict our alignments to assign multiple video segments to one text chunk. That is, we employ a many-to-one mapping scheme, and do *not* allow one-to-many (one shot to many sentences) or many-to-many (many shots to many sentences) schemes. While this is a fairly strict restriction, we observe that owing to the choice of our text and video unit sizes, both plot synopsis and book alignment satisfy it. In the data set statistics we will see that the number of video segments is much larger than the text chunks. Nevertheless, restricting the potential alignments to a many-to-one (many shots to one sentence) scheme massively reduces alignment complexity allowing the use of efficient dynamic programming approaches.

One clear advantage of the MAX alignment method is that it maximizes the global similarity measure. However, it treats video segments independent of one another, which is especially harmful in the case of a linear storyline where neighboring shots are strongly related with each other and are often described by the same plot synopsis sentence.

In this section, we propose to include temporal constraints in the alignment technique. We allow to assign a video segment v_{j+1} only to the same text unit t_i as v_j or the subsequent chunk t_{i+1}. If t_i has been aligned with v_j, a valid alignment results only when

$$\text{if } t_i \Longleftrightarrow v_j; \quad \text{then } t_m \Longleftrightarrow v_{j+1} : i \leq m \leq (i+1). \tag{4.14}$$

To search for the best alignment, we are able to use a modified form of the Dynamic Time Warping (DTW) (Myers and Rabiner, 1981) algorithm used commonly for speech recognition. Similar to structural alignment approaches, this results in an "continuous" and "unbroken" alignment, warped to pass through regions of high similarity (see Fig. 4.3(3)).

Constrained alignment (DTW2). The temporal constraint presented in Eq. 4.14 is easily captured via the DTW algorithm. In the first phase of the DTW, we compute the similarity score at all possible sets of alignments. We construct a matrix D using on the similarity function $\phi_g()$ with two possible movement rules

$$D(i,j) = \max \begin{cases} D(i,j-1) + \phi_g(t_i, v_j) \\ D(i-1,j-1) + \phi_g(t_i, v_j). \end{cases} \tag{4.15}$$

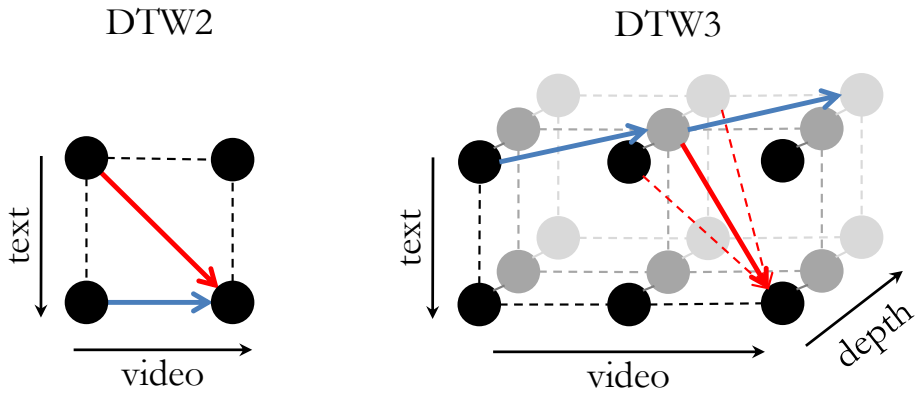


Figure 4.4: Valid transition paths that are supported in the constrained alignment schemes. Each circle is a node in the forward-computation matrix D. A shot being assigned to the same sentence is indicated by the blue/light arrow, while assigning the shot to a new sentence is achieved through the red/dark arrow. LEFT: DTW2 follows simple transition rules respecting the temporal constraints of Eq. 4.14. RIGHT: The depth in DTW3 is indicated by lighter circles behind the original layer 1 grid. Note how the blue transition not only moves right, but also one layer back (depth) allowing us to keep count of the number of shots aligned with the sentence.

At each step, we also store the chosen path in a separate matrix P which is used to perform backtracking. The first transition assigns v_j to the same text unit as shot v_{j-1}, while the second assigns v_j to the next text unit. Note that we do not consider the third rule used in the original DTW algorithm $D(i-1,j)+\phi_g(t_i,v_j)$ as it assigns unit v_j to two text units violating our many-to-one policy. The valid transitions are visualized in Fig. 4.4(left).

The second phase of DTW involves backtracking to find the best path through the matrix D. Backtracking is initiated from the last row and column of the matrix $D(N_T,N_V)$. The nodes visited during backtracking is the optimum alignment that maximizes similarity while respecting temporal constraints. The computational complexity of this algorithm is in $\mathcal{O}(N_T N_V)$.

We call our slight modification of the original DTW as DTW2 to indicate the use of a two-dimensional forward computation matrix (in contrast to DTW3 we will discuss next). Note that the DTW2 is similar to the method proposed in Sankar et al. (2009). However, it is used to align transcripts with parts of the video, an arguably easier problem given a good automatic speech recognizer that can be used to map dialogs.

Constrained regularized alignment (DTW3). One of the problems that DTW2 introduces is evident when a text unit scores high with a large number of video shots. For

example, this happens during plot synopsis alignment when the sentence mentions many names and thus generates high similarity scores with all shots. The alignment resulting from DTW2 prefers to align most video segments to this text unit. We believe that this is a highly skewed case and introduce the concept of regularization to DTW, and propose an automatic scheme to control the number of shots that are assigned to one sentence.

We introduce a quadratic decay factor α that reduces the impact of additional high similarity nodes that may be aligned with a "heavy" text chunk that is already aligned with many video segments. α depends on the number of video segments k that are already assigned to the text unit.

$$\alpha(k) = 1 - \left(\frac{k-1}{z}\right)^2, \quad k = 1, \ldots, z, \tag{4.16}$$

where z is theoretically upper bounded by N_V, but in practice is sufficient to limit to $z = 5N_V/N_T$. As we will see, setting $z < N_V$ improves computation speed without sacrificing performance.

The above decay factor is easily incorporated in the dynamic programming framework thus yielding an efficient solution. We first extend the forward computation matrix D by a third dimension $k = 1, \ldots, z$. The transition that assigns the shot to the same sentence now not only moves right, but increments the depth counter while doing so.

$$D(i, j, k) = D(i, j-1, k-1) + \alpha(k)\phi_g(t_i, v_j), \quad \text{when } k > 1. \tag{4.17}$$

Note that assigning more shots to the same sentence reduces the impact of the similarity through the reducing decay factor $\alpha(k)$.

Assigning v_j to a new sentence resets the depth counter to 1. This is facilitated by enabling transitions from all possible depths back to the first layer.

$$D(i, j, 1) = \max_{k=1,\ldots,z} D(i-1, j-1, k) + \alpha(1)\phi_g(t_i, v_j). \tag{4.18}$$

Similar to DTW2, we compute the three-dimensional forward computation matrix D while storing the chosen path to arrive at each node. However, backtracking is not initiated at $D(N_T, N_V, 1)$, as the last text unit is aligned with an unknown number video segments. We start backtracking at the node with maximum score across the depth layers $\max_k D(N_T, N_V, k)$. The row and column coordinates of the path provide us with the

constrained regularized alignment. The computational complexity of this method is $\mathcal{O}(N_T N_V z)$ with the worst case being $\mathcal{O}(N_T N_V^2)$ when depth z spans the entire video.

We name this approach DTW3 to indicate the use of a three-dimensional matrix. Fig. 4.4(right) illustrates the valid transitions in the DTW3 approach.

Constrained alignment as scene detection. Earlier in Sec. 3.4.3 our proposed scene detection algorithm used a dynamic programming scheme to detect shots. DTW3 is easily modified for the task of "aligning" shots with scenes, or rather detecting scene boundaries while maximizing within-scene similarity through color consistency and shot threading.

4.3.4 Allowing freedom of movement (SHORT)

MAX and DTW2/DTW3 are conflicting strategies, in which they either completely disconnect the video segments (MAX) or impose severe restrictions and connectivity among them (DTW3). We observe quite often that an intermediate path would be an ideal solution. While we wish to have strong connectivity among close neighbors, allowing freedom to swap text and video segments is a desirable property. This is especially true when the TV series or movie consists of multiple storylines that take place in parallel.

Another critical drawback of the methods proposed so far is the fact that *every* video segment is aligned with some text chunk. However, this is not necessarily true, as plot synopses summarize the story of the video and often leave out unimportant parts of the video. In the case of aligning books, a different problem appears where the adaptation may not be translated faithfully. Here, new scenes are introduced which are not part of the book. Hence, it is desirable for alignment schemes to consider "skipping" text or video segments. For example, Fig. 4.3(4) presents an alignment that satisfies the above properties.

In this section, we attempt to merge the desirable properties of constrained alignment and maximizing similarity. This is achieved by modeling the alignment problem as finding the shortest path through a sparse Directed Acyclic Graph (DAG). Similar to DTW approaches, the DAG is constructed using a grid of nodes corresponding to each pair of text and video units. The edge distances of the graph are devised such that the nodes visited while taking the shortest path through the graph correspond to the best matching alignment. This approach – named SHORT – incorporates information from similarity

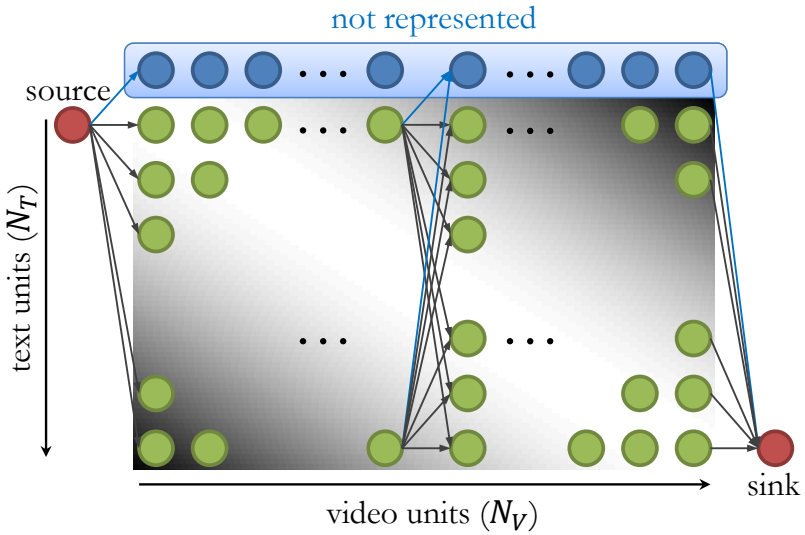


Figure 4.5: The structure of the graph used in the SHORT alignment approach. Each green node represents a pair of text and video units (t_i, v_j). Blue nodes are used as a proxy to determine when the video unit is not represented in the text. Edges connect every consecutive pair of columns allowing freedom to assign video segments to any text chunk.

cues with a structure that can align non-sequential storylines, and at the same time is very efficient to solve.

Graph construction. Fig. 4.5 illustrates the structure of the graph along with its edges. The main graph consists of $N_T \times N_V$ nodes ordered as a regular matrix (green nodes in the figure), with text chunks associated with rows and video segments with columns. For simplicity, we will index rows with i and columns with j, and each node represents the pair (t_i, v_j). We create a dense connection of edges between every consecutive column of nodes, *i.e.* every node in column j has an edge to reach every node in column $j+1$ resulting in N_T^2 edges. Note that these edges respect the many-to-one alignment scheme (no edges between nodes of the same column), and provide the freedom to align each video segment with any text unit.

In the constrained alignment approach, we assume that the first (last) shot is aligned with the first (last) sentence. However, this need not be the case, and we circumvent this by introducing additional source and sink nodes (red nodes in the figure). The source has edges to all nodes in column 1, and all nodes of column N_V are connected to the sink.

The DAG, as it is so far, forces each video segment to be associated with some text chunk. As motivated earlier, this is often not true and we bypass this problem by adding a set

of N_V additional nodes (blue nodes in the figure) which represent aligning the video segment v_j with $\varnothing$ corresponding to no text chunk. As in the main grid, these nodes have incoming edges from all nodes of the previous column. Fig. 4.5 presents a glimpse of the graph structure.

Edge initialization and diagonal prior. As the alignment problem is transformed into finding the shortest path through the graph, we first initialize the edges with an initial offset distance, and reduce it based on information supplied by similarity cues.

As in the constrained alignment, we wish to (softly) encourage neighboring shots to be assigned to same or nearby sentences. The *local* distance from any node (t_i, v_j) to a node in the next column (t_r, v_{j+1}) is modeled by a quadratic distance depending on how far r is from i:

$$d_{(i,j)\to(r,j+1)} = \alpha + \frac{|r-i|^2}{2 \cdot N_T}. \tag{4.19}$$

For ease of notation, we address node (t_i, v_j) as (i, j) and $d_{n1\to n2}$ is the distance to go from node $n1$ to node $n2$. α serves as a positive distance offset from which we will subtract the similarity scores.

While the above local edge distance encourages assigning shots to the same sentence, we add an additional multiplicative Gaussian factor to capture the *global* likelihood of being at any node in the graph. This is similar to the relation between $\phi_g(\cdot,\cdot)$ and $\phi(\cdot,\cdot)$ (*cf*. Eq. 4.9). All incoming edges to node (i, j) are multiplied by a factor

$$g(i,j) = 2 - \exp\left(-\frac{(j-\mu_i)^2}{2 \cdot N_V^2}\right), \tag{4.20}$$

where $\mu_i = (i - 0.5) \cdot N_V / N_T$ and $2 - \exp(\cdot)$ ensures that the multiplicative factor $g \in [1, 2]$. This factor increases distances to visit off-diagonal nodes (top-right or bottom-left corners) thus preferring an alignment hugging the main diagonal.

Reducing edge distances with similarity cues. After initializing the edge distances (Eq. 4.19) and applying the Gaussian multipliers (Eq. 4.20) the shortest path corresponds to assignment of equal number of video segments to each text chunk (similar to DIAG).

We now modify incoming distances to node (i,j) from all nodes n in the previous column ($n = \{(r, j-1)\} \forall r = 1, \ldots, N_T$):

$$d_{n\to(i,j)} = d_{n\to(i,j)} - \phi_n(t_i, v_j), \tag{4.21}$$

where $\phi_n(t_i, v_j)$ is a normalized similarity such that $\sum_i \phi_n(t_i, v_j) = 1$. When the text chunk t_i and video segment v_j exhibit a high similarity, the reduction in the edge distances to visit node (i,j) encourages the shortest path to "funnel through" the node. Most importantly this takes place while respecting other aspects of alignment such as staying close to the main diagonal and assigning the current shot to a sentence close to the previous shot.

The above edge distance reductions are applied to the edges of the main grid. Nodes corresponding to shots that are not referenced in the text are denoted by $(\varnothing, j)$ and their distances are modified as a function of the accumulated similarity of the entire column:

$$d_{n\to(\varnothing,j)} = d_{n\to(\varnothing,j)} - \max(0, 1 - \sum_i \phi_n(t_i, v_j)). \tag{4.22}$$

Thus, when a video segment v_j shows little to no similarity with any of the text chunks, the distance to visit the null node $(\varnothing, j)$ is reduced. Visiting the node $(\varnothing, j)$ represents the special case of aligning the video segment with no text object, or determining that the video segment is not described by any text chunk.

4.4 Related work on aligning videos with text

We compare the proposed similarity measures and alignment approaches against key related work on aligning videos with text. A broader overview was presented earlier in Sec. 2.2.

- Everingham et al. (2006); Laptev et al. (2008) align videos with transcripts through the use of subtitles. This is achieved using a model similar to DTW2, however the similarity function is trivial as matching words in the dialogs from subtitles and transcripts suffices. Note that both our forms of text – plot and books – do not contain all the dialogs from the video, making the alignment problem considerably harder. Sankar et al. (2009) align videos with transcripts when no subtitles are found. The model used here is similar to DTW2.

- A short period after our work, Zhu et al. (2015) propose to align video shots with sentences or paragraphs from the book. They propose several joint text-video similarity measures including a novel embedding for video shots and book sentences. However, the approach is from the standpoint of pure vision-language analysis and characters that form a key part of the story are not considered. To provide freedom of movement (as in SHORT), Zhu et al. (2015) model the alignment problem as inference in a Conditional Random Field where pair-wise constraints (similar to DIAG or *local* distance in SHORT) are considered.
- Another closely related work appearing after our alignment with plots and books is Bojanowski et al. (2015). Here, the authors propose to align cooking videos with their natural language descriptions. They model the alignment as a integer quadratic program, and use constraints on the obtained alignment path (similar to our Gaussian prior) and the size of video segment durations (similar to our regularized DTW3). Vision and language similarity is bridged through the use of bag-of-words representation and analysis of the grammatical structure on the text side, and classical vision features for action recognition – improved dense trajectories (Wang and Schmid, 2013).

The recent years have seen growing interest in joint vision and language analysis, and we believe that high-level story overviews (plot synopses) and detailed plot descriptions (books) will play an important role in understanding stories.

4.5 Evaluation

We evaluate the alignment of videos with plot synopses and books separately. To the best of our knowledge, this is the first work in the area of aligning complex videos with stories in the text-form. We thus propose new data sets, and also explain simple metrics that can be used to quantify the alignment performance. Note that the general properties of TV series used in this thesis were explained earlier in Sec. 3.1.

Data sources for plot synopsis alignment. We focus on two diverse TV series to analyze how well our models perform to align plot synopsis with videos. As a representative for standard drama series, we use the fifth season (22 episodes, ∼15 hours) of *Buffy the Vampire Slayer* (BF), which consists of a relatively central storyline revolving around one character (*Buffy*). The other data source is the first season (10 episodes, ∼9 hours) of

Game of Thrones (GOT), a fantasy series that portrays several stories that take place in parallel across a large continent. Note that these two data sets are severely different in the filming styles and the narrative structures. One of the primary challenges in GOT, is the segregation of plot synopses into paragraphs for each sub-story that appears in the video. This is in contrast to the video itself, where the scenes of a particular sub-story appear several times and not just during one period.

Data sets for book alignment. Books are commonly adapted to films, but there are also some recent examples of TV series. To analyze the performance of alignment between videos and books, we address both films and TV series. TV series episodes cannot be treated like a really long movie, as the episodes will typically end with a cliff-hanger thus creating a need for introduction of additional scenes, or re-ordering the content. We use the first book of the *A Song of Ice and Fire* series which corresponds to the first season of the TV series *Game of Thrones* (GOT). Our second data source is the first book and film *Harry Potter and the Sorcerer's Stone* (HP1). Both data sources HP1 and GOT exhibit large amounts of diversity in the amount of video and text material (see Table 4.3), the type of stories (single thread vs. multiple sub-stories) and even the target audience (kids vs. adults).

Alignment evaluation metric. We adopt a simple metric *accuracy* to quantify the alignment performance. For every video segment v_j, let the text unit t_j^* be the ground truth alignment, and text unit t_j the predicted. The accuracy is defined as

$$\text{accuracy} = \frac{\text{\# of video segments aligned with the correct text chunk}}{\text{total \# of video segments}}, \tag{4.23}$$

$$= \frac{1}{N_V} \sum_{j=1}^{N_V} \mathbb{1}(t_j^* == t_j). \tag{4.24}$$

Note that our measure follows the many-to-one paradigm and is applied to the modality with higher number of elements (video shots or scenes) to better evaluate the alignment performance of various models. Unlike the popular segmentation metric intersection-over-union, accuracy provides information that is directly related to the use cases of the alignment (*e.g.* retrieval, describing).

Accuracy also works with video segments that are not represented in the text $\varnothing$, however does not provide special insights about how well we perform at finding them. We model

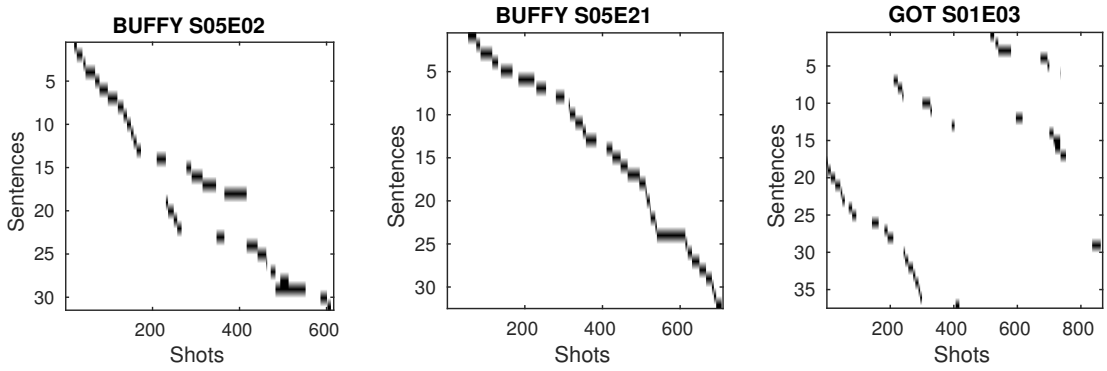


Figure 4.6: Ground truth alignments for plot synopsis of different episodes of our data sources (SXXEYY represent season and episode number). We show shots on the x-axis and sentences on the y-axis. The black path in the chart represents the shots in that segment that are aligned with the corresponding sentence. Note how the episodes of BF are closer to the diagonal as they represent a single storyline, while the episode of GOT shows an extreme case where multiple stories are intermixed in the video, however are separated into paragraphs in the plot.

finding such video segments as a "detection" task and evaluate it using standard precision and recall.

4.5.1 Ground truth alignments

The alignment of videos with plot synopses or books is a fairly subjective task, and different people may have slightly differing opinions about the exact boundaries of the alignment. While we treat alignment as an objective problem for the rest of the chapter, we provide some insights about differences in alignments.

We obtain one set of ground truth alignments for all episodes or movies in our data sources, BF and GOT for plots, and GOT and HP1 for books. The plot synopsis ground truth alignment includes the start and end timestamp for each sentence of the plot. For books, we also collect the start and end timestamp of the video segments that belong to the same chapter. The slight difference in this approach to data collection renders the video segment corresponding to a plot synopsis sentence to be contiguous, while chapters may be aligned with multiple video segments.

Fig. 4.6(a, b) show examples of ground truth alignment for episodes 2 and 21 with plot synopses for BF, and Fig. 4.6(c) for episode 3 of GOT. Note how the alignments in BF follow a central storyline (and are thus more contiguous and closer to the main diagonal) while GOT, with its multiple storylines presents a case for complex intermingling of

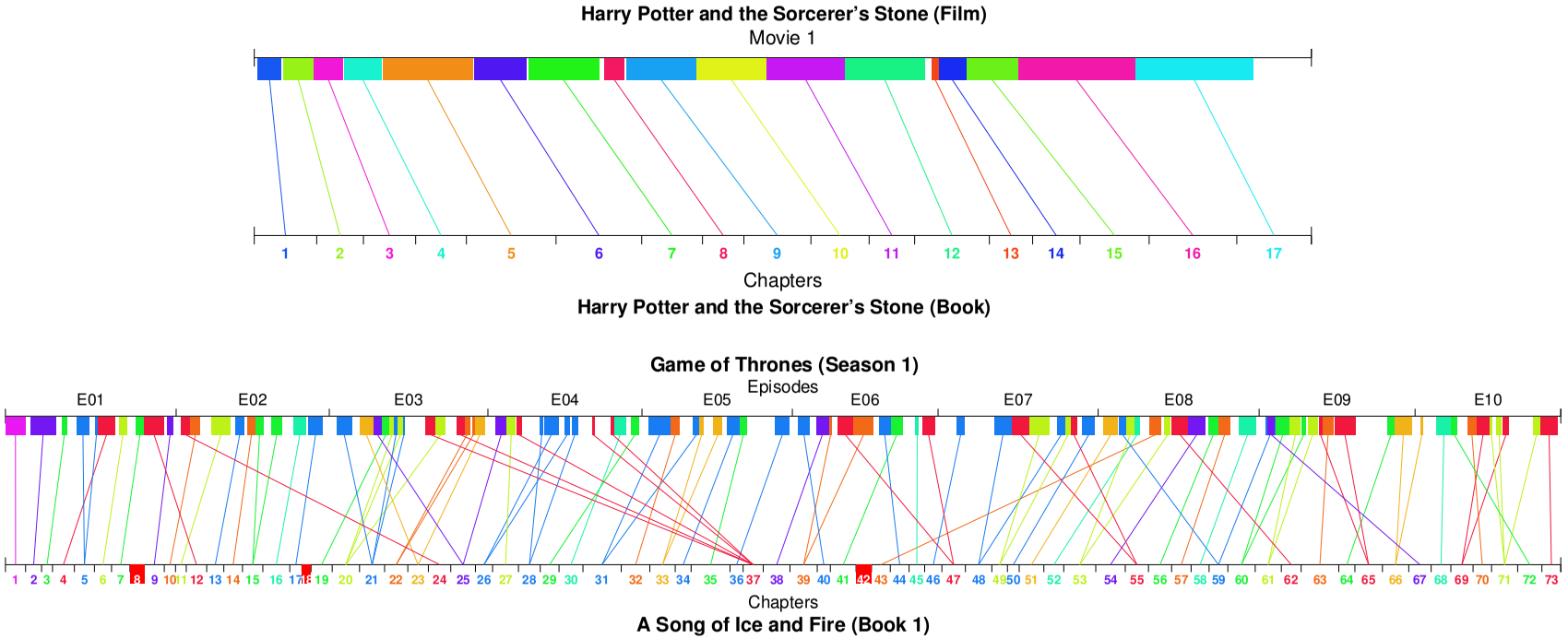


Figure 4.7: We present the ground truth alignments for chapters from the book to their TV or movie adaptations. We plot the alignment for both our data sources: *Harry Potter and the Sorcerer's Stone* (top) and *Game of Thrones* (bottom).
In both plots, book chapters are displayed on the lower axis with tick spacing corresponding to the number of words in the chapter. As the book for GOT follows a point-of-view narrative, each chapter is color coded based on the character. Chapters with a red background (8, 18, 42) are not represented in the video adaptation. The video adaptation (ten episodes of GOT E01 – E10, the first movie for HP) corresponding to the book are plotted on the upper axis. Each bar of color represents the location and duration of time in the video which corresponds to the specific chapter from the novel. A line joining the center of the chapter to the bar indicates the alignment.
While HP1 is a simple linear alignment, GOT is a perfect example of a complex narrative with multiple sub-stories which can be depicted before/after each other leading to a strongly non-sequential book-video alignment. White spaces between the colored bars indicate that those scenes are not aligned to any part of the novel: almost 30% of all shots in GOT do not belong to any chapter. Another challenge is that chapters can be split and presented in multiple scenes (*e.g.*, chapter 37 of GOT). We propose several methods to find a good alignment, among which DTW3 (Sec. 4.3.3) and SHORT (see Sec. 4.3.4) are able to cope with different storylines.

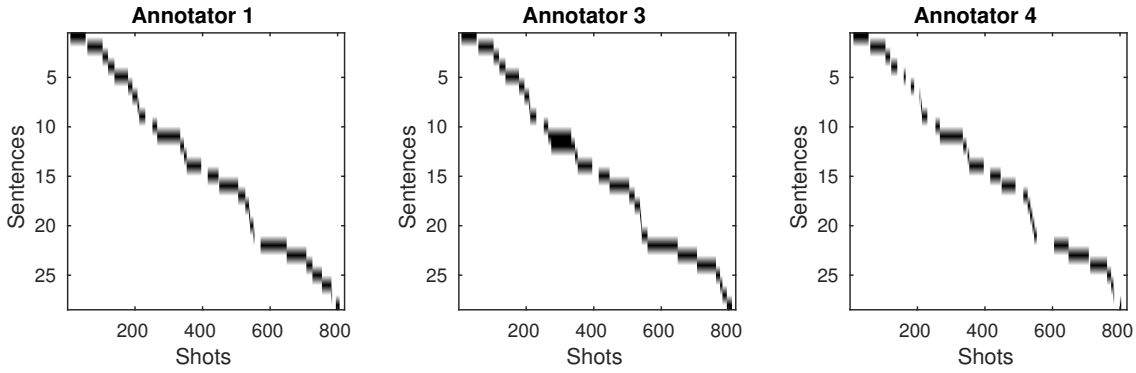


Figure 4.8: Plot synopses alignment performed by different human annotators showing the subjective nature of the problem. We present the set of video shots (x-axis) that are aligned with plot sentences (y-axis) indicated by the black segments in the graph. The Fleiss kappa agreement for this episode is 0.701 indicating *substantial agreement*.

text and video. Fig. 4.7 presents the ground truth alignment between book chapters and video scenes for HP1 and GOT, along with a description of some of the difficulties faced in the alignment involving a complex narrative consisting of multiple sub-stories. Note that the ground truth even for aligning chapters with scenes is at the shot-level. We map timestamps onto shot boundaries, as (unlike scene detection) they are assumed to be error free, and thus not influenced by an arbitrary automatic processing step. We will evaluate the impact of scene detection on the alignment performance (measured at the shot level) in the evaluation.

Gathering such ground truth annotations is a time-consuming and costly affair. Thus, we study the differences between alignments annotated by four human annotators on a small subset of the data (first four episodes of BF). Fig. 4.8 presents plot synopses alignments for the third episode of the BF season obtained from three different annotators. Note how there are significant differences, especially with respect to the boundaries of the video segments. To quantify the differences, we compute the Fleiss kappa (Fleiss, 1971) inter-rater agreement for the videos. Each shot is treated as a *data sample*, and the sentence with which it is aligned as a *category*. The $\varnothing$ category is used to represent shots that are not aligned to any sentence. The kappa values are 0.80, 0.83, 0.70, and 0.70 for the first four episodes of BF respectively, indicating *substantial* (0.61-0.80) to *almost perfect* inter-annotator agreement (0.81-1.0). We leave a more detailed analysis of this subjective problem for future work.

4.5.2 Alignment with plot synopses

We present some statistics about the used data sources and evaluate alignment of plot synopses with videos. As mentioned before, we evaluate our techniques on the complete season 5 of BF and season 1 of GOT.

Table 4.1 and Table 4.2 presents statistics and the alignment performance using different similarity functions and alignment techniques for BF and GOT respectively. We discuss a few salient aspects of the alignment data set statistics performance:

- **Statistics.** Plot synopses are fairly uniform in the number of sentences (on average, 36 for BF and 35 for GOT), however, are made up of many words (especially compared to standard image captioning, or question-answering data sets) with each sentence averaging slightly over 20 words. With over 700 shots on average, video segments are numerous in comparison to the number of plot sentences, making our many-to-one alignment scheme applicable to such a problem.

- **Structural methods.** BOW consistently outperforms DIAG validating the "climax" theory. On average, for BF, BOW shows an accuracy of 14.3% as compared to 10.1% of DIAG. Similarly, for GOT, BOW is able to correctly align 8.1% of the shots, while DIAG aligns 5.3%. We pick the best parameter for the Bézier curve using a grid search and cross-validation.

- **Similarity function cues.** Among the cues for similarity functions, identities are more important than the keywords obtained from matching plot synopses with dialogs. When using the DTW3 technique for BF, we see that identity-based alignment provides 41.2% accuracy as compared to 37.0% provided by dialogs. For GOT, we use the better performing technique to evaluate individual modalities. When using keywords, DTW3 performs slightly better, while SHORT performs better with identities. Nevertheless, we see a similar trend where identities provide 23.8% accuracy while keywords show 21.7%.

- **Shots not part of the plot.** In the tables, the "Align free" column presents the fraction of shots that are not part of the plot synopsis. 18.4% of the shots are ignored in BF, while, as the video size increases for GOT (60 minute episodes in contrast to BF's 40 minutes) 29.3% shots are left out. Assigning these shots to a sentence is counted as an error, and is a type of error that cannot be corrected by DTW2 and DTW3.

Episode	Data source statistics					Alignment performance (accuracy)							
	Video		Plot		Align	Structural		ϕ^{dlg}	ϕ^{id}	$\phi^{\text{dlg}}+\phi^{\text{id}}$			
	N_V	#FT	N_T	W/sent.	free	DIAG	BOW	DTW3	DTW3	MAX	DTW2	DTW3	SHORT
BF-01	678	780	42	16.6	18.0	2.80	37.91	21.39	40.85	24.04	18.29	**42.77**	31.27
BF-02	616	986	31	17.7	20.0	20.29	3.25	41.88	39.12	33.77	35.55	**48.05**	39.77
BF-03	820	1155	28	19.5	24.4	27.93	11.10	31.71	32.32	23.90	29.39	**32.68**	31.95
BF-04	714	875	29	20.6	29.8	4.20	13.31	24.37	37.81	26.47	36.70	**42.16**	23.95
BF-05	675	829	38	20.7	24.7	4.30	18.81	39.85	45.33	30.82	47.56	**51.11**	37.93
BF-06	745	1103	45	17.8	16.2	7.65	8.99	33.02	34.36	22.95	14.36	**35.17**	30.60
BF-07	792	847	27	27.1	10.5	12.37	22.47	52.15	31.06	39.02	29.17	**55.43**	50.76
BF-08	605	804	49	24.3	24.3	12.73	5.95	39.67	36.69	20.99	35.21	**42.98**	42.48
BF-09	664	851	45	19.3	21.2	4.67	16.57	40.21	40.96	29.67	42.17	**48.80**	45.33
BF-10	613	780	32	21.1	15.8	5.71	15.33	45.35	43.23	38.17	52.85	50.73	**54.81**
BF-11	751	1144	54	21.7	13.8	4.26	6.13	**50.73**	45.14	29.83	42.61	49.80	50.47
BF-12	797	1262	42	19.1	12.2	9.54	4.27	41.91	45.67	30.99	44.42	**55.96**	32.25
BF-13	826	985	36	20.5	15.1	5.69	17.07	37.29	48.67	41.40	58.60	**61.62**	48.91
BF-14	583	860	50	18.7	7.0	1.89	27.10	46.14	21.27	36.02	34.99	**51.97**	33.28
BF-15	754	1188	38	19.1	14.5	20.29	8.22	45.89	57.56	38.73	53.45	**60.34**	45.09
BF-16	538	786	34	16.2	18.4	9.66	2.60	27.70	43.31	24.16	30.11	**49.63**	37.36
BF-17	572	762	41	21.3	5.4	12.76	17.66	57.34	64.69	49.13	59.27	**69.93**	62.41
BF-18	792	1031	36	21.0	24.7	6.06	5.30	27.27	38.13	32.20	34.85	**39.77**	30.30
BF-19	740	930	28	21.0	12.6	16.35	10.95	32.97	54.59	35.27	51.89	**62.16**	46.08
BF-20	940	1138	24	24.5	20.6	10.00	15.85	19.79	38.94	25.53	25.53	**39.79**	26.28
BF-21	710	984	32	20.8	21.3	2.54	32.96	13.94	34.51	15.07	49.01	**51.83**	28.59
BF-22	853	973	22	15.9	34.6	20.75	12.31	**43.38**	31.54	23.45	32.01	38.92	22.86
Average	717	957	36	20.2	18.4	10.11	14.28	37.00	41.17	30.53	39.00	**49.16**	38.76

Table 4.1: Statistics and plot synopsis alignment performance for all episodes of the fifth season of *Buffy the Vampire Slayer*. We first present the number of shots and face tracks from the video source; the number of sentences and average words per sentence from the plot; and the fraction of shots that are not aligned with any sentence. We present alignment accuracy for (i) structural alignment techniques (DIAG, BOW); (ii) comparison between the influence of similarity functions from identities (ϕ^{id}) and dialog (ϕ^{dlg}) using DTW3; and (iii) comparison between proposed alignment techniques using a combination of both similarity functions. The best performance is highlighted with bold, while the second-best is underlined.

Episode	Data source statistics					Alignment performance (accuracy)							
	Video		Plot		Align	Structural		ϕ^{dlg}	ϕ^{id}	$\phi^{\text{dlg}} + \phi^{\text{id}}$			
	N_V	#FT	N_T	W/sent.	free	DIAG	BOW	DTW3	SHORT	MAX	DTW2	DTW3	SHORT
GOT-01	968	1277	34	20.9	45.5	4.03	7.64	14.98	15.29	14.98	10.74	8.47	**27.69**
GOT-02	815	1037	40	22.7	37.7	1.23	9.33	24.54	20.49	33.00	22.95	29.82	**34.36**
GOT-03	868	854	37	21.2	46.9	0.00	6.34	6.34	20.74	24.77	**38.02**	37.56	33.76
GOT-04	856	994	26	24.1	48.4	5.37	1.64	22.90	20.09	23.83	**42.17**	40.54	25.35
GOT-05	990	995	37	20.4	19.6	8.69	8.59	12.32	26.87	31.41	20.61	27.27	**35.86**
GOT-06	1145	1356	34	23.3	13.8	4.54	13.80	25.94	28.21	29.26	34.93	17.73	**37.56**
GOT-07	979	1281	31	26.4	23.4	6.33	13.79	26.35	26.05	31.36	23.19	27.68	**39.53**
GOT-08	1193	1444	38	21.2	20.3	1.34	1.17	23.13	28.25	36.63	28.08	26.40	**38.22**
GOT-09	839	1217	39	24.1	25.4	8.82	1.67	24.91	15.73	**26.10**	19.55	25.98	24.43
GOT-10	633	639	34	22.0	11.8	12.95	17.38	36.02	36.65	44.23	34.76	30.17	**53.40**
Average	929	1109	35	22.6	29.3	5.33	8.13	21.74	23.84	29.56	27.50	27.16	**35.02**

Table 4.2: Statistics and plot synopsis alignment performance for all episodes of the first season of *Game of Thrones*. We first present the number of shots and face tracks from the video source; the number of sentences and average words per sentence from the plot; and the fraction of shots that are not aligned with any sentence. We present alignment accuracy for (i) structural alignment techniques (DIAG, BOW); (ii) comparison between the influence of similarity functions from identities (ϕ^{id}) and dialog (ϕ^{dlg}) using DTW3; and (iii) comparison between proposed alignment techniques using a combination of both similarity functions. The best performance is highlighted with bold, while the second-best is underlined.

- **Impact of errors in PersonID.** The adverse impact of errors due to automatic person identification is minor as the identities get averaged when their influence is spread over a window of shots. When using perfect ground truth identities in the case of BF, alignment performance goes up from 41.1% to 47.2% when using identities as the sole similarity cue and DTW3. However, after fusion with the keywords cue, the overall improvement is from 49.2% to 51.9%. In the case of GOT, identifying characters is much harder. Thus, when using ground truth identities for alignment, GOT shows an improvement in performance of about 4% for both MAX and SHORT.

- **Alignment in the neighborhood.** For BF, the ground truth alignment between episodes and plots is significantly more linear as compared to GOT. We are able to align 71.1% of the shots within a tolerance range of ± 1 sentence from the ground truth sentence. This ± 1 sentence range, especially when considered within the same paragraph corresponds to a small time shift, on average less than one minute away. While searching within large video collections of several ten to hundreds of hours, being one minute away from the video clip is an insignificant drawback. We see a similar improvement for GOT, where the accuracy increases to 48.8% from 35.0% using the SHORT technique.

- **SHORT vs. DTW3.** SHORT performs much worse than DTW3 for episodes of BF. The primary reason is that SHORT assigns many shots to some sentences leading to problems seen in DTW2 (see Fig. 4.9). However, on GOT, we see that SHORT outperforms DTW3 by a considerable margin (35.0% vs. 27.2%). This can be attributed to the fact that GOT contains several intertwined stories, and a higher fraction of shots that are not part of the plot. SHORT is able to find such shots with a precision of 51.7% and recall of 28.0% averaged over all episodes.

- **BF vs. GOT.** The two TV series data sets used in our study show widely differing properties. GOT is much harder than BF due to two primary reasons: (i) challenging person identification with more than 50 characters; and (ii) multiple sub-stories shown intermixed, but written in the plot as separate paragraphs. Thus, we see 35% accuracy in GOT, while we are able to achieve almost 50% for BF.

- **Humans as alignment methods.** As the alignment for the first four episodes of BF was annotated by multiple humans, we evaluate humans by pitting them against each other. When using the ground truth of one, and evaluating with the other three (cyclically) we are able to obtain an averaged "human" alignment evaluation

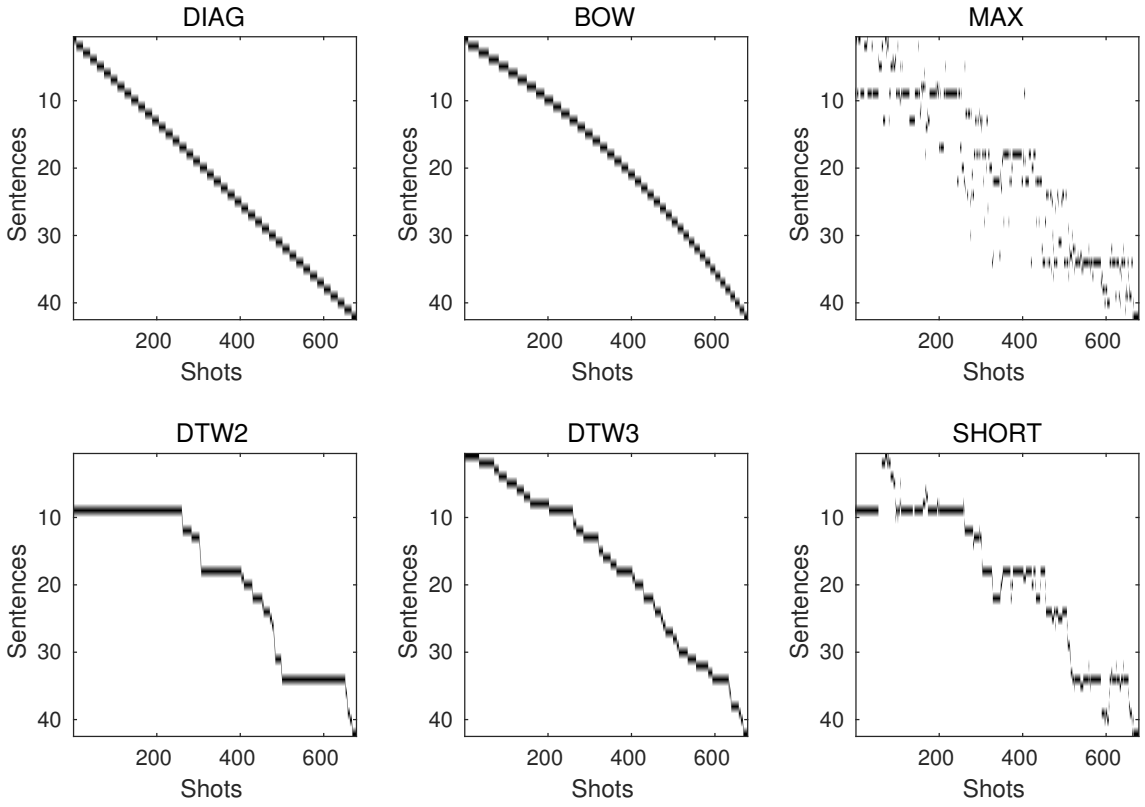


Figure 4.9: Qualitative performance of all 6 alignment methods (DIAG, BOW, MAX, DTW2, DTW3, SHORT) on the first episode of BF.

accuracy of 79.6%. As compared to the best performance among all methods 41.4% (DTW3, average of BF1-BF4), clearly there is scope for future work in this area.

Fig. 4.9 presents a qualitative analysis of the various alignment techniques for the first episode of BF. Note how DIAG and BOW attempt to perform alignment by capturing the structure of the episode, and ignoring the text-video similarity scores. In contrast, MAX treats all shots independent of each other. DTW2 and DTW3 show a semblance of connectivity, while adapting the path to go through high similarity regions. In contrast to DTW2, the length regularized DTW3 does not assign too many shots to one sentence. SHORT exhibits qualitative alignment performance somewhere between MAX and DTW2.

4.5.3 Alignment with books

We evaluate the alignment performance for books on our two data sources – GOT and HP1. In contrast to the episode level plot synopsis alignment for GOT, video scenes from

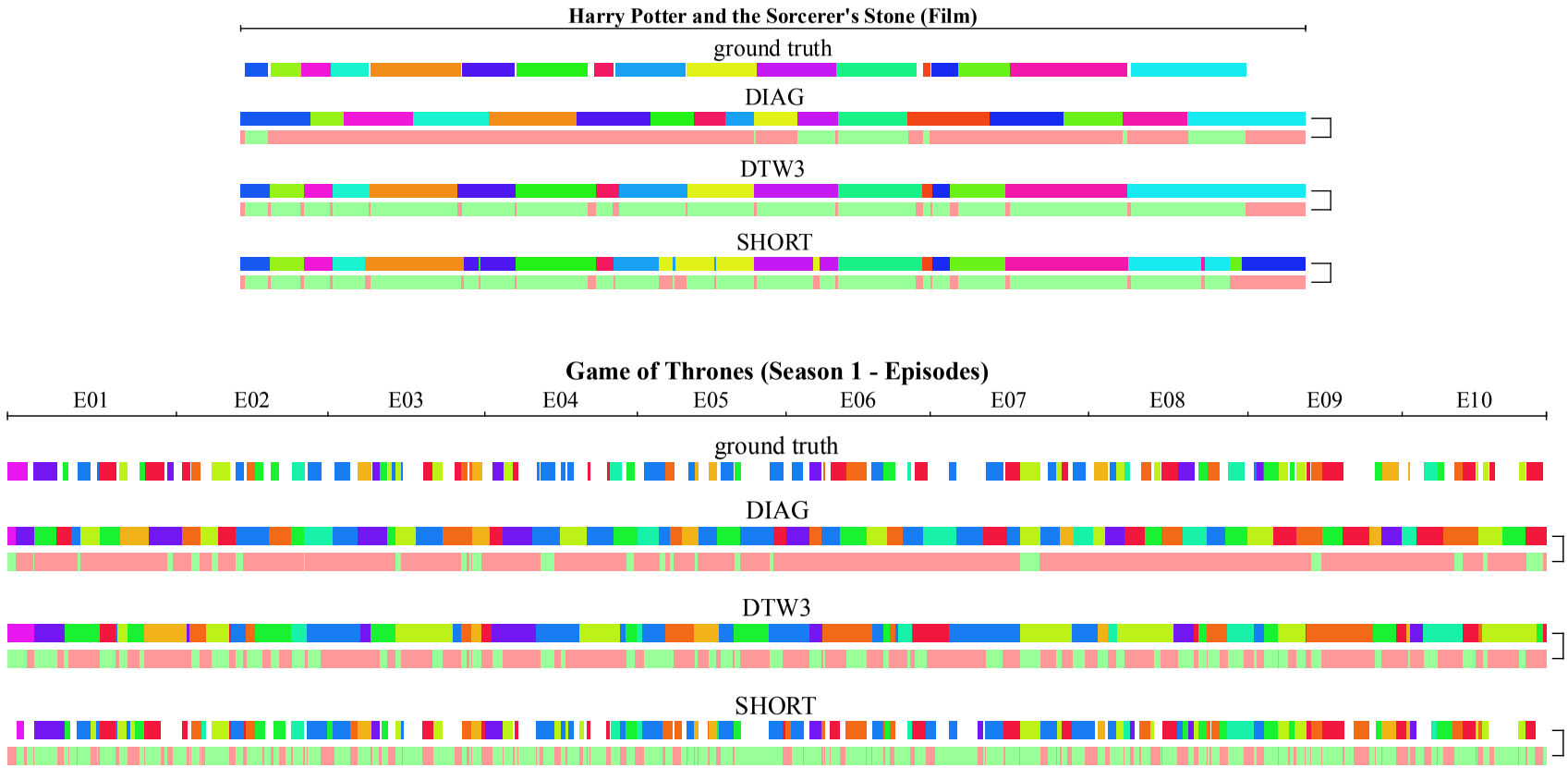


Figure 4.10: Qualitative analysis of chapter-scene alignment methods on HP1 (top) and GOT (bottom). The ground truth alignment (same as in Fig. 4.7) is presented on the top row. Each aligned scene is indicated by block of color, and the white spaces indicate scenes that are not part of the book (∅). We present the alignment performance for DIAG (row 2), DTW3 (row 3) and SHORT (row 4). The color of the first sub-row corresponds to the aligned chapters, while the second row indicates whether the alignment is correct (green) or wrong (red) and can be validated by checking the alignment vertically against the ground truth.

VIDEO		duration	N_V	#shots (#nobook)
	GOT	8h 58m	369	9286 (2708)
	HP1	2h 32m	138	2548 (56)
BOOK		N_T	#words	#adj, #verb
	GOT	73	293k	17k, 59k
	HP1	17	78k	4k, 17k
Face-ID		#characters	#FT (unknown)	id accuracy
	GOT	95	11094 (2174)	67.6
	HP1	46	3777 (843)	72.3

Table 4.3: Some statistics about the data sources for aligning book chapters with video scenes. The table is divided into three sections related to information about the *video*, the *book* and the *face identification* scheme. Note the vast differences between the two sources from their video durations and book lengths, to the number of characters, and even the number of shots that are not part of the book.

the entire season (all episodes) are aligned with the book chapters. Table 4.3 presents statistics for both sources indicating the number of chapters and scenes. We also present the number of words that are adjectives or verbs and can be used to mine attributes or frame rich descriptions of the video content. A major difference between the two sources is the number of shots that are not part of the book. In contrast to a meager 2% for HP1, about 29% of shots from GOT remain unmapped. This is in sync with the observations derived from Fig. 4.7.

We present the alignment performance obtained through the use of different alignment models in Table 4.4. Along with alignment accuracy, we also tabulate how well we are able to detect scenes which are not part of the book in terms of precision (∅-Prec) and recall (∅-Recl). The central aspects of this table are:

- **Upper bound from automatic scene detection.** As mentioned before, we collect ground truth video segments for each chapter, and map them on to shots for a more detailed evaluation. As shots are very fine-grained and on their own contain very little information (typically lasting just a few seconds), we approach the alignment task as associating video scenes with book chapters. However, the scene detection itself can influence the quality of alignment. In "Scenes upper bound" we analyze the impact of automatic scene detection on the alignment. We assign each scene to the most "correct" chapter based on the ground truth alignment and calculate the

Alignment parameters		GOT Acc	GOT ∅-Prec	GOT ∅-Recl	HP1 Acc	HP1 ∅-Prec	HP1 ∅-Recl
Scenes upper bound		95.1	97.9	86.4	96.7	40.0	7.1
Structural	DIAG	12.4	–	–	19.0	–	–
	BOW	15.6	–	–	50.3	–	–
ϕ^{id}	SHORT	55.3	52.8	48.7	80.4	0.0	0.0
ϕ^{dlg}	SHORT	73.1	55.8	74.2	86.2	20.0	3.6
$\phi^{\text{id}} + \phi^{\text{dlg}}$	MAX	54.9	–	–	73.3	–	–
	MAX+∅	60.7	68.0	37.7	73.0	0.0	0.0
	DTW2	44.7	–	–	94.7	–	–
	DTW3	44.7	–	–	**94.8**	–	–
	SHORT	**75.7**	70.5	53.4	89.9	0.0	0.0

Table 4.4: Alignment performance in terms of overall accuracy of shots being assigned book chapters (Acc). We also present the quality of detecting scenes that are not part of the book in terms of precision (∅-Prec) and recall (∅-Recl).

number of correctly assigned shots. A performance over 95% indicates that the scene detection makes few mistakes.

- **Structural methods.** Both structural methods fair badly, and assign each scene to some chapter, particularly harming GOT. Curiously, for HP1, the bow-shaped alignment model performs best in the opposite direction (*i.e.* $\gamma < 0.5$). We believe this stems from the fact that the book being the first in the series needs to spend quite some chapters establishing a new universe, while the video is able to do this at a much faster pace.

- **Dialogs stronger than identities.** Dialogs are a strong cues in aligning books with videos. Using our matching technique involving the longest common subsequence, we obtain much improved performance (73.1% for GOT with dialogs) as compared to using identities alone (55.3% for GOT with identities). However, we wish to point out that different adaptations may have varying levels of integrity with the source book, thus dialogs need not necessarily perform well. Character identities are a good fall back, especially for adaptations with major differences and demonstrate 55% and 80% accuracy for GOT and HP1 respectively.

- **Alignment using similarity function.** We present the performance of all models (MAX, DTW2, DTW3, SHORT) while using the combined similarity functions

based on identity and dialog. In case of HP1 which follows a linear sequence, DTW3 and SHORT both perform well demonstrating over 90% accuracy. Note that detecting the 2% shots which are not part of the book in HP1 is an extremely difficult task, especially as most of them are already lost owing to small errors in scene detection. For the non-linear alignment sequence of GOT, we see that SHORT (75%) strongly outperforms DTW3 (45%). In fact, even MAX (treating scenes independent) which focuses on scene-chapter nodes with a strong similarity function score performs better than DTW3. Additionally, SHORT is able to find 50% of all shots that were not part of the book (∅-Recl) with a precision (∅-Prec) of 70%.

This is similar to the observations in the case of aligning plot synopses with videos. *Buffy the Vampire Slayer* with it's simpler structure demonstrated good performance with DTW3, while most episodes of *Game of Thrones* work well with MAX and SHORT.

- **Independent assignment with ∅ class.** We introduce MAX+∅ as a method that treats scenes independent from one another, but includes an additional row in the similarity matrix representing ∅. Similar to Eq. 4.22, we treat aligning scenes to ∅ as an inverse problem, *i.e.* assignment happens when similarity with all other chapters is low. For GOT, we see that MAX+∅ outperforms MAX by about 6%, but is 15% away from SHORT, the model that incorporates the connectedness of DTW and independence of MAX.

Fig. 4.10 presents a qualitative analysis of the alignment performance for HP1 and GOT. Notice how SHORT is quite successful at predicting scenes which are not part of the book (white chunks in the first row) as compared to DTW which needs to assign all scenes to some chapter. One of the drawbacks of SHORT is when there is insufficient evidence in the form of chapter-scene similarity, it prefers assigning a video segment to ∅.

4.6 Applications

Aligning videos with rich and diverse text sources opens interesting avenues for semantic analyses. Historically, transcripts have provided information about person identities (Everingham et al., 2006) and actions (Laptev et al., 2008). In this chapter, we proposed techniques for aligning videos with plot synopses and books and evaluated the quality of

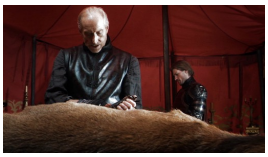
(a) BF-03 (s27, 39m37s): While moving Xander into his new apartment, Xander and Riley talk.

(b) BF-05 (s18, 24m24s): Buffy performs the ritual in her bedroom then walks around her house looking for anything unusual.

(c) BF-09 (s1, 1m43s): The Summers women are at the hospital when the doctor informs Joyce she'll have surgery in two days.

(d) BF-20 (s10, 19m31s): While Buffy battles the Knights from the top of the RV, one knight impales Giles with a thrown spear, causing the RV to crash onto its side.

(e) GOT-03 (s17, 49m33s): Before leaving the Wall, Tyrion fulfills his ambition to urinate off the top of the Wall and says goodbye to Jon, who finally accepts Tyrion as a friend.



(f) GOT-07 (s1, 2m40s): Lord Tywin Lannister has a talk with his son Jaime while the former is skinning a stag, symbolic animal of the house Baratheon.

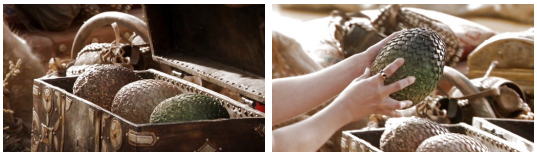
Figure 4.11: Descriptions from a plot synopsis used as a proxy for "video captioning". We show multiple examples from both our data sources (BF and GOT), along with the episode number, the plot synopsis sentence number, and the timestamps of the selected frames. Unlike typical captions, note how most plot synopses sentences are compound clauses and contain and summarize several minutes of video information.

the resulting alignments. We now present a few applications towards improving story understanding that are derived by performing such an alignment.

4.6.1 Story description

First, we present examples of rich descriptions that are obtained by aligning both plots and books. Note that plots provide a very terse description summarizing the story of the event. On the contrary, books (especially from the fantasy genre) introduce vivid visual descriptions for the story conveyed in the video.

(a) GOT (Ch11, P47, E02 12m23s): Arya was in her room, packing a polished ironwood chest that was bigger than she was. Nymeria was helping. Arya would only have to point, and the wolf would bound across the room, snatch up some wisp of silk in her jaws, and fetch it back.



(b) GOT (Ch12, P33, E01 51m21s): One egg was a deep green, with burnished bronze flecks that came and went depending on how Dany turned it. Another was pale cream streaked with gold. The last was black, as black as a midnight sea, yet alive with scarlet ripples and swirls.

(c) HP1 (Ch4, P23, M 14m23s): From an inside pocket of his black overcoat he pulled a slightly squashed box. Harry opened it with trembling fingers. Inside was a large, sticky chocolate cake with Happy Birthday Harry written on it in green icing.

(d) HP1 (Ch10, P78-79, M 1h06m59s): Hermione rolled up the sleeves of her gown, flicked her wand, and said, "Wingardium Leviosa!" Her feather rose off the desk and hovered about four feet above their heads.

Figure 4.12: Narrative descriptions obtained from a book often contain rich literary material. Even with our coarse alignment between book chapters and video scenes, we are able to leverage matching dialogs to describe videos using the book. We show two examples each from our data sources (GOT and HP1), indicating the chapter and paragraph numbers of the book, and the episode or movie timestamps in the video.

Describing using plot synopsis alignments is a fairly straightforward task. In fact, similar to the work by Rohrbach et al. (2015) where they use Descriptive Video Service to describe the video, aligning the video with plot synopsis provides semantically captioned video clips. Fig. 4.11 presents examples of several correctly aligned instances of the plot synopsis where the video clip (shown as 2 frames for simplicity) is succinctly described. As captioning and understanding improves, these methods and data can serve as very good launching pads for full-fledged story captioning (*e.g.* writing the plot synopsis for a new episode).

Books provide a unique opportunity for obtaining vast amounts of rich description. As a textual medium of story-telling, authors spend considerable effort describing the visual world in great detail. Some of these details (especially pertaining to characters) may be used for image description, but also to model intermediate attributes for characters and their images. While details provide interesting tidbits for image analysis, owing to improvements in natural language processing, especially with respect to vector embeddings

of sentences (Kiros et al., 2015b), analyzing vasts chunks of text is a promising direction. As the core story of both the video and the book is often the same, jointly analyzing their stories is an interesting future problem. Fig. 4.12 shows examples of video captioning that can be performed by narrative paragraphs from the book. While our proposed alignment only matches book chapters with video scenes, we are able to use matching dialogs within an aligned chapter-scene pair to provide this fine-grained alignment. We interactively select the closest narrative paragraph, and a few representative shots from the video, and present the resulting descriptions. Compared to descriptions from plot synopses (see Fig. 4.11), books exhibit a flavorful and non-summarized description.

In Fig. 4.11, we presented captioning through plot synopses alignment, and in Fig. 4.12 by aligning books with videos. As one of our data sources (GOT) is used in both forms, we have the unique opportunity to compare captions obtained from plots and books. Through Fig. 4.13, we show the vast difference between detailed book descriptions and single sentence plot synopses descriptions.

Note the detailed description of the scene in the first clip – *e.g.* ancient weirwood, small pool with black and cold waters, dark red leaves, *etc*. In sharp contrast, the plot sentence only provides the story outline, without referencing the visual aspects of the scene.

4.6.2 Story-based video retrieval

Aligning plot synopses with videos enables story-based video retrieval through text queries. In the general case, semantic search for story-related events is a very difficult task as it first involves translating the query and the video into a suitable common representation so as to facilitate ranking of the shots. By using plot synopses, the query can be looked up first within the plot using standard text-retrieval techniques, and by leveraging the alignment, relevant video clips can be served. We use Whoosh (Who), a full-text indexing and search library that indexes individual, and groups of two and three sentences of the plot taken at a time as independent documents. The grouping of sentences enables search within a larger time context, especially one that involves usage of pronoun references (*e.g.* He slips away in his mist form. is helped by the previous sentence Dracula attempts to re-form again ...). We use the BM25F (Zaragoza et al., 2004) algorithm (based on TF·IDF) to generate a ranked list of documents for retrieval. As each document consists of sentences that are aligned with specific parts of the video, we

#	Query	Ground Truth		Retrieval		Time
		Location	Sentence	top 5	Sentence	deviation
1	Buffy fights Dracula	E01:m35-36	(33) Buffy and Dracula fight a vicious battle.	✓	E01 (33)	$o_{IoU} = 10\%$
2	Toth's spell splits Xander into two personalities	E03:m11-12	(7) The demon hits Xander with light from a rod ... (8) He gets to his feet, walks off, but then we see there is another Xander ...	✗	–	–
3	Monk tells Buffy that Dawn is the key	E05:m36-39	(34) He tells her that the key is a collection of energy put in human form, Dawn's form.	✓	E05 (34-35)	$o_{IoU} = 31\%$
4	A Queller demon attacks Joyce	E09:m32-33	(30) In Joyce's room, the demon falls from the ceiling and spits a thick layer ...	✓	E09 (28-30)	$o_{IoU} = 12\%$
5	Willow summons Olaf the troll	E11:m18-19	(17) Willow starts a spell, but Anya interrupts it ... (18) Accidentally, the spell calls forth a giant troll.	✗	–	–
6	Willow teleports Glory away	E13:m39-39	(34) ... before Willow and Tara perform a spell to teleport Glory somewhere else.	✓	E13 (34)	$o_{IoU} = 63\%$
7	Angel and Buffy in the graveyard	E17:m14-18	(13) At the graveyard, Angel does his best to comfort Buffy when she worries ...	✓	E17 (13-14)	$o_{IoU} = 61\%$
8	Glory sucks Tara's mind	E19:m24-27	(15) Protecting Dawn, Tara refuses, and Glory drains Tara's mind of sanity.	✓	E19 (14-15)	$o_{IoU} = 74\%$
9	Xander proposes Anya	E22:m16-19	(6) Xander proposes to Anya.	✓	E22 (6)	$t_{\Delta} = 2m44s$

Table 4.5: Performance of story-based retrieval on a representative subset of queries from the BF data set. We show the textual query on the left, along with its associated ground truth location in the episode and corresponding plot synopsis sentence. To the right, we show performance of the textual retrieval, indicating whether the relevant document (plot sentences) were returned in the top 5 results, and the document index (which covers the ground truth document). Finally, we show the time deviation or amount of overlap between the video clip and the ground truth timestamps. E01:m35-36 means minutes 35-36 of episode 1. (33) indicates sentence number 33.

(a) **BOOK** (Ch3, P8, E01 19m50s): At the center of the grove an ancient weirwood brooded over a small pool where the waters were black and cold. "The heart tree," Ned called it. The weirwood's bark was white as bone, its leaves dark red, like a thousand bloodstained hands. A face had been carved in the trunk of the great tree, its features long and melancholy, the deep-cut eyes red with dried sap and strangely watchful.(P28) Catelyn took her husband's hand. "There was grievous news today, my lord. I did not wish to trouble you until you had cleansed yourself." There was no way to soften the blow, so she told him straight. "I am so sorry, my love. Jon Arryn is dead."

PLOT (s19): Back at Winterfell, Catelyn informs her husband of a letter announcing the death of Lord Arryn, Eddard's old mentor and Catelyn's brother-in-law.

(b) **BOOK** (Ch23, P83, E03 50m37s): Three days later, at midday, her father's steward Vayon Poole sent Arya to the Small Hall. The trestle tables had been dismantled and the benches shoved against the walls. The hall seemed empty, until an unfamiliar voice said, "You are late, boy." A slight man with a bald head and a great beak of a nose stepped out of the shadows, holding a pair of slender wooden swords. "Tomorrow you will be here at midday," He had an accent, the lilt of the Free Cities, Braavos perhaps, or Myr. (P90) Arya took her right hand off the grip and wiped her sweaty palm on her pants. She held the sword in her left hand. He seemed to approve. "The left is good. All is reversed, it will make your enemies more awkward. Now you are standing wrong. Turn your body sideface, yes, so. You are skinny as the shaft of a spear, do you know. That is good too, the target is smaller. Now the grip. Let me see." He moved closer and peered at her hand, prying her fingers apart, rearranging them. "Just so, yes. Do not squeeze it so tight, no, the grip must be deft, delicate."

PLOT (s29): Learning that his younger daughter aspires to be a swordsman and has a sword of her own, he hires a Braavosi "water dancer", Syrio Forel to teach her the art of swordsmanship.

Figure 4.13: As the data source *Game of Thrones* is aligned with both plot synopses and books, we present two examples depicting the differences in the book and plot information. Note how large parts of the book (not all displayed) are represented using a few shots of the video, and are depicted in the plot synopsis as a single sentence.

are able to easily associate a video clip with each retrieved document, thus providing a means for story-based video search.

We evaluate the performance of story-based retrieval on a total of 62 queries related to story events in the entire fifth season of *Buffy the Vampire Slayer*. The queries are sourced uniformly from the 22 episodes (about 2-5 from each). To reduce the bias involved in creating text queries, a subset of our queries are obtained from a fan forum[3]

[3] www.buffy-boards.com

based on the TV series. The other queries are collected by us, about 6 months after the alignment annotations, and *without* looking at the text of the plot synopsis. The queries are associated with ground truth timestamps during which the story appears in the episode.

Given a query, the first stage involves searching through plot synopsis documents (sentences and their groups) sourced from all episodes. This provides us a ranked list of documents, and we deem text-retrieval a success when the story appears in the document. Of the 62 queries, 24 (38%) obtain correct documents in the first position, 43 (69%) in the top 5 and 48 (77%) are within the top 10 (first page) of results. For 9 of 62 queries, we are unable to find a relevant document. This is partially due to the limited vocabulary of the plot and our choice of document retrieval scheme which is based on word-matching. This problem may be alleviated by resorting to techniques which embed words or sentences into a vector space (Kiros et al., 2015b; Mikolov et al., 2013).

In the second stage, we serve the video clip relevant to the query. Here, we evaluate on 53 (of 62) queries for which we find a document. We are able to provide a video clip (of roughly 2-3 minutes) which has an overlap with the ground truth segment for 40 (75%) of these queries. For 13 clips, the video timestamps we return are on average 3 minutes away from the ground truth segments. Note that this is a very small deviation considering we search within 15 hours of video.

Table 4.5 presents qualitative results for a range of queries that perform well or for whom we are unable to find the relevant document. We are able to find matches for most queries without a large number of common words. The two failure cases (query 2 and 5) demonstrate a peculiar case of story abstraction where multiple sentences and future knowledge (*e.g.* name of the demon) are combined.

4.6.3 Differences between books and their adaptations

Another application that arises from the alignment is the ability to find differences between the book and its video adaptation. Determining how specific parts of the book are adapted to the video domain is not only an interesting problem for performing arts literature (Giddings et al., 1990; McFarlane, 1996; Wagner, 1975), but pin-pointing differences between them is also a fan-favorite task[4]. It is important to note that video

[4] There are fan websites which track differences such as http://thatwasnotinthebook.com.

adaptations are interpretations of the story, and not just an illustrated version of the book[5].

Our alignment method SHORT (Sec. 4.3.4) provides a straightforward approach to find differences between books and their adaptations. For example, we are currently able to predict (with about 70% precision) whether a video scene was not part of the book. We believe this to be an important aspect as a first step in understanding of books and movies. We leave fine-grained differentiation (*e.g.* character or scene appearance) for future work.

[5] http://www.kvenno.is/englishmovies/adaptation.htm presents a nice summary of the differences and freedoms that the textual and visual media offer while depicting the same underlying story

Chapter 5

StoryGraphs: Visualizing Character Interactions

This chapter is primarily based on the work published in Tapaswi et al. (2014b). Additional elements are the overlay of event and location labels on to the StoryGraphs, and a user-driven story event retrieval experiment.

Today, story-telling is seen in a variety of media forms such as books, movies, TV series, and even news and documentaries. Among all of these stories, an important aspect that makes them relatable to our lives are characters and their interactions[1]. Note that the characters need not take a human form and could be animals (as is the case in many children's books), or even inanimate objects (as are often used to inculcate creative writing). However, *who did what with/to whom* is a common underlying theme of all stories.

Inspired by a popular web comic XKCD[2], we present a technique to automatically visualize character interactions in TV series episodes. The original comic strip is hand-drawn by Randall Munroe and depicts the temporal dynamics of the characters in the video in a single image. Overlaid with "event bubbles" showing the location or situation in that scene, the chart succinctly portrays the story conveyed in the movies. Such charts

[1] https://www.writersandartists.co.uk/writers/advice/410/a-writers-toolkit/story-and-plot/ presents a strong case for character development in the plot lines of a story.
[2] http://xkcd.com/657

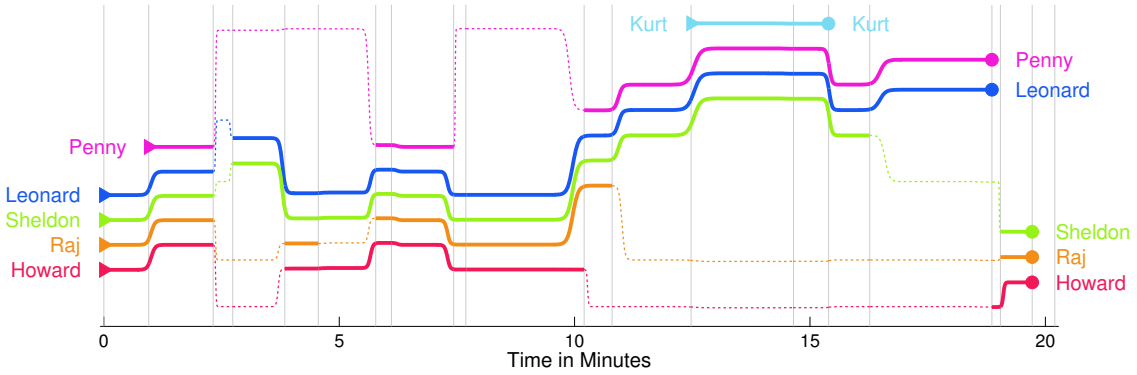


Figure 5.1: StoryGraph generated for season 1, episode 6 of the TV series *The Big Bang Theory*. The graph clearly shows the 5 primary characters (*Penny, Leonard, Sheldon, Raj* and *Howard*) interacting together at the start of the video (first 7 minutes), while splitting into two distinct groups: (1) *Penny* and *Leonard*, and (2) *Sheldon, Raj* and *Howard*, at the end of the video (16 - 20 minutes).

are a unique take on video summarization, where the story in a long video is conveyed through a single image.

Sec. 2.4 presents a thorough analysis of automatic story visualization techniques that have been addressed in the domain of TV series or movies. We also discuss good practices that are commonly employed in visualization.

StoryGraph design. In this chapter, we attempt to automatically recreate such charts, called *StoryGraphs* (see Fig. 5.1). Unlike the comic, our graph uses a linear horizontal time axis demarcated by vertical gray lines that correspond to scene boundaries that are detected using a combination of color-similarity and shot threading (see Sec. 3.4.3). Each horizontal line corresponds to the progression of the character over time, and is solid (dotted) when the character appears (does not appear) in the scene. Vertical grouping of characters indicates that they co-occur in the scene. The start (▶) and end (●) of a horizontal line indicate the first and last appearance of the character in the episode. For example, in Fig. 5.1 we see that *Kurt* appears only for a brief period in the entire episode.

StoryGraphs are plotted based on the identity of characters visible on-screen and are thus a direct application of years of research on identifying characters in TV series (Bäuml et al., 2013; Everingham et al., 2006; Sivic et al., 2009). StoryGraphs lend themselves to and enable applications such as *smart browsing* or *video retrieval*. For example, a video player's seek bar can be augmented by the chart, thus providing a sneak-preview of what happens in the video, while not revealing any major spoilers in the plot. While we

propose the visualizations as a means to analyze story videos, note that the technique used to layout StoryGraphs is generic and can be easily used to generate similar charts to analyze a discussion between people (*e.g.* in meeting rooms, group discussions, and post-event analysis for control rooms).

5.1 StoryGraphs layout

We formulate the layout for StoryGraphs as an optimization problem that attempts to minimize the energy of the system. The free variable that determines the structure of the graph is the y-coordinate for each character and time segment (video scene). Drawing the horizontal character lines is essentially transformed into the problem of connecting the dots (coordinates) for each scene. Note that, in contrast to typical graph layouts (*e.g.* force-directed (Kobourov, 2012)), our graph is essentially a one-dimensional layout problem – involves estimating the placement of characters within a scene – that is linked through time.

5.1.1 Energy minimization

Any visualization involves a trade off between functionality and aesthetics. We design our energy function to capture four desirable properties of StoryGraphs.

Our free variable, x_t^c denotes the y-coordinate through which the line for character c passes during the scene (time period) t. The total number of characters and scenes is denoted by N_C and N_T respectively. For brevity, we define the total number of pairwise interactions between N_C characters as $N_P = N_C \cdot (N_C - 1)/2$.

We define a combined energy function on temporal and spatial coordinates $\mathbf{x}$ for all characters that consists of four terms that capture our desired properties:

$$\mathscr{E}(\mathbf{x}) = \underbrace{w_p E_p(\mathbf{x})}_{\text{proximity}} + \underbrace{w_c E_c(\mathbf{x})}_{\text{crossings}} + \underbrace{w_s E_s(\mathbf{x})}_{\text{straight lines}} + \underbrace{w_m E_m(\mathbf{x})}_{\text{min. separation}} . \tag{5.1}$$

w_p, w_c, w_s and w_m are weight terms designed to emphasize the importance of different energy terms. The optimal line positions are obtained by constrained function minimization

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}}\, \mathscr{E}(\mathbf{x}) \quad \text{such that } 1 \leq x_t^c \leq N_C . \tag{5.2}$$

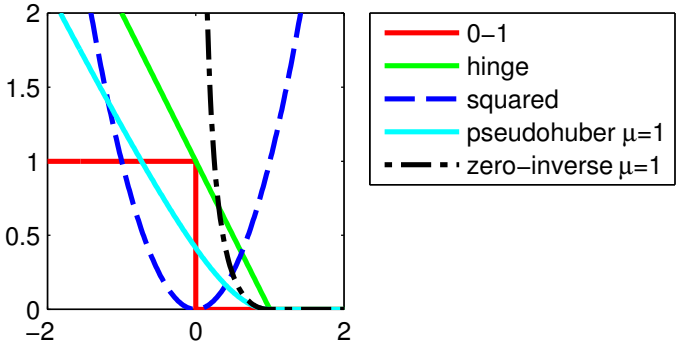


Figure 5.2: Visualization of typical loss functions used in machine learning. Note that we only use completely differentiable functions in this work (squared, pseudo-Huber and zero-inverse) to facilitate optimization through gradient computation.

We describe the four energy terms from Eq. 5.1 below.

1. Proximity. Estimating the spatial (vertical) position of character lines is the most important criterion for drawing a good StoryGraph. Following the original `XKCD:657` comic, lines for characters that appear together are drawn close to each other, while characters which are not visible in the scene are pushed apart. The proximity energy E_p consists of two parts. The first component

$$E_p^{(1)}(\mathbf{x}) = \frac{1}{N_P N_T} \sum_{c_i,c_j,t} p_{c_i,c_j,t} \cdot (x_t^{c_i} - x_t^{c_j})^2, \tag{5.3}$$

is responsible for reducing the separation between lines of characters that co-occur in a scene.

Here, $p_{c_i,c_j,t} \in [0,1]$ is the normalized "co-occurrence score" between two characters c_i and c_j during time period t, and is computed using the geometric mean of the number of frames in which c_i and c_j appear. To reduce the impact of erroneous person identification, we say that character c appears in the scene only if he/she appears in at least θ_f frames. When two characters appear for a long duration during the scene, their co-occurrence score is high, and minimization of $E_p^{(1)}$ encourages $x_t^{c_i}$ and $x_t^{c_j}$ to come closer.

The second component of the proximity energy is

$$E_p^{(2)}(\mathbf{x}) = \frac{1}{N_P N_T} \sum_{c_i,c_j,t} \mathbb{1}(p_{c_i,c_j,t} = 0) \cdot \left(x_t^{c_i} - x_t^{c_j}\right)^2. \tag{5.4}$$

This term, when maximized, encourages to push away lines of characters that do not appear (encoded using $p_{c_i,c_j,t} = 0$) in the scene.

The combined proximity energy function is $E_p = E_p^{(1)} - E_p^{(2)}$ and its joint minimization forces the lines for character c_i and c_j to come closer (or move apart) depending on whether they appear (or not). While ideally we would like to minimize $|x_t^{c_i} - x_t^{c_j}|$, we use the squared loss (Fig. 5.2-blue) to obtain a differentiable function with respect to x_t^c

$$\frac{\partial E_p^{(1)}(\mathbf{x})}{\partial x_t^c} = \frac{1}{N_P N_T} \sum_{c'} 2 \cdot p_{c,c',t} \cdot (x_t^c - x_t^{c'}). \tag{5.5}$$

We use these gradients to speed-up optimization.

2. Line crossings. A second and important property in improving the aesthetics of StoryGraphs is to reduce the number of character line crossings. As the number of characters increases, tracking characters across an episode can get quite difficult, and is made worse in the presence of line crossings. The number of line crossings can be easily computed by counting the number of pairs that satisfy $(x_t^{c_i} - x_t^{c_j})(x_{t+1}^{c_i} - x_{t+1}^{c_j}) < 0$ for all combinations of $(c_i, c_j, t, t+1)$.

To discourage crossings, we penalize the energy function through a smooth and differentiable hinge function: the pseudo-Huber function (Huber, 1964) (Fig. 5.2-cyan).

$$\mathscr{H}(x, \mu) = \begin{cases} \sqrt{1 + (x-\mu)^2} - 1 & x < \mu \\ 0 & x \geq \mu\,. \end{cases} \tag{5.6}$$

The line crossings energy function is formulated as

$$E_c(\mathbf{x}) = \frac{1}{N_P N_T} \sum_{c_i,c_j,t} \mathscr{H}\left((x_t^{c_i} - x_t^{c_j})(x_{t+1}^{c_i} - x_{t+1}^{c_j}), \mu_c\right), \tag{5.7}$$

where we set $\mu_c = 0$, *i.e.* the energy is 0 when lines do not cross. The gradient with respect to x_t^c is

$$\frac{\partial E_c(\mathbf{x})}{\partial x_t^c} = \frac{1}{N_P N_T} \sum_{c'} \Big[(x_{t-1}^c - x_{t-1}^{c'}) \mathscr{H}' \left((x_{t-1}^c - x_{t-1}^{c'})(x_t^c - x_t^{c'}) \right) + (x_{t+1}^c - x_{t+1}^{c'}) \mathscr{H}' \left((x_{t+1}^c - x_{t+1}^{c'})(x_t^c - x_t^{c'}) \right) \Big]. \tag{5.8}$$

3. Line straightness. The above energy functions warp character lines to fulfill proximity and crossing constraints. However, straight character lines are aesthetically preferable to ones that wiggle around. We define the mean position for character c's line evaluated at all time periods except t as

$$\mu_{\setminus x_t^c} = \frac{1}{N_T - 1} \sum_{\tau \neq t} x_\tau^c. \tag{5.9}$$

At every time step, we wish to move x_t^c close to $\mu_{\setminus x_t^c}$. The straightness enforcing energy function to be minimized is defined based on the amount of deviation from the mean

$$E_s(\mathbf{x}) = \frac{1}{N_C N_T} \sum_{c,t} (x_t^c - \mu_{\setminus x_t^c})^2, \tag{5.10}$$

and has a gradient

$$\frac{\partial E_s(\mathbf{x})}{\partial x_t^c} = \frac{1}{N_C N_T} 2(x_t^c - \mu_{\setminus x_t^c}). \tag{5.11}$$

4. Minimum separation. The proximity loss $E_p^{(1)}(\mathbf{x})$ brings lines of characters that appear together closer and is minimized when $x_t^{c_i} = x_t^{c_j}$. However, this is a degenerate case for visualization, as the character lines overlap and are not visible. We introduce an opposing energy function $E_m(\mathbf{x})$ that ensures a minimum separation between the lines and prevents them from collapsing on top of each other. This is function is based on a smooth differentiable modification of $1/x$ going to $0, \forall x \geq \mu$ and ∞ at $x = 0$. We call this the zero-inverse function (Fig. 5.2-black)

$$\mathscr{I}(x, \mu) = \begin{cases} \frac{1}{x} \cdot (\sqrt{1 + (x - \mu)^2} - 1) & 0 < x < \mu \\ 0 & x \geq \mu. \end{cases} \tag{5.12}$$

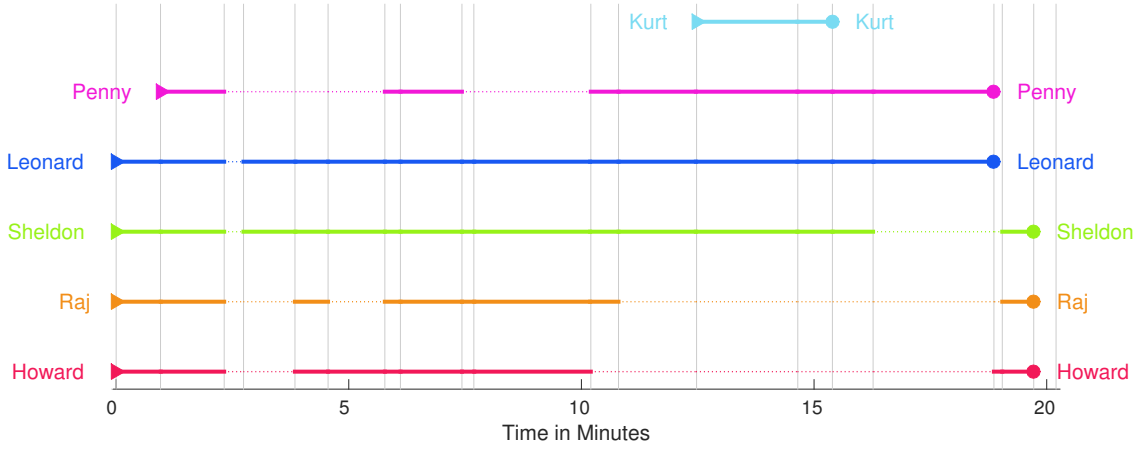


Figure 5.3: StoryGraph for episode 6, season 1 of *The Big Bang Theory* just after initialization using the optimal leaf ordering. The result post-optimization is presented in Fig. 5.1.

The minimum separation energy function is defined as

$$E_m(\mathbf{x}) = \frac{1}{N_P N_T} \sum_{c_i, c_j, t} \mathscr{I}\left((x_t^{c_i} - x_t^{c_j})^2, \mu_s\right), \tag{5.13}$$

where μ_s is the desired minimum separation between all lines on the graph. The gradient can be computed as

$$\frac{\partial E_m(\mathbf{x})}{\partial x_t^c} = \frac{1}{N_P N_T} \sum_{c'} 2 \cdot \mathscr{I}'((x_t^c - x_t^{c'})^2, \mu_s) \cdot (x_t^c - x_t^{c'}). \tag{5.14}$$

5.1.2 Implementation details and drawing procedure

We present a few details about the optimization procedure.

- The optimization (Eq. 5.2) is implemented using Matlab's constrained function minimization `fmincon`, that has access to the gradients computed using the equations discussed above.

- The overall energy function to be minimized $\mathscr{E}(\mathbf{x})$, is not convex, and initialization is therefore an important aspect of the optimization. We use the co-occurrence scores between characters (through all scenes) as a form of "distance" and perform hierarchical agglomerative clustering to order characters. The clustering yields an optimal leaf ordering (Bar-Joseph et al., 2001) that is used to initialize our spatial

coordinates at iteration 0, $\mathbf{x}^0$. Fig. 5.3 presents the post-initialization version of the StoryGraph for the same episode as in Fig. 5.1.

- Consider a case where character A co-occurs strongly with C, while weakly with B, *i.e.* $p_{A,C,t} > p_{A,B,t}$. However, due to a bad initialization, it could happen that B's line appears between A and C. To move the A's line closer to C, we need to jump over B's line, which when done with small steps (*e.g.* due to a small learning rate) brings A close to B. However, the minimum separation gradient strongly opposes bringing lines of A and B close to each other, leaving the optimization procedure in a deadlock. To circumvent this, after every 50 iterations of `fmincon`, we make an additional "swapping pass" that exchanges the positions of lines for all pairs of characters, computes the energy function in the original and swapped configuration, and accepts the latter when it results in a lowered energy.

- The energy functions are evaluated only for the duration during which the character appears in the episode. All other time periods are masked away and do not contribute to the energy.

- Weights for combining the various energy functions (Eq. 5.1) are chosen by eyeballing the visual quality of generated StoryGraphs. While we select a different set of weights for each TV series as the number of characters and the episode durations change, we observe trends in the weights making them easier to choose. The order of magnitudes within which the weights (proximity, crossings, straightness, minimum separation) are set are: $w_p \sim 1, w_c \sim 0.1, w_s \sim 1, w_m \sim 100$. As all our energy functions are normalized, the weights allow for a valid interpretation where enforcing minimum separation is a must (emphasized by the high value of w_m), while reducing crossings is not very critical (w_c is smaller than all other weights by one order of magnitude).

Drawing. To draw the StoryGraph, we start by first plotting the scene boundaries. Within each scene, we position and plot character lines based on the optimized y-coordinate. Characters that appear in the scene are displayed with solid lines, while those that do not appear with a dotted line. Finally, we generate smooth transitions when the positions of character lines change from one scene to another using the sigmoid function.

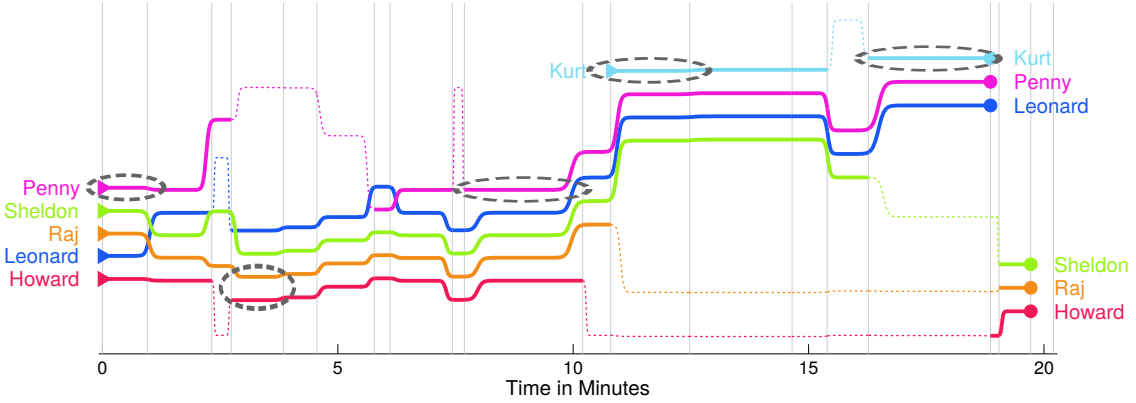


Figure 5.4: StoryGraph for episode 6, season 1 of *The Big Bang Theory* using ground truth person identification. Compare this against the visualization generated using automatic person identification Fig. 5.1. The scenes where characters are predicted as absent (in Fig. 5.1) are highlighted using gray dashed circles.

5.2 Evaluation

We evaluate generation of StoryGraphs on three diverse TV series encompassing situational comedies (BBT, 6 episodes, ∼20 min), supernatural drama (BF, 6 episodes, ∼40 min), and fantasy drama (GOT, 10 episodes, ∼55 min). While the former two series have a central storyline, the latter GOT presents multiple sub-stories making StoryGraphs particularly interesting to visualize.

Impact of person identification. The number of characters in the three TV series varies widely: 11, 27 and 66 named characters for BBT, BF, and GOT respectively. To generate StoryGraphs we ignore unknown background characters (whose names the audience does not learn), however, we do keep minor roles as although they appear for a short duration, they might form an integral part of the episode.

The impact of person identification on StoryGraphs can be quantified by counting the number of segments (x_t^c) when a character is classified as present in a scene (and drawn with a solid line) while he/she is not present, and vice versa. Table 5.1 displays the person identification performance, and the absolute count and the ratio (with respect to $N_C \cdot N_T$) of erroneous segments. While the errors in face track classification are over 25% for BF and GOT, we are able to reduce their impact while generating the visualization and see less than 10% segments in error. This is primarily due to the pooling of track identities within a scene to compute co-occurrence scores, thus averaging out the error. Fig. 5.4

	BBT	BUFFY	GOT
#Episodes	6	6	10
#Characters	11	27	66
Mean Accuracy	92.36%	78.12%	75.25%
SG Presence Error Count	33	260	680
SG Presence Error Fraction	4.85%	8.08%	4.24%

Table 5.1: Errors in automatic person identification adversely impact the quality of StoryGraphs. We present the identification performance for each TV series and count the number of errors made in the visualizations.

presents the StoryGraph for episode 6, season 1 of BBT when generated using ground truth person identities. Note that we are able to reproduce the primary structure of the chart while using automatic id without problems (see Fig. 5.1).

5.2.1 Qualitative evaluation

Evaluating any visualization quantitatively is a hard task, however, based on our desired properties, we define some measures to quantify the performance of StoryGraphs.

1. **Move** is a maximum coordinate movement score for each character that is ideally small, as we prefer to obtain straight characters lines. It is defined as

$$\text{Move} = \frac{1}{N_C} \sum_{c} \left(\max_t x_t^c - \min_t x_t^c \right). \tag{5.15}$$

2. **#Cross** counts the number of line crossings

$$\#\text{Cross} = \sum_{c_i, c_j, t} \mathbb{1}\left((x_t^{c_i} - x_t^{c_j})(x_{t+1}^{c_i} - x_{t+1}^{c_j}) < 0 \right). \tag{5.16}$$

3. **MaxSep**$_t$ is a metric related to the separation between character lines. We define this as the worst case minimum separation between two characters who are present during the scene:

$$\text{MaxSep}_t = \max_{c_i} \left(\min_{c_j, j \neq i} |x_t^{c_i} - x_t^{c_j}| \right). \tag{5.17}$$

   The above number picks the maximum separation between two lines which are adjacent (due to the minimization of the internal term) to each other. Ideally,

	The Big Bang Theory (BBT)					
	E01	E02	E03	E04	E05	E06
Move	0.32	0.34	0.19	0.31	0.22	0.30
#Cross	0	1	0	0	0	0
#MaxSep $> 2\mu_s$	4	12	11	11	10	9
#MaxSep $< 0.5\mu_s$	1	1	1	1	1	1

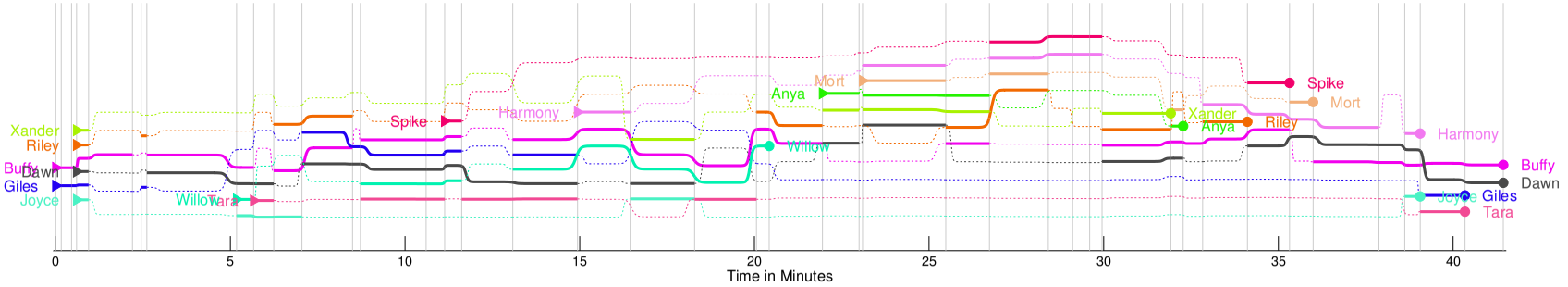
	Buffy the Vampire Slayer (BF)					
	E01	E02	E03	E04	E05	E06
Move	0.46	0.29	0.21	0.28	0.15	0.39
#Cross	83	43	19	46	8	128
#MaxSep $> 2\mu_s$	0	2	7	27	13	1
#MaxSep $< 0.5\mu_s$	1	0	1	1	1	0

	Game of Thrones (GOT)									
	E01	E02	E03	E04	E05	E06	E07	E08	E09	E10
Move	0.30	0.21	0.21	0.18	0.17	0.12	0.20	0.19	0.18	0.19
#Cross	212	66	134	44	44	26	116	60	44	26
#MaxSep $> 2\mu_s$	1	6	31	11	25	14	1	34	6	8
#MaxSep $< 0.5\mu_s$	1	3	2	3	4	3	3	2	4	2

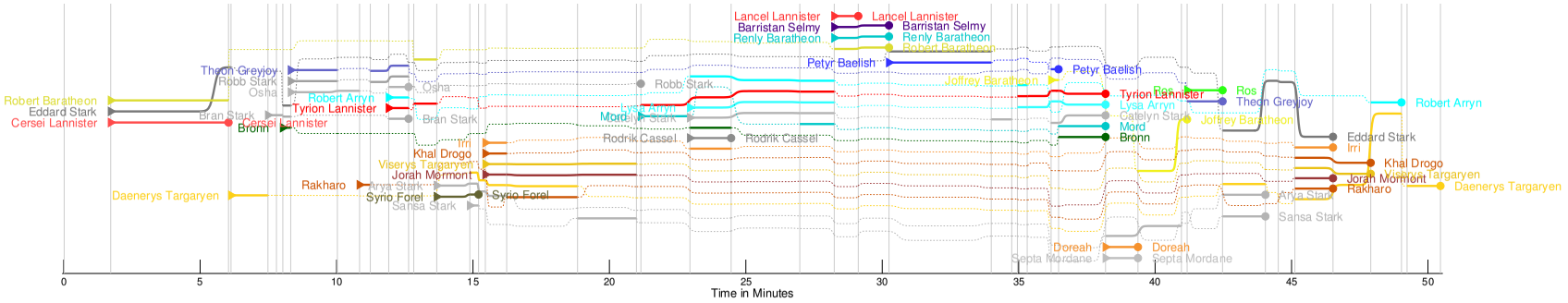
Table 5.2: Evaluation of the quality of StoryGraphs over all episodes from our three data sources (BBT, BF, GOT). The StoryGraphs are generated using automatic person identification.

this should be close to the minimum separation width μ_s as the proximity energy function $E_p^{(1)}$ pushes lines together, while the E_m asserts a minimum separation between the lines. Using this, we count the number of violations when the lines are too far apart (MaxSep $> 2\mu_s$) and when they are too close to each other (MaxSep $< 0.5\mu_s$).

Table 5.2 displays the metrics for StoryGraphs generated on all episodes. Among them BBT is the most consistent with a very low number of crossings and MaxSep $< 0.5\mu_s$. In contrast, we see that GOT exhibits a large number of crossings, some of which are justified owing to the multi-story scenario. All series exhibit very few violations of the minimum separation constraint, partially owing to the strong emphasis laid on that term through the usage of a high weight w_m (evident due to a low count for #MaxSep $< 0.5\mu_s$). The proximity function is also able to push lines for characters that appear in a scene closer together (counted using #MaxSep $> 2\mu_s$).



(a) The episode starts with Buffy and Giles in training (m1). We then see Dawn annoy Buffy while Joyce asks her to take her shopping (m3-6). Buffy has to cancel her plans with Riley (m6-7) as Giles drives off with the two sisters. They meet with Willow and Tara and discover that the owner of the magic shop is killed (m9-11). Tara consoles Dawn as she is kept out of the gang (m12) while the others discuss future plans (m13-15). As Joyce leaves home, Xander keeps Dawn company (m17) while Buffy rants to Riley (m21). Harmony and her gang visit Buffy's home only to find that she is not there (m22-25). On reaching home, Buffy is informed that Dawn invited Harmony to come into the house (m25-27) and she goes out in search of Harmony (m35). Harmony, after her failed plan, later encounters Spike (m28-30). In the meanwhile, Mort kidnaps Dawn and brings her to the lair (m33). We see a fight between Harmony's minions and Buffy, who rescues her sister Dawn (m36-38) after which we cut to the shop where Dawn is writing about Buffy in her diary (m40-41).



(b) Note the clear demarcation between groups of characters from different sub-stories and places in the fantasy world. We see scenes from *King's Landing* @ m1-6, m12-15, m27-35, m42-45 involving Robert Baratheon, Cersei Lannister, Eddard, Arya and Sansa Stark and Petyr Baelish; from the *Eyrie* @ m21-m27 and m35-38 involving Catelyn Stark, Rodrick Cassel, Lysa and Robert Arryn, and the imprisonment of Tyrion Lannister; from *Winterfell* @ m8-12 primarily involving Bran and Robb Stark and Theon Greyjoy; and finally *Beyond the Narrow Sea* @ m15-20 and m45-50 involving Targaryens, Khal Drogo and Jorah Mormont.

Figure 5.5: StoryGraphs presented along with a short description of the story events that take place at different time durations. The detailed description for BF (top) is obtained from the corresponding plot synopsis. Due to lack of space, we only present the grouping and locations of the stories for the episode of GOT (bottom).

StoryGraph in relation with stories. StoryGraphs provide a way to generate a snapshot of the story. We can estimate the representation ability of generated visualizations by comparing them against compare a short description of the episode, *e.g.* the plot synopsis obtained from Wikipedia. Fig. 5.5 shows examples StoryGraphs for episode BF-02 and GOT-06. For brevity, we show plot synopsis description in the figure caption, and video segments from minute A to B are represented as "mA-B".

Reviewer opinions. As judging the quality and usefulness of StoryGraphs is a subjective problem, we present some of the positive and negative aspects reviewers pointed out during the review phase of Tapaswi et al. (2014b). The reviewers thought StoryGraphs as an "interesting research area linking previous work on face naming in TV series", a "clever idea for the problem of efficient and intelligent scanning of lots of visual information" and "being able to generate them in an automated manner is a step forward for ways in which we can look at video summaries". However, they also had problems comprehending the end-user application: "what advantage does the narrative chart provide to the audience" or "the results did not convince me about the usefulness of the idea from an end-user perspective". To demonstrate the use of StoryGraphs, we perform a user trial to see how well people are able to "read" StoryGraphs and find events in the video.

5.3 User experiment on story event retrieval

In Sec. 4.6.2, we presented an approach to perform automatic story-based video retrieval by bridging the gap between textual queries and video shots through plot synopses. Here, we present a user-driven story event retrieval experiment and compare the impact of the availability of StoryGraphs.

5.3.1 Event labels from plot synopses

Prior to the discussion of the retrieval experiment, we create event bubbles (similar to the original comic `XKCD:657`) depicting the current status of the story and add them to the chart. Fig. 5.6 presents an example of such a graph, where we see the character lines augmented with information about location (*e.g.* High IQ sperm bank), story-specific events (*e.g.* Leonard and Sheldon discover a new neighbor; and Penny uses the guys' shower), and social events (*e.g.* Dinner).

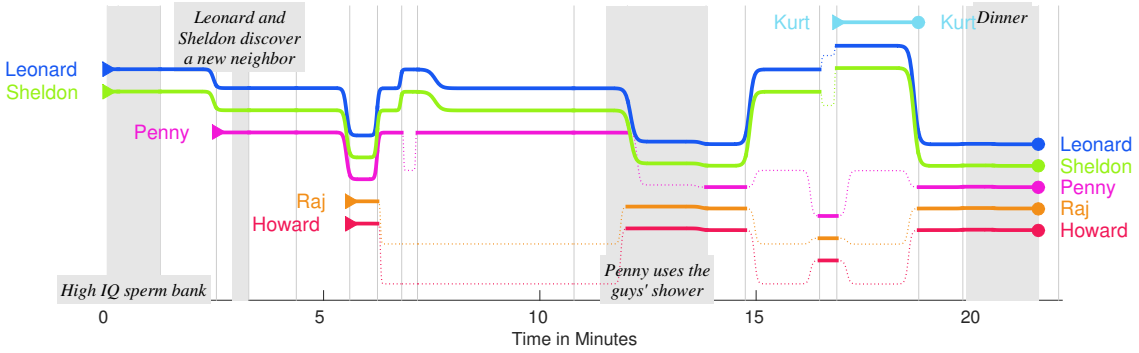


Figure 5.6: StoryGraph generated for season 1, episode 1 of the TV series *The Big Bang Theory*. In addition to the character lines, we see event bubbles that are added to the graph. Such event information enhances and augments the story representation power and graphs with events are used for the retrieval experiment.

We use SEMAFOR (Das et al., 2014), a method that analyzes the frame-semantic structure of English language. It is based on FrameNet (Baker et al., 1998), a lexical resource that groups words into a hierarchy of structured concepts. Apart from tagging predicates (words) with a concept, SEMAFOR also provides information about relationships to other predicates in the sentence. We manually select a subset of such tags that are related to story events – Becoming aware, Intentionally create, Emotion, Experiencer, Locative relation – that form the basis of our event bubbles. We present the frame-level parsing for a sentence from the plot synopsis in Fig. 5.7. Note how the "Becoming aware" tag helps find an important event (Leonard and Sheldon see Penny) in the episode (Fig. 5.6).

Once we obtain events from the plot synopsis, they are localized in the video using the plot to video alignment presented in Chapter 4. The event labels are then overlaid on the character line StoryGraphs so as to minimize visual clutter.

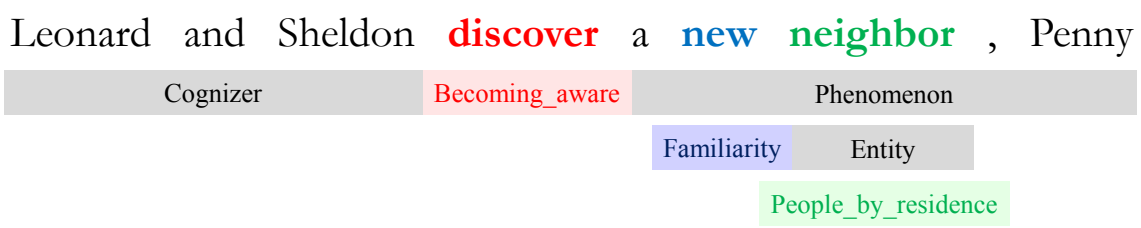


Figure 5.7: SEMAFOR (Das et al., 2014) parses a sentence into semantic frames, and provides information about the dependencies. We tag three predicates in this sentence. The first, "discover" is of type *Becoming aware* and is associated with a *Phenomenon* and a knowledge entity *Cognizer*. The word "new" depicts *Familiarity*, here about the *Entity* "neighbor". Finally, the word "neighbor" is tagged as belonging to the *People by residence* concept, and has no associations.

5.3.2 Retrieval experiment

We evaluate the use of StoryGraphs for obtaining a global picture of the content of the video via a human story event retrieval experiment. We hypothesize that given a StoryGraph, the time taken to find the event is reduced as opposed to searching in a TV episode without any meta information.

As the experiment hinges on timing information, it is influenced by multiple factors such as (i) the user may have watched (and remembers) the episode; (ii) concentration of the user on the website may not be constant throughout the experiment leading to different timings; and (iii) sheer luck could play an important role where the user skims to the correct part of the video, or enters an acceptable timestamp by chance.

We address the first problem by separating the performance for users that have watched the episode (and remember it) as compared to those who have not. The latter two are hard to model, however, we see that averaging the duration across several users is sufficient to overcome the impediments.

Our queries are randomly selected events in the story, and on average are about 1 query for 20 minutes of video. This corresponds to 1 query per episode of *The Big Bang Theory* (BBT) and 2 queries per episode of *Buffy the Vampire Slayer* (BF). We evaluate on a total of 18 queries ($6 + 12$), and present the results obtained from 9 users. The ground truth video clip duration corresponding to each query is about 80 seconds, and users have to provide a timestamp within this clip to be counted as correct retrieval. The average search region corresponding to each query is over 21 minutes.

Experimental setup. For a new user, we first randomly shuffle and store the queries. We familiarize the user with the concept of the StoryGraph, and ask him/her to find the relevant part of the video for every query. The user has to enter one timestamp indicating the time during which the event occurs, and is deemed to have found it correctly if it is within the start and end boundaries of the ground truth duration. We relax the search process, and provide an additional 10 second slack on either side.

The user is afforded three trials to find the correct timestamp, and information about each submission is logged to the database. We start with the simpler StoryGraphs corresponding to BBT and then move on to BF. We leave out GOT for this experiment

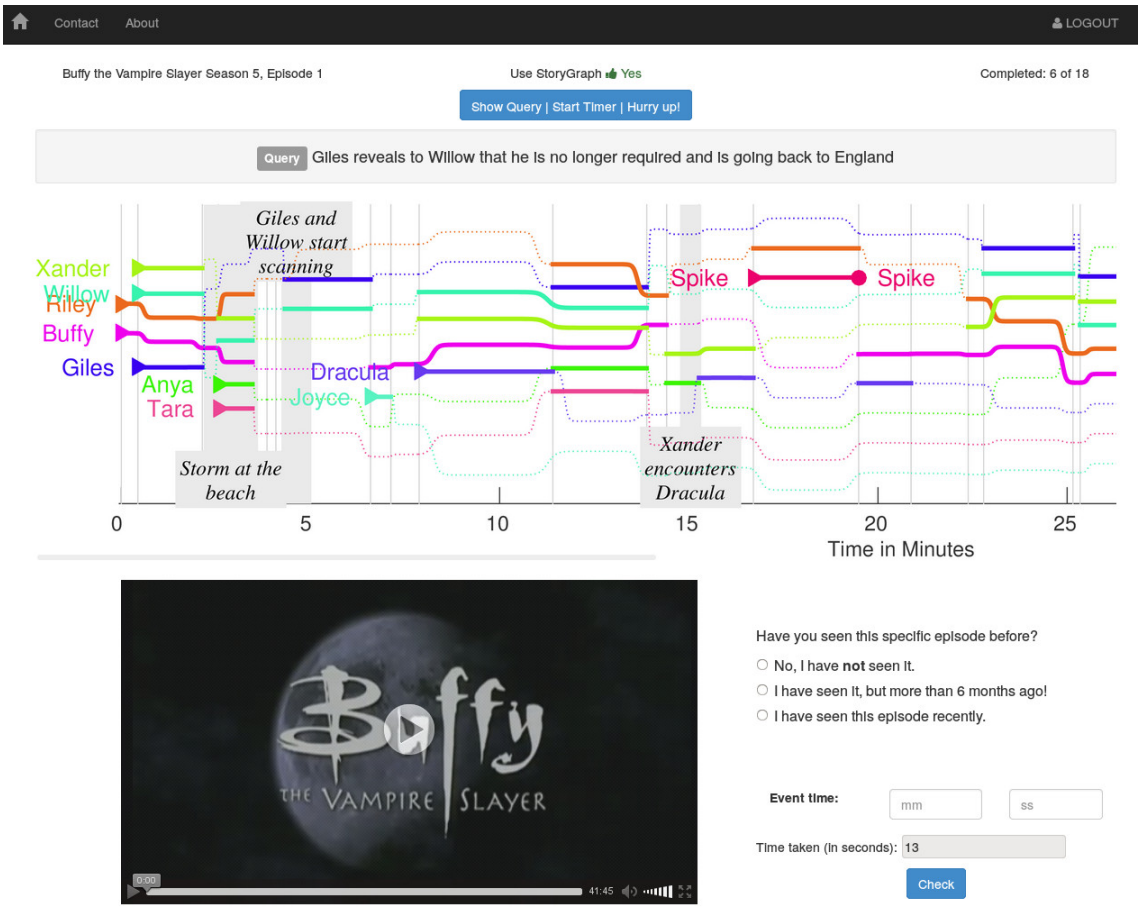


Figure 5.8: Web interface for the StoryGraphs retrieval experiment.

as solving those queries takes a large amount of time and reduces the concentration of our volunteers.

At the start of a timer, the user is presented a query, the corresponding StoryGraph augmented with event information, and the TV episode. We log the time taken by the user to find the relevant video clip and post the timestamp back to our server for evaluation. Note that we present the StoryGraphs for alternating queries, and owing to the initial random shuffling, each query is both solved with and without the StoryGraph by different users. This eliminates the bias created by the presence of easier or harder queries. Fig 5.8 presents a screen capture of the web interface built to conduct this experiment.

Results. We analyze the retrieval performance of our short experiment below. We observe that 5 of our 9 users are able to locate the correct timestamp for all queries within

	Seen video		Not seen		Total	
	✓	✗	✓	✗	✓	✗
User 1	131	–	86	56	91	56
User 2	59	70	104	114	89	99
User 3	–	–	60	88	60	88
User 4	118	124	–	–	118	124
User 5	32	80	172	335	125	250
User 6	31	39	50	125	43	96
User 7	50	112	144	346	120	268
User 8	89	81	77	279	87	147
User 9	–	–	80	91	80	91
Average	82	92	96	162	91	138

Table 5.3: Mean time taken (in seconds) for different users in our story event retrieval experiment with and without StoryGraphs. ✓indicates the user had access to the StoryGraph, and ✗not.

3 trials. The other users were unable to correctly localize about 1-3 queries, and we leave out such unsolved queries to simplify further analysis. We also note that while most queries are solved in the first trial, some users complained of ambiguity in the query text and thus required a few extra trials. Of the 155 correctly solved queries across all users, 27 took more than one attempt to localize the video.

For solved queries, we present the time taken to find the correct localization by different users in Table 5.3. We denote the events which are retrieved using StoryGraphs as ✓and those without as ✗. We also split the queries based on whether the user has seen the episode before (Seen video vs. Not seen). On average, and over all the users and queries, we see that StoryGraphs help users retrieve events in the video, reflected by a reduction in average time from 138 to 91 seconds. Of particular interest is the fact that the average time reduces strongly when the users have not seen the video (from 162 to 96), while also showing reasonable improvement in the case of videos that have been seen before.

While skimming through the video is a reasonable way to browse the video and search for events, video streaming can often be quite slow making it harder to skim efficiently. For example, User 7 takes 268s without StoryGraphs, while 120s with. Both these numbers stand out as compared to other users who did the experiment in house (with fast access to the videos).

User 1 presents a different picture, where he/she took much longer to find the queries with StoryGraphs. When asked, the user told us that he/she failed to understand the information visualized in the StoryGraphs, and thus spent more time looking for events within it. However, we believe this to be an anomaly as indicated the by the improved performance of other users.

Feedback. We received interesting user feedback about the presentation of the StoryGraphs. While most users found StoryGraphs useful, one user asked for the addition of a "legend" showing the names and faces of characters to help in the case of unseen videos. Another user mentioned that "Skimming through a video to search for a segment was only applicable when I knew the characters or the scene was set in a certain location: supermarket, bedroom, etc.". The other comments were to make the graph more interactive through web-based visualization techniques such as *D3.js*. We leave these as exploration for future work, but can safely conclude that StoryGraphs are not only a nice visualization, but can also aid users in speeding up search for story events in a video.

Chapter 6

Understanding Stories through Question-Answering

The work in this chapter is performed in collaboration with Prof. Sanja Fidler and Prof. Raquel Urtasun from the University of Toronto and Prof. Antonio Torralba from Massachusetts Institute of Technology. One of the answering method presented in this chapter (Neural similarity) was implemented by Yukun Zhu. The benchmark data set and the various answering schemes will be published in Tapaswi et al. (2016).

Among the most popular means to test understanding among humans is to examine whether a person can answer questions about the topic. As visual comprehension of AI matures and the tasks of detection, classification, and segmentation become commonplace, drawing higher levels of semantics from the images and videos is the next interesting problem.

Here, building upon advances in NLP and vision, answering questions about the content or context of images is becoming an increasingly popular task (Antol et al., 2015; Malinowski and Fritz, 2014; Yu et al., 2015). The type of questions asked are typically based on *what* and *where* are the objects in the image, *what* attributes do the objects exhibit (*e.g.* color, shape), pairwise relations between objects (*e.g.* spatial) in the scene, yes/no questions, and finally counting the number of items in the scene. While such questions verify the holistic nature of vision tasks, they are so far confined to a static image, and thus high-level semantics such as intentions and reasoning are mostly lost.

We believe that stories, in particular movies, are a perfect test bench to analyze reasoning as they require long-term temporal analysis of the content. Movies provide us with snapshots to people's lives that link into coherent stories. As with our previous work on aligning videos with text (Chapter 4) or automatically generating visualizations (Chapter 5), the questions in our data set are focused on the story conveyed in the movies and not facts or meta-data (*e.g.* actor names).

In this chapter, we introduce a novel data set, establish a competitive benchmark, and analyze several algorithms for question-answering in movies. We discuss the data set collection procedure and present some statistics and unique properties (Sec. 6.1) followed by an overview of answering methods. We make key contributions in the field of answering and propose a Neural similarity-based answering network (Sec. 6.2.4), and make significant modifications to end-to-end memory networks (MemN2N) (Sukhbaatar et al., 2015) that are necessary to make the model applicable to our task (Sec. 6.2.5).

A complete review about related work in the field of Question-Answering (QA), both textual and visual, was discussed in Sec. 2.5. In the review, we not only present examples of several data sets in both the modalities (text and images), but also attempt to provide a brief overview on the popular answering approaches. In this chapter, we will revisit the memory network architecture built for QA in Sec. 6.2.5 and propose key modifications to make it suitable for our problem.

6.1 Data set

We create a challenging benchmark data set for Question-Answering to further the field of semantic understanding of stories. Our data set facilitates answering using a variety of information sources: *plot synopses*, *videos*, and derivatives from the video such as *dialogs* (using subtitles as a proxy for perfect speech recognition), *scripts*, and *DVS* (Rohrbach et al., 2015). This places our data set in both language and vision communities.

Unlike other data sets (*e.g.* bAbI (Weston et al., 2015a), VQA (Antol et al., 2015)), the answers to our questions require natural language text and are typically longer than a single word making open-ended answering difficult to automatically evaluate. We thus resort to the paradigm of multiple-choice question-answering and provide for each question five competing answers, only one of which is correct. Automatic systems are encouraged to learn to select the correct answer option given the story in various forms.

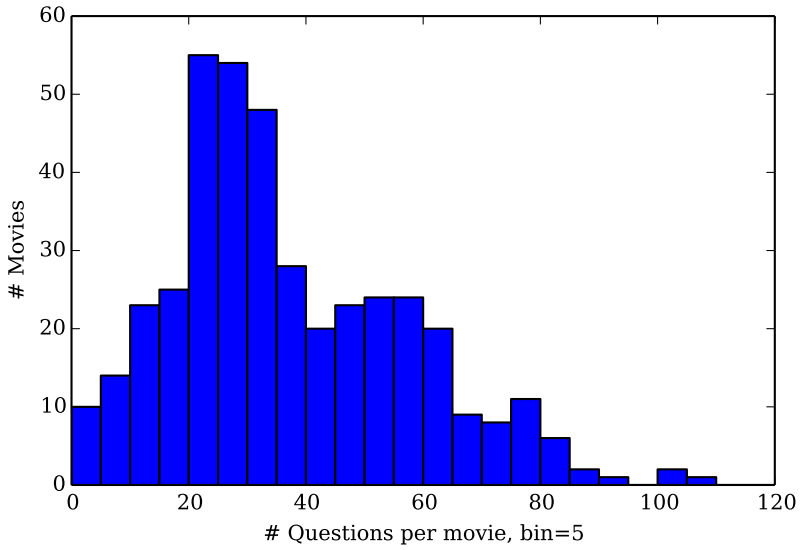


Figure 6.1: Distribution of the number of questions collected per movie.

We collected 14,944 questions from 408 movies, all of which come with Wikipedia plot synopses and subtitles. We obtain a variable number of questions for each movie plotted as a histogram in Fig. 6.1. We crawled *imsdb* for scripts, which are available for 199 (49%) of our movies, and use DVS transcriptions provided by Rohrbach et al. (2015) for 60 (15%) of the movies. Video clips are our final source of information to answer questions, and clips are available for 140 (34.3%) of the movies, and cover 43.2% of all questions.

We presented a short overview about various text sources in Sec. 3.2. Here, we present additional key characteristics about the different answering sources.

Plot synopses describe the content that is relevant to the story while not presenting details about the visual information. Thus, plots are what a perfect and automatic story analysis algorithm should understand by "watching" the movie. We use plots to gather questions about movies, speeding up collection and automatically restricting questions to story-related events.

Videos and subtitles. An average movie is about 2 hours in length and has over 198K frames with 2000 shots. While videos depict information about who did what to whom, they are not self-contained to explain why something happened. Dialogs play an important role, and we believe that together, both modalities can help fully understand the story.

DVS is a proxy for a perfect visual system and potentially allows quizzes to be answered without looking at videos. However, they are hard to obtain for all movies.

Scripts. Written by screenwriters, scripts serve as a guideline while making the movie. They contain both detailed scene and dialog descriptions including speaker information. However, they are not entirely faithful to the final movie as post-production may alter the final movie content.

6.1.1 QA collection strategy

Asking annotators to watch movies and come up with targeted questions about the story is a time-consuming, expensive, and hard to scale task. While people may volunteer questions about key aspects of movies they remember, this results in uncontrolled questions which are not localized in the video. We use plot synopses as a replacement for the video story and force annotators to ask story-relevant questions using the text from the plots. Using plots to collect questions also helps to localize the QA in the video by (manually or automatically) aligning plot sentences with the video. We split our data collection and annotation process into two parts to ensure high data quality.

Question and correct answer. In the first phase, our annotators are asked to select a movie from a large (> 1300) list of movies and are shown one paragraph of the plot synopsis at a time. For each paragraph, the annotator has the freedom to form any number and type of question. On average, we obtained 5.4 questions per paragraph. Along with the question, each annotator is also asked to provide the correct answer and select a minimal set of sentences from the plot paragraph that are sufficient to answer the question. The latter serves as our source of QA localization in the plot.

In our instructions for the annotators, we ask them to provide context for each question. QAs are obtained such that a person who has watched the movie should be able to answer the question without reading the plot synopsis. Generic questions such as "What are they talking about?" are discouraged. All annotators go through a short training phase and are paid by the hour (as opposed to per QA). This allows us to obtain thoughtful and complex QAs rather than short questions with single-word answers.

Multiple choices for questions. In the second phase of data collection, we harvest competing multiple-choice options for each question. Our annotators are shown the

Movie	Harry Potter and the Chamber of Secrets	Snatch.	Revolutionary Road
Question	What does Harry trick Lucius into doing?	Why does a robber tell Franky to buy a gun from Boris?	Why does April die?
Story		- When you get to London... - if you want a gun... - call this number. - Boris?	April dies in the hospital due to complications following the abortion.
Answer	Freeing Dobby	Because the robber and Boris want to steal the diamond from Franky	She performs an abortion on her own
Option 1	Releasing Dobby to Harry's care	He wants to hook him up	Due to injuries from an accident
Option 2	Releasing Dobby to Hagrid's care	He plans on robbing and killing him	She kills herself
Option 3	Releasing Dobby to Dumbledore's care	Because otherwise Boris would kill him	Due to a drug overdose
Option 4	Admitting he gave Tom Riddle's diary to Ginny	The robber plans to steal a painting from Franky	She is shot

Movie	Indiana Jones and the Last Crusade	The Lord of the Rings: The Return of the King	Star Wars: Episode III - Revenge of the Sith
Question	What does Indy do to the grave robbers in the beginning of the movie?	Who sees Denethor trying to kill himself and Faramir on a bonfire?	What does Palpatine reveal to Anakin?
Story		- Gandalf! - Gandalf! - Denethor has lost his mind! - He's burning Faramir alive!	Palpatine entices Anakin with knowledge of the dark side of the Force, including the power to "cheat death". When Palpatine reveals himself as the Sith Lord Darth Sidious, Anakin reports his treachery to Mace Windu, ...
Answer	He steals their golden crucifix	Pippin	His knowledge of the dark side of the Force, including the power to cheat death
Option 1	He kills them while they're sleeping	Aragorn	That he is Darth Vader
Option 2	He tells the Boy Scouts to beat up the grave robbers	Gandalf	How to kill the Jedi Master
Option 3	He steals their horses	Eowyn	Where Padme is
Option 4	He calls a museum and tells them that he found people that stole the golden crucifix	Sam	That the Jedi Council favors Anakin

Table 6.1: Example QAs from our a selection of movies from our data set. Three answering sources – videos, subtitles, and plots – are shown here.

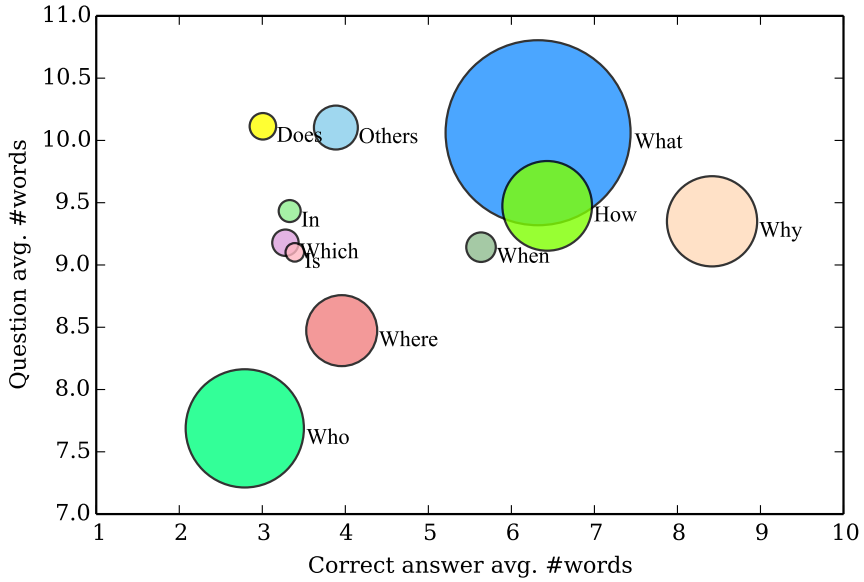
same paragraph and the question, but not the correct answer. They are asked to answer the question correctly and provide 4 wrong answers framed using deceiving facts from the same paragraph or common-sense answers (*e.g.* a "Who" question consists of other names from the same movie as competing options). The annotators are also given the freedom to rephrase questions, thus providing an automatic sanity check for all questions gathered in the first step. Finally, the correct answers from both phases are tallied to verify the quality of the QA. Table 6.1 presents examples from our data set, with three story sources – video shots, plots, and subtitles. Note how the multiple-choice answers are genuinely confusing, especially in absence of the story.

Localization in video. Along with collection of QAs, we ask an in-house annotator to align each sentence of the plot synopsis with the movie. We ask the annotator to mark the begin and end (in seconds) of the video that the sentence describes. Movie plot sentences can be fairly long and complicated, and thus are often aligned to multiple non-consecutive video clips. For each question, we are given the set of plot synopsis sentences, and using the text-video ground truth alignment, we obtain video clips that are suitable to answer the question. For the benchmark, we release the video clips in contrast to providing the entire video which has copyright problems.

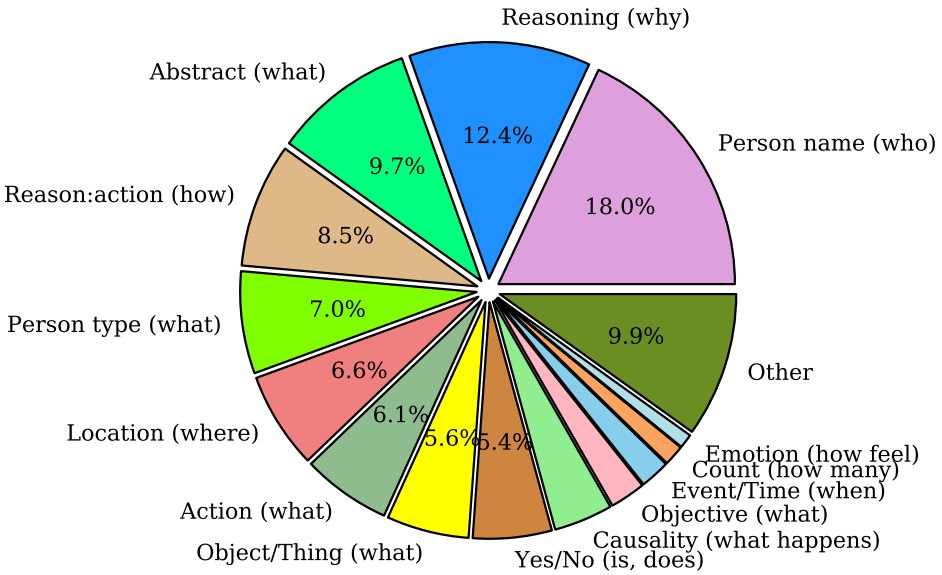
6.1.2 Statistics

We present some statistics about the questions and answers collected in the MovieQA data set.

Data set comparison. First, we compare the goal and numbers of our data set against other recent QA data sets for both text and video. In Sec. 2.5, we presented several data sets that are being used popularly, and we compare to them in Table 6.3. MCTest (Richardson et al., 2013) tests machine reading comprehension on children stories. bAbI (Weston et al., 2015a) is a set of discrete toy tasks that test different aspects of reasoning. CNN+DailyMail (Hermann et al., 2015) operates on news articles and their summaries, and tests information abstraction. DAQUAR (Malinowski and Fritz, 2014), Visual Madlibs (Yu et al., 2015), and VQA (Antol et al., 2015) all test image-based reasoning, asking questions about counting, objects, their relations, colors, *etc*. and are built upon vi-



(a) The length of questions and answers in our MovieQA data set is quite long (as compared to other data sets) making our data set challenging. We plot the average number of words in the question against that in the correct answer. Each bubble represents the data point corresponding to the first word used in the question (*e.g.* What, Why, Who) and the area of the bubble indicates the number of questions starting with that word.



(b) The first word of the question is an insufficient indicator of the type of expected answer. For example, "What is the name of" starts with "What", but corresponds to a "Who" like question. For a subset of our questions, we annotate the type of the answer, and show the distribution of questions depending on their answer type as a pie chart.

Figure 6.2: Statistics about the types and lengths of QAs in our MovieQA data set.

	TRAIN	VAL	TEST	TOTAL
Movies with Plots and Subtitles				
#Movies	269	56	83	408
#QA	9848	1958	3138	14944
Q #words	9.3	9.3	9.5	9.3 ± 3.5
CA. #words	5.7	5.4	5.4	5.6 ± 4.1
WA. #words	5.2	5.0	5.1	5.1 ± 3.9
Movies with Video Clips				
#Movies	93	21	26	140
#QA	4318	886	1258	6462
#Video clips	4385	1098	1288	6771
Mean clip dur. (s)	201.0	198.5	211.4	202.7 ± 216.2
Mean QA #shots	45.6	49.0	46.6	46.3 ± 57.1

Table 6.2: MovieQA stats. Our dataset has several text sources (plots and subtitles are presented above), as well as video clips (bottom). The two types of answering modes are text and video. Video-based answering supports 140 movies which have manually annotated video to plot alignments, and each question looks at 3.5 minutes of video to answer it. We present total number of questions, and some statistics about video durations with standard deviation in the TOTAL column.

	Txt	Img	Vid	Data source	AType	#Q	AW
MCTest	✓	-	-	Children stories	MC (4)	2640	3.40
bAbI	✓	-	-	Synthetic	Open/Word	20×2,000	1.0
CNN+DM	✓	-	-	News articles	Word	1,000,000*	1*
DAQUAR	-	✓	-	NYU-RGBD	Word/List	12,468	1.15
V. Madlibs	-	✓	-	COCO+Prompts	FITB/MC (4)	2×75,208*	2.59
VQA (v1)	-	✓	-	COCO+Abstract	Open/MC (18)	764,163	1.24
MovieQA	✓	✓	✓	Movie stories	MC (5)	14,944	5.29

Table 6.3: A comparison of various QA data sets. First three columns depict the modality in which the story is presented. The different answering types (AType) are MC (N): N multiple choice options; Word/List: single word, or word list as answer; Open: open ended answering (a word from the vocabulary) FITB: fill in the blanks. * represents estimated information. AW is the average number of words in the answer.

sion data sets such as NYU-RGBD (Silberman et al., 2012), MS-COCO Lin et al. (2014b), and Abstract Scenes (Zitnick and Parikh, 2013).

Our data set is the first to introduce temporal reasoning, and operates (similar to MCTest) at the highest semantic level of the story. The 14,944 QAs are sourced from 408 movies

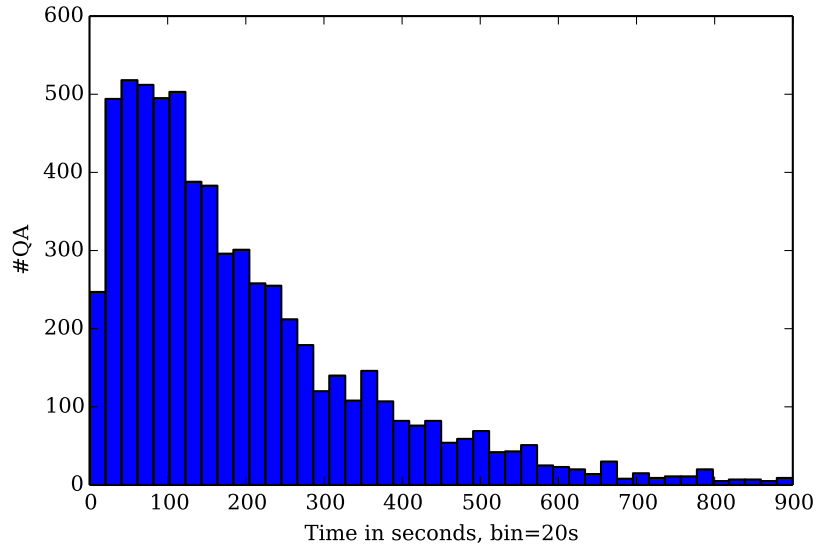


Figure 6.3: Duration of video used to answer questions.

that are arguably more complicated than images or text passages, and have long natural language answers (requiring a separate analysis). Table 6.2 presents the *train*, *val* and *test* splits of the benchmark for various answering sources: plots and dialogs (subtitles) and video clips.

Answer types. We analyze collected QAs in two ways. Based on the first word of the questions, we show in Fig. 6.2(a) the average number of words in the questions and correct answer for different types of questions. Note how questions beginning with "Why" require on average long answers with over 8 words. Also note that the answer choices for Yes/No questions (*e.g.* "Does") often contain a short reason (*e.g.* "No, they left him behind.") allowing the opportunity to create 4 misleading options.

A different approach to analyze questions is based on what the answer represents. We manually annotate a subset of the questions with regards to the answer type. As seen in Fig. 6.2(b), a large part of the QAs are based on high-level reasoning questions (Reasoning, Abstract, Reason:action), while other classical vision tasks (Person name - identification, Location - scene recognition, Action - action recognition, Counting, Emotion, *etc.*) also show have significant number of questions.

Text type	# Movies	#sent. / movie	avg. length
Plot	408	35.2	20.3
Subtitle	408	1558.3	6.2
Script	199	2876.8	8.3
DVS	60	636.3	9.3

Table 6.4: Statistics of the various text sources used for answering.

Story sources. Table 6.4 presents information about the variety of text sources that can be used to answer questions. Note how the number of words in a plot synopsis is higher than all others, making it a fairly complex, yet concise data source. On the other hand, scripts, DVS, and subtitles have a large number of sentences hinting that attention-based answering (as in memory networks (Sukhbaatar et al., 2015)) is important.

Our video-based answering benchmark consists of 6,771 video clips associated with 6,462 questions. Each clip stems corresponds to a sentence from the plot synopsis that was used to create the question. We create short non-overlapping video clips, and each QA is associated with 2.66 clips on average. Fig. 6.3 presents a distribution of the number of questions and the time in video with which they are associated. Even though we restrict the maximum video clip duration to 8 minutes, QAs that are associated with more than one clip may require the machine to "watch" a long duration before answering.

6.2 Answering models

We investigate a number of intelligent baselines for QA ranging from very simple cosine similarity computations to more complex neural architectures built specifically for the problem of QA.

Formally, let S denote the representation of the story, which can take the form of any of the available information sources – plots, video shots, subtitles, scripts or DVS. Each story S is associated with a bunch of questions, that the answering system sees one at a time. Each question q^S is correctly answered by selecting one of multiple answer options $\{a_j^S\}_{j=1}^M$, where $M = 5$ for our data set.

While open-ended question-answering should be the real goal, evaluating correctness of answers – especially when the answers are more than a few words – is very difficult. Thus, currently we resort to the problem of multiple-choice QA, and plan to include

open-ended answering for non-reasoning type questions at a later stage in time. However, multiple-choice QA converts the problem of "answer synthesis" to "answer analysis" and answering a question boils down to evaluating a three-way scoring function $f(S, q^S, a^S)$. The function f calculates the likelihood for each option being the correct answer by selecting (attending to) relevant information from the story through the question. Thus, finding the correct answer is achieved by

$$j^* = \underset{j}{\operatorname{argmax}} f(S, q^S, a_j^S). \tag{6.1}$$

Different answering techniques can be casted from this mold. Even classical CNN-RNN question-answering pipelines (Malinowski et al., 2015) are essentially a function that takes the story – image input through CNN, the question – as words fed to an LSTM, and chooses a word as the correct answer among the list of vocabulary possibly through a softmax.

For the remainder of this chapter, we drop the superscript $(\cdot)^S$ for notational brevity, and when q and S appear together, it is implied that the question q is based on the story S.

6.2.1 Hasty machine

Our first attempt at using a machine to answer questions from our MovieQA data set discovers hidden biases within the data. We consider a family of functions that ignore the story and attempt to answer the question directly. We call such a baseline the "hasty machine" as it does not bother to read/watch the actual story.

Random answering is the extreme and trivial case of a hasty machine that selects answers without looking at any of the story, question or answers.

One possible approach to hasty answering is to select answers based on the length of the answer. Here,

$$f(S, q, a_j) = g_{HM_1}(a_j), \tag{6.2}$$

where g_{HM_1} captures answer length in terms of number of words, and the correct answer is chosen as the *longest*, *shortest* or *most different* answer.

Another possible interpretation of the hasty machine is to analyze the vector representations of our natural language answer options. For example, it is likely that the correct answer is *closest* or *farthest* away in vector distance from all other answers. We attempt

to select the correct answer through the use of different text representations (as will be explained in Sec. 6.2.6).

A hasty machine is not shown the story, however, may read the question. We now explore a situation which tests whether something can be learned about the nature of questions and answers. For example, if we ask a "Where" question, we expect that the answer is a place. If the multiple choices are not all places, the question could be answered without looking at the story and just by learning common sense knowledge.

$$f(S,q,a_j) = g_{HM_2}(q,a_j), \tag{6.3}$$

where g_{HM_2} computes similarities between the question and the answer options using the different representations, and the most *similar* answer option is put forward as the correct answer.

6.2.2 Hasty turker

While the hasty machine is an example of AI finding hidden biases within the data set, we perform another experiment in which we ask humans to answer questions. We select a random subset of our QA data without the story, and post the multiple-choice questions to Amazon Mechanical Turk (AMT) (Buhrmester et al., 2011). Crowd-sourcing through such websites has massively helped annotate and create large data sets. Here, we show that when not shown the story, even humans are unable to perform well on our QA data set. This is in sharp contrast to other data sets, where common sense knowledge may help answer questions about the content in the image without looking at the image (*e.g.* What color is the football?).

We expect that some users may have watched the movie and remember a part of the story (something not easily controlled), and thus remove questions with names and places that might give away the story (*e.g.* Darth Vader) as compared to generic names (*e.g.* Bob). On this reduced set, we see performance even closer to random, thus validating the difficulty of our deceiving multiple choice options.

6.2.3 Cosine similarity

We consider a simple model that combines the information from the story and tries to select the correct answer for a question given the set of multiple choices. In particular, we factorize the three-way scoring function into two functions, computing the similarity between the story and the question, and the story and a answer option.

$$f(S,q,a_j) = g_{CS}(S,q) + g_{CS}(S,a_j), \tag{6.4}$$

where g_{CS} computes the cosine similarity between vector representations (presented in Sec. 6.2.6). The intuition for the scoring function is to "search" for the relevant parts of the story using the question and answer options. When using plot synopses there are often one or two sentences that are most relevant to a question, and scoring them against the list of answers is likely to provide the correct answer.

Only analyzing one sentence (or video shot) is too short (especially for scripts or dialogs) to answer complex questions. A simple approach to mend this problem is to use a sliding window of length H over the stories. Using a window simulates searching for a high similarity match in a span of multiple sentences (or shots) of a story.

$$f(S,q,a_j) = \max_{l} \sum_{k=l}^{l+H} < \phi(s_k), \phi(q) > + < \phi(s_k), \phi(a_j) > . \tag{6.5}$$

s_k denotes a sentence (or shot) from the story S and $< \cdot, \cdot >$ computes the inner product between $\phi(\cdot)$, the vector representations of sentences (or shots), questions and answers. The optimal window size is determined using training data. The correct answer option is determined as in Eq. 6.1.

6.2.4 Neural similarity

The Cosine similarity based model breaks the scoring function into smaller parts operating separately on the story and question, and story and answer (see Eq. 6.4). Instead of factoring $f(S,q,a_j)$ as a decomposition of two unweighted functions, we build a neural network that can learn complex similarity functions.

For a story of length n (*i.e.* n sentences or shots), the cosine similarity functions $g_{CS}(S,q)$ and $g_{CS}(S,a_j)$ can be seen as two vectors of length n, where the k^{th} entry holds the

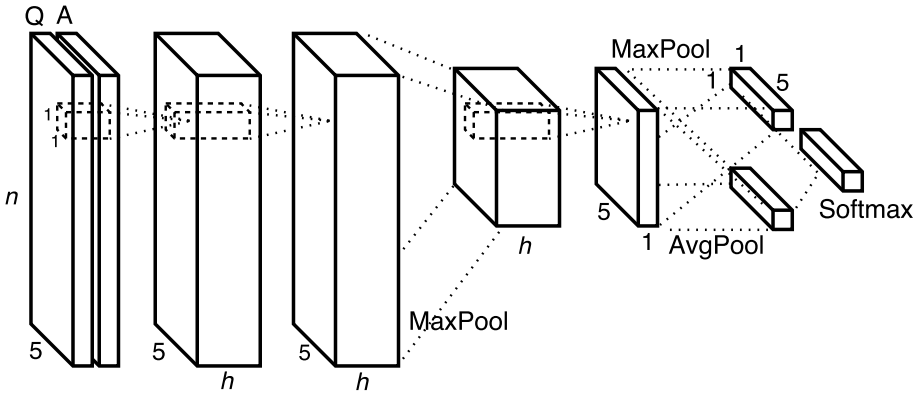


Figure 6.4: Our neural similarity model to answer questions by computing the three-way scoring function and applying a softmax layer at the top. h is the number of hidden layers, n is the size of the story, and $M = 5$ represents the number of answer options.

similarity $< \phi(s_k), \phi(q) >$ and $< \phi(s_k), \phi(a_j) >$ respectively. We concatenate the similarity vectors for all five answers $[g_{CS}(S, a_j)]$ to obtain a $n \times 5$ matrix. We also replicate $g_{CS}(S, q)$ (originally $n \times 1$) to obtain a $n \times 5$ matrix, and stack the question and answer similarity matrices to obtain an $n \times 5 \times 2$ tensor. Fig. 6.4 (layer 1) depicts such a layer.

Our neural similarity model is a Convolutional Neural Network, that performs several layers of 1×1 convolutions to learn a function $\psi(g_{CS}(S, q), g_{CS}(S, a))$. Note that the first convolutional computation corresponds to a weighted version of the cosine similarity model (Eq. 6.4) with $h = 10$ different weight options (number of filters). We add another layer of 1×1 convolutions, followed by a max pooling layer of kernel size 3 that allows to score the similarity within a window in the story and results in a tensor of size $n/3 \times 5 \times h$. We perform another convolution, that generates a $n/3 \times 5$ output, to which we apply both mean and max pooling across the storyline ($n/3$). We add these scores and select the final answer choice using softmax. The network is trained using the cross-entropy loss and Adam optimizer (Kingma and Ba, 2015). Our network is initialized with inner products rather than actual vector representations of the question, answers and story.

We thank Yukun Zhu for implementing the Neural similarity model.

6.2.5 Memory network

Memory networks are neural architectures originally proposed for question-answering (Sukhbaatar et al., 2015; Weston et al., 2015b). With regards to our scoring function, memory networks model complex three-way relationships between the story and the answer.

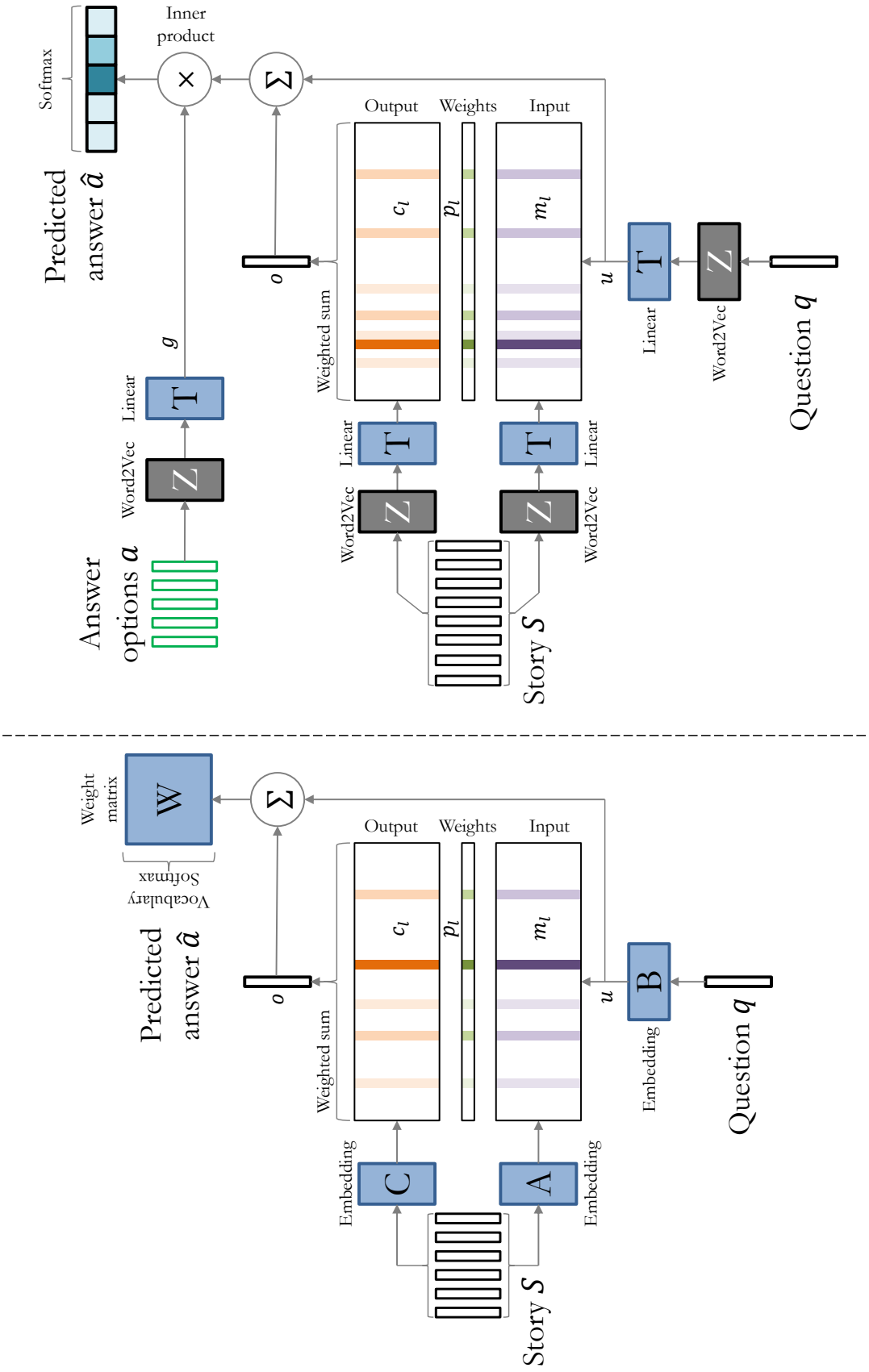


Figure 6.5: Memory Network architectures: Trainable parameters are shown as light blue boxes. (Left) One layer of the end-to-end Memory Network (Sukhbaatar et al., 2015) used for QA on the bAbI data set (Weston et al., 2015a). (Right) One layer of the proposed memory network that can handle natural language answers and large vocabulary. See Sec. 6.2.5 for a discussion.

End-to-End Memory Networks (MemN2N). We briefly describe MemN2N proposed by Sukhbaatar et al. (2015) and suggest extensions to make it suitable for our data and task. The MemN2N is built to provide single-word answers by picking the most likely word from the vocabulary. Designed and evaluated on the bAbI data set (Weston et al., 2015a), the vocabulary sizes are often quite small (20-40 words) enabling end-to-end training of the different embedding layers of the network. We present a glimpse of one layer of the architecture (input, layer, output) in Fig. 6.5 (left).

A question q is encoded using a word embedding matrix $B \in \mathbb{R}^{d \times |\mathcal{V}|}$ to obtain a vector u, where $\mathcal{V}$ is the vocabulary and d is the embedding dimension. Bag-of-words (average) or positional encoding (weighted average) schemes are used to pool word representations of the question into a vector. Simultaneously, the sentences of the story s_l are encoded using input and output word embeddings A and C respectively, and the representations m_l and c_l are stored in a "memory bank". The question u, in conjunction with the input-side story bank is used to produce an attention-like mechanism that selects sentences that are most likely to contain the answer to the question

$$p_l = \text{softmax}(u^T m_l). \tag{6.6}$$

The probability p_l is used to weight the output-side story representation c_l to obtain $o = \sum_l p_l c_l$, an importance weighted representation of the story sentences that emphasize picking the answer. Finally, a linear projection $W \in \mathbb{R}^{\mathcal{V} \times d}$ decodes the question and output to provide a soft score for each word

$$\hat{a} = \text{softmax}(W(o + u)), \tag{6.7}$$

and the answer is chosen as the top-scoring word.

As the entire network, including word representations are trained from scratch, it is truly an end-to-end network. However, single-word answering and small vocabulary size are two key problems that we encounter while using the model on our task.

MemN2N for natural language answers. To allow the memory network to rank multiple choice answers written in natural language (as sentences), we can add an additional embedding layer F that maps each answer option a_j to a vector g_j. Note that F is similar to the original word embeddings B, A, C and operates on the sentences from answer options instead of question or the story. The correct answer is now predicted by

computing the similarity between the answer representations g and the question u and output o as

$$j^* = \operatorname{argmax}_j (o+u)^T g_j \,. \tag{6.8}$$

We implement the answer selection in the network through a softmax on the M scores, and train the model using the cross-entropy loss.

In our general QA formulation, the inclusion of the answer options makes the modified memory network equivalent to

$$f(S,q,a_j) = g_{M1}(S,q,a_j) + g_{M2}(q,a_j) \,. \tag{6.9}$$

The former g_{M1} uses the question to draw attention to the story, and is combined with the answer to create a score, while the latter g_{M2} scores question and answer similarity directly.

Weight sharing and fixed word embeddings. The original MemN2N learns word embeddings directly for the task of question-answering. However, it is not feasible to scale this to large vocabulary data sets such as ours. For example, training a model with a vocabulary size of 12,000 words (obtained from plot synopses alone) and d =100 would require learning 1.2M parameters for each embedding, an infeasible task given training data only in the tens of thousands.

One way to prevent heavy overfitting (we tried using the original MemN2N and while it was able to improve training accuracy, validation and test accuracy were stuck at 20%, random performance) is to share all the word embeddings B, A, C, F of the network. However, this is still insufficient as one embedding itself is very large.

We thus drop the trainable embedding layers B, A, C, F and replace them by a fixed (pre-trained) word embedding $Z \in \mathbb{R}^{d_1 \times |\mathcal{V}|}$ obtained from the Word2Vec (Mikolov et al., 2013) model. Further, we introduce a shared linear projection $T \in \mathbb{R}^{d_2 \times d_1}$ to suitably modify the word embeddings and map the sentences (questions, words, and answers) into a common space. Here d_1 represents the Word2Vec dimension and d_2 is the projection dimension. In short, the new vector representations are

$$u = T \cdot Zq, \quad m_l = T \cdot Zs_l, \quad c_l = T \cdot Zs_l, \quad g_j = T \cdot Za_j \,. \tag{6.10}$$

We initialize the shared linear projection T either as an identity matrix $d_1 \times d_1$ or use PCA to reduce the dimensionality from $d_1 = 300$ to $d_2 = 100$. Fig. 6.5 (right) presents the network architecture after our proposed modifications to the original MemN2N (left). Note how the trainable parameters T are all tied with each other, and word embedding is performed using a fixed pre-trained Word2Vec representation Z.

6.2.6 Representations for text and video

We now discuss vector representations for text sentences and video shots. These have been employed in the hasty machine, cosine similarity, and the neural similarity models. Note that the modified MemN2N uses Word2Vec representations.

Term Frequency · Inverse Document Frequency (TF·IDF) is a popular and successful feature in information retrieval. We first preprocess all words to lower case, use stemming (Porter, 2001), and compute a vocabulary $\mathcal{V}$ that consists of all words in the document. We treat plot synopses (or other story sources) as documents, and compute the term frequency and inverse document frequency weights for each word. We represent each sentence (story, question or answer) in a bag-of-words scheme, using the TF·IDF score to weight each word. In particular, we adopt the logarithmic forms

$$\mathrm{TF}(t,d) = 1 + \log_{10}(n(t,d)) \tag{6.11}$$

$$\mathrm{IDF}(t) = \log_{10}\left(\frac{|D|}{\sum_d \mathbb{1}(t \in d)}\right) \tag{6.12}$$

where $d \in D$ is a document, $t \in \mathcal{V}$ a term (word), and $n(t,d)$ counts the number of instances of the word t appearing in document d.

Word2Vec. A drawback of TF·IDF is it's inability to capture similarities between words of the same meaning. We use the Word2Vec skip-gram model proposed by Mikolov et al. (2013) and train it on about 1300 movie plots to obtain $d = 300$ dimensional, domain-specific (related to movies) word embeddings. Each sentence is represented by mean pooling the word embeddings of individual words appearing in the sentence. The resulting vector is normalized to have unit norm.

Word2Vec representations are used as the primary vector representation in our modified memory network. Using a TF·IDF representation is very similar to the original MemN2N (with additional word weights) and demonstrates problems with overfitting.

SkipThoughts. Mean pooling the word embeddings to represent a sentence destroys the word order. To overcome this, we use SkipThoughts (Kiros et al., 2015b) a Recurrent Neural Network representation that aims to capture underlying sentence semantics. In particular, we compute $d = 4800$ dimensional sentence representations with the pre-trained model provided by Kiros et al. (2015b), and normalize them to have a unit norm. As we will see in the evaluation, the ability to represent semantics of a sentence increases as we move from TF·IDF to SkipThoughts. However, this adversely affects performance as answering a question is often about fine-grained differentiation, *e.g.* between names of characters, that is hard to perform using SkipThoughts.

Video embeddings. We learn a joint embedding between shots and sentences that map the two modalities into a common space. In such a space, scoring similarity between a text sentence and video shot is performed by simply computing a dot product. This allows us to leverage all the answering techniques in their original forms, by treating the video shots as just another representation of the story in the same space as the text.

We learn the joint embeddings by following Zhu et al. (2015) who extended the concept of Kiros et al. (2015a) to videos. Specifically, a shot is represented by averaging the features extracted from every 5th frame from the hybrid-CNN (trained on places and objects) (Zhou et al., 2014) and GoogLeNet (trained on object classes) (Szegedy et al., 2015) architectures. Simultaneously, the sentences are encoded using a Long-Short Term Memory RNN. The embedding is a linear mapping of the shot and sentence representations, and is trained on the Movie Description data set (Rohrbach et al., 2015) to reduce the dot product between matching video and textual descriptions using the ranking loss.

6.3 Evaluation

We present results for question-answering using several proposed methods that range from exploring the data set bias to novel neural architectures that compute three-way similarity functions to adapting the memory networks specifically built for question-answering.

Experimental setup. We have two primary modalities to perform answering: (i) text – where the story is presented as a plot synopsis, subtitle, script, or DVS; and (ii) video – where we are allowed to use video clips associated with individual QAs or all the clips from the entire movie, optionally along with dialogs (subtitles) to perform answering.

Data set structure. The data set is divided into three disjoint splits: *train*, *val*, and *test* based on uniqueness of movie titles in each split. The three splits are optimized to maintain ratios between the number of QAs, number of movies, and even the answering sources to be 10:2:3. For example, the *train* split has 269 movies with 9,848 QAs, *val* contains 56 movies with 1,958 QAs and the *test* set is made up of 83 movies with 3,138 QAs. Detailed statistics for each split were presented in Table 6.2.

As part of the benchmark, only the *train* set can be used for training automatic answering models and tuning hyperparameters. The *val* set should not be touched during training, and is used to report results for several models. The *test* set is a held-out set (ground-truth not provided) that is evaluated on our MovieQA server. The performance on this set will be reported on a public leaderboard. As the evaluation server is currently under construction, for the purpose of this thesis, we report results on the *val* set.

QA evaluation metric. Multiple choice QA leads to a simple and objective evaluation – a desirable property and motivating factor during data set planning and collection. We measure accuracy, the fraction of correctly answered questions over the total number of questions. Note that, for a question with 5 multiple choices, a randomly chosen answer has a 20% chance of being correct.

6.3.1 Hasty machine

Our first answering scheme, the *hasty machine* (Sec. 6.2.1) ignores the story and attempts to answer the question based on internal data set biases. We evaluate several means to answer the question without looking at the story. Table 6.5 presents results for such methods.

Our first option is to answer based only on the answer length. Note how long answers tend to be correct more often (25.5%) as it can be quite hard to come up with competitive wrong answers. Our second alternative exploits within answer similarity or distinctiveness. Here, we represent the answer options using the three descriptors – TF·IDF, Word2Vec

Answer length		longest	shortest	different
		25.5	15.2	21.2
Within answers		TF·IDF	SkipThoughts	Word2Vec
	similar	21.7	**27.1**	25.8
	distinct	20.9	15.3	15.2
Question-answer		TF·IDF	SkipThoughts	Word2Vec
	similar	12.4	19.3	**25.0**

Table 6.5: Accuracy for methods from the "Hasty Machine" which tries to answer questions without looking at the story.

and SkipThoughts. All of them show mediocre performance, with the answer most similar to the others trending towards slightly higher performance. Quite unexpectedly, SkipThoughts demonstrate good performance at 27.1% while choosing the most similar answer. Our final option chooses an answer that is most similar (in vector space) to the question. Here again, we see that all methods are unable to show promising performance.

Overall, we can conclude that the data set is highly competitive, especially in terms of the confusing multiple choices and shows few weak spots for easy attack.

6.3.2 Hasty turker

In this experiment, we test the deceiving nature of our multiple choices by asking humans to answer questions without providing access to the story. We carve a subset of 200 QAs, and ask humans on Amazon Mechanical Turk to pick the best possible answer. Each question is asked to 10 turkers, and to encourage them to answer correctly, we reward the turkers when they select an answer that agrees with the majority.

As evaluation criteria, we evaluate and average the annotations of all turkers to obtain *overall accuracy* for the experiment. We also compute *accuracy majority* by choosing the answer predicted by the majority to be the correct one. Finally, we count the number of questions for which the correct answer is never chosen *always wrong*.

Table 6.6 presents the performance for the first choice of 200 QAs. We observe that the majority is able to do quite well (37.0%), and this is probably due to unique names that appear in the questions. For example, even if we do not mention the movie name, asking a question like "Who kills Dumbledore?" gives away the movie for those who have seen

	Overall accuracy	Majority accuracy	Always wrong
200 QAs	27.6	37.0	12.0
135 QAs (no famous names)	24.7	30.4	14.1

Table 6.6: Accuracy for methods from the "Hasty Turker" experiment, where we ask humans to answer questions without looking at the story. We evaluate on a subset of QAs (200), and redo the experiment after removing famous names that might give away the movie.

it. This makes answering such questions easier as it is impossible to separate the turker and his/her prior knowledge about movies that he has seen.

Towards this goal, we discard questions which give away names as blatantly as above (*e.g. Darth Vader*, *Indiana Jones*, *etc.*) and ask 10 additional turkers to answer questions in this restricted set of 135 QAs. From the Table 6.6 we see that the majority accuracy shows a 7% drop (from 37.0% to 30.4%) indicating that it is now much harder to decipher the name of the movie. An interesting point to note is that the turkers were unable to unravel that long answers are more likely to be correct.

6.3.3 Text-based answering

We present a detailed evaluation of using text-based stories: plots, subtitles, scripts, and DVS; along with various representations: TF·IDF, Word2Vec, and SkipThought. We evaluate methods that use the story to answer the question: Cosine Similarity (Sec. 6.2.3), Neural similarity (Sec. 6.2.4), and the modified Memory Network (Sec. 6.2.5).

We present the results for all methods in Table 6.7. Overall, we observe that the plot synopsis is among the best performing story types, especially in the case of bag-of-words like representations as the QAs are collected using plots, and the annotators often reproduce words from the plot verbatim.

Cosine similarity. The first section of Table 6.7 presents results for the proposed Cosine similarity model. The results of different representations are shown as separate rows. SkipThoughts show bad performance across all answering sources, and are in fact close to random except in the case of plot synopses. We suspect that while SkipThoughts are good at capturing the overall semantics of a sentence, proper nouns (such as names and places) are often quite hard to distinguish. Word2Vec and TF·IDF show similar performance, also peaking at the plot synopsis. The window length is the only parameter

#	Method (parameters)	Plot	DVS	Subtitle	Script
	Cosine similarity				
1	Cosine TFIDF	47.6	24.5	24.5	24.6
2	Cosine SkipThoughts	31.0	19.9	21.3	21.2
3	Cosine Word2Vec	46.4	26.6	24.5	23.4
	Neural similarity				
4	NeuralSim TFIDF	48.5	24.5	27.6	26.1
5	NeuralSim SkipThoughts	28.3	24.5	20.8	21.0
6	NeuralSim Word2Vec	45.1	24.8	24.8	25.0
7	NeuralSim Fusion	**56.7**	24.8	27.7	28.7
	Modified memory network				
8	lin proj, 1 layer, shared	40.6	33.0	**38.0**	42.3
9	PCA 100d, 1 layer, shared	37.0	31.9	35.4	40.0
10	lin proj, 3 layer, shared	42.3	33.0	37.1	**43.0**
11	(row 8) with PosEnc.	40.3	**34.0**	**38.0**	39.7

Table 6.7: QA accuracy using Text-based information sources. We present results for the Cosine similarity in the first section, Neural similarity model in the second, and analyze the modified Memory Network in the last. Numbers in bold are the best performance across all methods, and those underlined are the top performers within each approach.

in this model and is optimized using the *train* set. For plots, this is $H = 1$ since most QAs are based on a single plot sentence. Note that plot sentences are much longer on average (*cf*. Table 6.4). The table reports numbers for the *val* set.

To better understand different feature representations, we split the results for cosine similarity with plot synopsis based on the first word of the questions. Fig. 6.6 presents accuracy for 7 most frequent question words, from which we can draw several insights. Firstly, TF·IDF and Word2Vec perform quite well. We suspect that Word2Vec performs best on "Where" questions, as the correct place name is often referenced by synonyms that are close together in the semantic Word2Vec space.

Further, SkipThoughts show poor performance in general, however, work well on "Why" and "When" questions. SkipThoughts show a tendency to capture the semantic meaning of a sentence, and are thus likely to perform well on "Why" questions. Based on the training procedure (a SkipThought embedding for sentence i is trained to predict the next $i+1$ and previous $i-1$ sentences), causality seems to be a strong suite for SkipThoughts, and they outperform other representations on "When" questions.

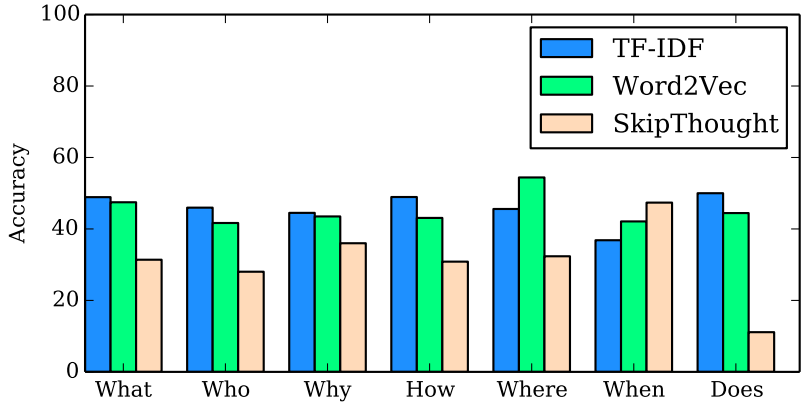


Figure 6.6: Accuracy of QA using plot synopses and cosine similarity split across different question types based on the first word.

Neural similarity. The middle section of Table 6.7 presents the results for our neural similarity model. We further split the *train* set (90% train, 10% development) to train the neural similarity model and report performance on the *val* set. The *test* split is not used. The Neural similarity model performs slightly better or equal to the Cosine similarity model. We expect such a result as the CNN essentially takes the individual inner product scores as part of the input tensor, and performs a weighted combination on the question-story and answer-story similarity.

In addition to the three features, we present performance with feature fusion (concatenation) in row 7 of Table 6.7. The fusion presents the best results across all techniques when using plot synopses as stories. We see small improvements over the individual descriptors for all other story forms. In practice, the Convolutional model was found to be quite unstable. We overcome the problem by training several different weight initializations of the model for 30 epochs, and use the development split (a part of the *train* set) to choose the best model.

Modified memory network. The original end-to-end memory network was proposed by Sukhbaatar et al. (2015) on the bAbI QA task (Weston et al., 2015a). The question and answers from that problem are synthetically generated and come from a small vocabulary, and thus allow end-to-end training with small amounts of training data (*e.g.* 1000 QAs). We presented key modifications that allow us to use the memory network for our problem in Sec. 6.2.5. The network without these modifications overfits heavily on the training data, and obtains near random performance on an unseen test set.

In the bottom part of Table 6.7 we present the results of several configurations of the memory network for answering questions from our data set. In the simplest case, we use a linear projection layer (recall T) initialized using an identity matrix. We use a single-layer network and share weights across all projections (question, two story banks, and multiple choice answer options). The results for this configuration are presented in row 8 of the table. The memory network shows superior performance on three out of four story types: DVS, Subtitle, and Script. The performance when answering using plot synopses is diluted through the use of the complex (and possibly not so necessary) attention mechanism as is evident from the better performance using Cosine or Neural similarity models. The plots are typically short documents (about 30-40 sentences) as compared to Subtitles or Scripts (in the order of 1000 sentences) and attention benefits the longer story sources better. In fact, *scripts* that contain the most information about the story (scene, dialog, and speaker) show performance slightly higher than using plots.

We further analyze several variants of this memory network. We present on row 9 of the Table 6.7 the scenario when the linear projection T is a 300×100 layer initialized through PCA. On row 10, we present a variant of row 1 that shares weights across three layers of story representations. All through the above variants, sentences are encoded as a bag-of-words and the individual word representations of the sentence are averaged. On row 11, we present a scheme that encodes the position of the words in the sentences prior to averaging (see Position Encoding (Sukhbaatar et al., 2015)).

The original MemN2N demonstrates strong benefits on the bAbI data set by including multiple layers and encoding the position of the words in the sentence. We see a couple trends from the analysis. Using a linear projection initialized with an identity matrix is preferable to dimensionality reduction and initialization with PCA. The 3 layered network performs slightly better on plots and scripts, but worse on subtitles. All the variants show very minor changes (increase or decrease) from the simple single layer, all weights shared model (row 8).

The memory networks are trained on 90% of the *train* split and validated on the remainder (just like the Neural similarity model) for 100 epochs, and the numbers in Table 6.7 report accuracy on the *val* split. During training, we update parameters using Stochastic Gradient Descent with a learning rate of 0.01 decaying by 10% every 10 epochs, and a batch size of 8-32 (constrained by the GPU memory).

Method		Video	Subtitle	Video+Subtitle
Neural Similarity	All clips	21.6	22.3	21.9
Memory Network	All clips	**23.1**	**38.0**	**34.2**
	QA clips	22.6	**38.0**	33.3

Table 6.8: QA accuracy using video as the information source for answering. We present results for the Neural similarity and the modified Memory Network.

6.3.4 Video-based answering

We present the results for question answering using video clips in Table 6.8. As is evident, answering through a complex structure such as video is a difficult problem. We use the top performing methods in text-based answering, Neural similarity and the modified Memory network to attempt this task.

We present two modes of answering: (i) All clips: using all the provided clips for the movie; and (ii) QA clips: using only those clips that are associated with the QA, obtained through the plot to video alignment. In both scenarios, the video stories are represented using the video embeddings presented in Sec. 6.2.6. The question and answers are embedded into the common text-video ranking space, and directly used in place of the Word2Vec features for the Memory network or Neural similarity models.

From Table 6.8 we see that video-based answering shows very poor performance with both methods. This can be justified by the following key observations. Firstly, the video embeddings are trained for ranking descriptions and not question-answering. Secondly, they use object recognition features, and thus ignore a significant aspect of human analysis (identities and actions) that is critical for story understanding. For example, a large number of questions that should be able to answer with videos are of the type "Who did something", and without identities, they are answered at random. The questions that we may be able to answer are places ("Where") and objects ("What"). However, given an accuracy so close to random, we are unable to determine whether a particular type of question gets answered correctly.

Further, we see that looking at all the clips from the video is slightly beneficial as compared to a restricted look at the clips associated with the QA only. This can be attributed to the attention mechanism of the memory network as it can sift through more information.

As described in the introduction, not all questions can be answered using video alone. For example, "Why" something happened in the story, often becomes clear due to the dialogs. We wish to promote an answering task that leverages information jointly from video clips and corresponding subtitles.

To this end, we first measure the quality of the video-text embedding space, by evaluating subtitle-only answering with the Neural similarity model. Here, we see performance close to that achieved by SkipThoughts (22.3% vs. 20.8%), as both spaces try to encode the semantic meaning of sentences, and not their details. For the memory network, we use the Word2Vec mean pooled sentence representation as before.

We combine the information from the subtitles and video clips using late fusion. Regrettably, the bad performance on the video clips drags down the result of the fusion. We demonstrate the difficulty of video-based answering, and hope that the MovieQA benchmark will encourage innovative solutions to truly bridge the semantic gap between text and video in relation to story understanding.

6.4 Benchmark

The data set is made publicly available at http://movieqa.cs.toronto.edu/ and is being developed as a competitive benchmark including an evaluation server. We encourage teams to submit the answers predicted by their automatic systems on the *test* set and compete with the rest of the world. All forms of story sources: plots, subtitles, scripts, DVS, and video clips are made available. We also include suitable Python code for reading and parsing the data set. From the launch of the first version of the benchmark data on 30th March 2016, we currently have over 20 registered teams from all over the world.

Acknowledgment. We thank several annotators on the freelance website upwork.com that were recruited to create QAs. We also thank Lea Jensterle for aligning movie videos with plot synopsis sentences; Soča Fidler for sharing her experiences as an English teacher, and helping with data collection process; and Relu Patrascu for tremendous help in setting up the benchmark data set, and other infrastructure related issues. I thank the Research Travel Grant provided Karlsruhe House of Young Scientists for a funded visit to the University of Toronto.

Chapter 7

Conclusion

AI has seen rapid progress in the last few years, and conventional vision tasks such as object detection and recognition, semantic segmentation, action recognition, *etc*. are paving way to joint language and vision tasks such as captioning and question-answering. In this thesis, we argue for analysis and understanding of stories as the next frontier of AI research. We broadly define "machine understanding of stories" as the ability to (i) index, search, and summarize stories; (ii) answer questions about them; (iii) find common patterns across stories; and (iv) translate or generate stories in different media forms. In this thesis, we focus on the first two areas.

Our emphasis is on analyzing videos, TV series and films, that are a rich and engaging medium of story-telling. Building upon previous work on identifying people in such videos, we analyze the story understanding problem through three different lenses:

- We propose to align videos with narrative forms of text such as plot synopses and books, and unlock applications such as story-based search (Chapter 4);

- We present methods to automatically generate visualizations that summarize character interactions, and thereby, the underlying story of videos (Chapter 5); and

- We create a large question-answering data set based on movie stories, propose several baselines, and set it up as a challenge for machine understanding (Chapter 6).

7.1 Contributions

We summarize the key contributions made in this thesis, along with their corresponding publications:

Plots and Books: alignment of novel data sources for story understanding.
Related publications: (Tapaswi et al., 2014a, 2015a,c)
We propose to leverage information from natural language descriptions of videos. We introduce the use of *plot synopses*, commonly found on Wikipedia, as concise descriptions of the story conveyed in the TV episodes or movies; and *books* whose video adaptations are increasingly common. The modality gap between the text and video is bridged using similarity functions based on common cues found in both sources – identities and dialogs. We motivate the necessity of aligning video segments with text units, and in particular focus on fine-grained alignment between plot synopsis sentences and shots of the video; and a coarse alignment between book chapters and scenes of its video adaptation. Alignment is treated as an optimization problem that maximizes joint similarity while respecting several story progression constraints.

We perform a thorough evaluation, and in the case of plots are able to obtain almost 50% accuracy for *Buffy the Vampire Slayer* (BF) and 35% for the much more challenging *Game of Thrones* (GOT) series. In the case of books, we are able to align over 90% of the scenes with the correct book chapter for *Harry Potter and the Sorcerer's Stone*, and achieve 75% when aligning scenes from the GOT TV series. We are also able to predict whether the scene was part of the book for GOT with a precision of 70% and recall of 53%.

We investigate several applications that arise from the alignments. In particular, rich and concise video descriptions can be obtained from books and plots respectively. Alignment with plot synopsis also facilitates story-based video retrieval – searching for story events within large video collections. For BF, even with an alignment performance in the range of 50%, we obtain good retrieval performance and are able to find a video segment that overlaps with the ground truth for 64% of the queries. We also take a first step towards finding differences between books and their video adaptations.

StoryGraphs: a snapshot of the story in a video.
Related publication: (Tapaswi et al., 2014b)
Inspired by a web comic `XKCD:657`, we propose a method to automatically generate StoryGraphs – a single picture overview that visualizes character interactions in a video.

We contemplate several properties that such a graph should exhibit, and capture their essence through energy functions. Generating a good layout is thus transformed into an energy minimization problem. We quantify the properties of StoryGraphs, discuss them qualitatively, and compare them against stories in plot synopses. Through a user experiment, we show that availability of StoryGraphs results in speeding up the process of finding story events in a video.

MovieQA: a question-answering data set and benchmark.
Related publication: (Tapaswi et al., 2016)
Machine understanding of stories can be verified by analyzing whether it is able to answer questions about the story. While there is an increasing interest in Visual QA on images, we are the first to introduce a data set for QA using videos. In contrast to images, movies have the distinct advantage of requiring the machine to learn causality and long-range temporal reasoning to answer questions about the story.

We create a large QA data set with almost 15,000 questions sourced from over 400 movies. Each question comes with 5 deceiving multiple choice answers only one of which is correct. Apart from standard questions such as "Who", "What", and "Where", our data set also features "Why" and "How" questions that are much more challenging to answer.

We analyze the data set bias, ask humans to answer questions without looking at the story, and develop answering methods using Convolutional Neural Networks (CNN). We also propose modifications to end-to-end Memory Networks (MemN2N) so as to make them suitable for our task involving large vocabulary and multiple natural language answers. The evaluation shows the difficulty of the data set. In particular, without the story, even humans perform close to random, thus validating the difficulty of the collected multiple-choice options. Our CNN and Memory Network are able to obtain accuracies close to 40% when answering questions using text information sources. Video-based answering is considerably harder.

Our data set is unique, in that it supports both text- and video-based answering. The difficulty in answering the questions demonstrates a large scope for improvement. We are building *MovieQA*, a benchmark to improve machine understanding of stories.

Video analysis: scene detection and person identification.
Related publications: (Tapaswi et al., 2014b,c, 2015b)
Apart from the main contributions listed above, we also make minor improvements in the field of video analysis. In particular, we propose a scene boundary detection

algorithm based on dynamic programming that provides optimal scene cuts, and suggest shot threads as a very good cue for the task. In the field of person identification, we work on a stage-wise clustering approach that facilitates discriminative face track clustering, while maintaining a purity close to 1. Using such a clustering, we propose important modifications to improve weak labeling of face tracks used to train character models by first aligning subtitles with transcripts. These modifications prove to be essential while working with high production quality TV series (*e.g. Game of Thrones*) or movies.

Publicly released data sets and benchmarks.
As the thesis addresses mostly novel problems, we build and make available data sets for future research on each of them. Most prominently, we introduce the MovieQA data set and benchmark to facilitate and encourage improvements in story understanding. For plot synopsis alignment, we work with one season each of two diverse TV series – *Buffy the Vampire Slayer* and *Game of Thrones*. For aligning books, we focus on the fantasy genre, and look at first book of the *A Song of Ice and Fire* and *Harry Potter* series. The ground truth alignments and face tracks are available for future research.

While not a key aspect of this thesis, we extend the KIT-TV series data set to include 8 episodes of the *Lost* TV series (Haurilet et al., 2016). We also build a face track data set based on the characters appearing in all movies of the *Harry Potter* series that span 10 years. This allows to study the effect of aging on the face appearance (Ghaleb et al., 2015).

7.2 Future work and open directions

Joint analysis of vision and language has received much attention in the last couple years. In particular, joint embeddings for sentences encoded using RNNs, and images (or videos) using CNNs are a hot topic. An obvious extension of our work on text-to-video alignment is to augment the similarity functions with information from such joint embeddings. A different direction is to merge information from different sources, *e.g.* person identification with holistic image features. We mention several applications of text-to-video alignment that have not been explored in detail: (i) video summarization by first performing text summarization on plot synopsis; (ii) fine grained differentiation between books and their video adaptations; and (iii) mining rich descriptions from books to automatically describe, or tag video shots. Finally, the MovieQA data set is a very challenging problem, and we hope that it will inspire interesting models and architectures both in vision and language communities.

Short CV

Makarand Tapaswi

Contact	makarand.tapaswi@kit.edu
Website	https://cvhci.anthropomatik.kit.edu/~mtapaswi

Education and Experience

since 10/2011	Research Assistant and Ph.D. student Computer Vision for Human-Computer Interaction Lab Karlsruhe Institute of Technology
09/2015 - 12/2015	Research Internship Prof. Fidler and Prof. Urtasun, University of Toronto, Canada
04/2013 - 07/2013	Research Internship Prof. Zisserman, University of Oxford, U.K.
09/2009 - 09/2011	Master of Science Information and Communication Technologies Erasmus Mundus MERIT UPC Barcelona and KIT Karlsruhe Overall grade: 1.1 / 5 with distinction (German scale)
08/2005 - 04/2009	Bachelor of Technology Electronics and Communication Engineering National Institute of Technology Karnataka, India Overall grade (CGPA): 9.09 / 10

Own Publications

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. "Knock! Knock! Who is it?" Probabilistic Person Identification in TV series. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2012. 27

Martin Bäuml, Makarand Tapaswi, Arne Schumann, and Rainer Stiefelhagen. Contextual Constraints for Person Retrieval in Camera Networks. In *IEEE Conference on Advanced Video and Signal-based Surveillance (AVSS)*, Sep. 2012.

Martin Bäuml, Makarand Tapaswi, and Rainer Stiefelhagen. Semi-supervised Learning with Constraints for Person Identification in Multimedia Data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2013. 3, 15, 18, 27, 41, 49, 96

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Story-based Video Retrieval in TV series using Plot Synopses. In *ACM International Conference on Multimedia Retrieval (ICMR)*, Apr. 2014a. 8, 53, 142

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. StoryGraphs: Visualizing Character Interactions as a Timeline. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2014b. 8, 37, 95, 107, 142, 143

Martin Bäuml, Makarand Tapaswi, and Rainer Stiefelhagen. A Time Pooled Track Kernel for Person Identification. In *IEEE Conference on Advanced Video and Signal-based Surveillance (AVSS)*, Aug. 2014.

Makarand Tapaswi, Cemal Çağrı Çörez, Martin Bäuml, Hazım Kemal Ekenel, and Rainer Stiefelhagen. Cleaning up after a Face Tracker: False Positive Removal. In *IEEE International Conference on Image Processing (ICIP)*, Oct. 2014c.

Makarand Tapaswi, Omkar M. Parkhi, Esa Rahtu, Eric Sommerlade, Rainer Stiefelhagen, and Andrew Zisserman. Total Cluster: A person agnostic clustering method for

broadcast videos. In *ACM Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, Dec. 2014d. 8, 41, 43, 46, 143

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Aligning Plot Synopses to Videos for Story-based Retrieval. *International Journal of Multimedia Information Retrieval (IJMIR)*, 4(1):3–16, 2015a. 8, 53, 142

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Improved Weak Labels using Contextual Cues for Person Identification in Videos. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, May 2015b. 8, 15, 46, 51, 61, 143

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Book2Movie: Aligning Video scenes with Book chapters. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015c. 8, 53, 142

Esam Ghaleb, Makarand Tapaswi, Ziad Al-Halah, Hazım Kemal Ekenel, and Rainer Stiefelhagen. Accio: A Data Set for Face Track Retrieval in Movies Across Age. In *ACM International Conference on Multimedia Retrieval (ICMR)*, Jun. 2015. 144

Monica-Laura Haurilet, Makarand Tapaswi, Ziad Al-Halah, and Rainer Stiefelhagen. Naming TV Characters by Watching and Analyzing Dialogs. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2016.

Ziad Al-Halah, Makarand Tapaswi, and Rainer Stiefelhagen. Recovering the Missing Link: Predicting Class-Attribute Associations for Unsupervised Zero-Shot Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. MovieQA: Understanding Stories in Movies through Question-Answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016. 8, 113, 143

Bibliography

Whoosh - a Python full text indexing and search library. http://pypi.python.org/pypi/Whoosh. 90

E. Agichtein, S. Lawrence, and L. Gravano. Learning Search Engine Specific Query Transformations for Question Answering. 2001. 22

Z. Al-Halah, M. Tapaswi, and R. Stiefelhagen. Recovering the Missing Link: Predicting Class-Attribute Associations for Unsupervised Zero-Shot Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 12

J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Deep Compositional Question Answering with Neural Module Networks. In *arXiv:1511.02799*, 2015. 25

A. Aner and J. R. Kender. Video Summaries through Mosaic-Based Shot and Scene Clustering. In *European Conference on Computer Vision (ECCV)*, 2002. 19

S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015. 3, 24, 25, 113, 114, 118

J. Ba, K. Swersky, S. Fidler, and R. Salakhutdinov. Predicting Deep Zero-Shot Convolutional Neural Networks using Textual Descriptions. In *International Conference on Computer Vision (ICCV)*, 2015. 12

D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations (ICLR)*, 2015. 3, 14

C. F. Baker, C. J. Fillmore, and J. B. Lowe. The Berkeley FrameNet Project. In *International Conference on Computational Linguistics (COLING)*, 1998. 108

S. Banerjee and A. Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, jointly held with ACL*, 2005. 12

J. Bao, N. Duan, M. Zhou, and T. Zhao. Knowledge-Based Question Answering as Machine Translation. In *Association of Computational Linguistics (ACL)*, 2014. 22

Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 22(1):22–29, 2001. 101

A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanswamy, and D. Salvi. Video In Sentences Out. In *Uncertainty in Artificial Intelligence (UAI)*, 2012. 13

K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching Words and Pictures. *Journal of Machine Learning Research (JMLR)*, 3:1107–1135, 2003. 10

G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1998. 20

M. Bäuml, K. Bernardin, M. Fischer, H. Ekenel, and R. Stiefelhagen. Multi-Pose Face Recognition for Person Retrieval in Camera Networks. In *International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, 2010. 41

S. Bhattacharya, M. M. Kalayeh, R. Sukthankar, and M. Shah. Recognition of Complex Events: Exploiting Temporal Dynamics between Underlying Concepts. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 18

P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly Supervised Action Labeling in Videos under Ordering Constraints. In *European Conference on Computer Vision (ECCV)*, 2014. 16

P. Bojanowski, R. Lagugie, E. Grave, F. Bach, I. Laptev, J. Ponce, and C. Schmid. Weakly-Supervised Alignment of Video With Text. In *International Conference on Computer Vision (ICCV)*, 2015. 16, 74

P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding Actors and Actions in Movies. In *International Conference on Computer Vision (ICCV)*, 2013. 16

C. Booker. *The Seven Basic Plots: Why We Tell Stories*. Continuum, 2005. 1

M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's Mechanical Turk: A New Source of Inexpensive, Yet High-Quality Data? *Perspectives on Psychological Science*, 6 (1):3–5, 2011. 124

L. Byron and M. Wattenberg. Stacked Graphs - Geometry and Aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14:1245–1252, 2008. 20

M. Bäuml. *Contextual Person Identification in Multimedia Data*. PhD thesis, Karlsruhe Institute of Technology (KIT), 2014. URL http://digbib.ubka.uni-karlsruhe.de/volltexte/1000047232. 41, 46

M. Bäuml, M. Tapaswi, and R. Stiefelhagen. Semi-supervised Learning with Constraints for Person Identification in Multimedia Data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 3, 15, 18, 27, 41, 49, 96

G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised Learning of Semantic Classes for Image Annotation and Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(3):394–410, 2007. 10

V. T. Chasanis, A. C. Likas, and N. P. Galatsanos. Scene Detection in Videos Using Shot Clustering and Sequence Alignment. *IEEE Transactions on Multimedia*, 11(1):89–100, 2009. 37

B.-W. Chen, J.-C. Wang, and J.-F. Wang. A Novel Video Summarization Based on Mining the Story-Structure and Semantic Relations Among Concept Entities. *IEEE Transactions on Multimedia*, 11(2):295–312, 2009. 19

D. L. Chen and W. B. Dolan. Collecting Highly Parallel Data for Paraphrase Evaluation. In *Association of Computational Linguistics (ACL)*, 2011. 10, 11

X. Chen and L. C. Zitnick. Mind's Eye: A Recurrent Visual Representation for Image Caption Generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5, 13

X. Chen, A. Shrivastava, and A. Gupta. NEIL : Extracting Visual Knowledge from Web Data. In *International Conference on Computer Vision (ICCV)*, 2013. 18

K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 3

V. Chvátal and D. Sankoff. Longest Common Subsequences of Two Random Sequences. *Journal of Applied Probability*, 12(2):306–315, 1975. 62

P. Clark, J. Thompson, and B. Porter. A Knowledge-Based Approach to Question-Answering. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 1999. 22

T. Cour, B. Sapp, C. Jordan, and B. Taskar. Learning from ambiguously labeled images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 15, 18

T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription. In *European Conference on Computer Vision (ECCV)*, 2008. 15, 37

T. Cour, B. Sapp, A. Nagle, and B. Taskar. Talking Pictures : Temporal Grouping and Dialog-Supervised Person Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 15

J. E. Cutting and A. Candan. Shot Durations, Shot Classes, and the Increased Pace of Popular Movies. *Projections*, 2(2):40–62, 2015. 34

N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 2

D. Das, D. Chen, A. F. T. Martins, N. Schneider, and N. A. Smith. Frame-Semantic Parsing. *Computational Linguistics*, 40(1):9–56, 2014. 108

P. Das, C. Xu, R. F. Doell, and J. J. Corso. A Thousand Frames in Just a Few Words: Lingual Description of Videos Through Latent Topics and Sparse Object Stitching. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 10, 13

A. Dehghan, H. Idrees, and M. Shah. Improving Semantic Concept Detection through the Dictionary of Visually-distinct Elements. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 18

D. DeMenthon, V. Kobla, and D. Doermann. Video Summarization by Curve Simplification. In *ACM Multimedia (MM)*, 1998. 19

E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *arXiv:1506.05751*, 2015. 5

S. K. Divvala, A. Farhadi, and C. Guestrin. Learning Everything about Anything: Webly-Supervised Visual Concept Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 18

J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 14

P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In *European Conference on Computer Vision (ECCV)*, 2002. 10

M. Elhoseiny, B. Saleh, and A. Elgammal. Write a Classifier: Zero-Shot Learning Using Purely Textual Descriptions. In *International Conference on Computer Vision (ICCV)*, 2013. 12

P. Ercolessi, H. Bredin, and C. Sénac. StoViz: Story Visualisation of TV Series. In *ACM Multimedia Demo*, 2012a. 20

P. Ercolessi, H. Bredin, and C. Sénac. Toward Plot De-Interlacing in TV Series using Scenes Clustering. In *International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2012b. 21

M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, 2015. 3, 10

M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is ... Buffy" – Automatic Naming of Characters in TV Video. In *British Machine Vision Conference (BMVC)*, 2006. 15, 16, 18, 29, 31, 46, 47, 48, 50, 53, 61, 64, 73, 87, 96

H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, J. G. P. Dollar, X. He, M. Mitchell, and J. Platt. From Captions to Visual Concepts and Back. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 13

A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth. Describing Objects by their Attributes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 11

A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every Picture Tells a Story : Generating Sentences from Images. In *European Conference on Computer Vision (ECCV)*, 2010. 13

V. Ferrari and A. Zisserman. Learning Visual Attributes. In *Advances in Neural Information Processing Systems (NIPS)*, 2008. 11

D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty. Building Watson: An Overview of the DeepQA Project. *AI Magazine*, 31(3), 2010. 22

M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981. 37

J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971. 78

B. Fröba and A. Ernst. Face Detection with the Modified Census Transform. In *International Conference on Automatic Face and Gesture Recognition (FG)*, 2004. 41

A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. A. Ranzato, and T. Mikolov. DeViSE: A Deep Visual-Semantic Embedding Model. In *Advances in Neural Information Processing Systems (NIPS)*, 2013. 12, 56

C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann. DevNet: A Deep Event Network for Multimedia Event Detection and Evidence Recounting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 18

E. Ghaleb, M. Tapaswi, Z. Al-Halah, H. K. Ekenel, and R. Stiefelhagen. Accio: A Data Set for Face Track Retrieval in Movies Across Age. In *International Conference on Multimedia Retrieval (ICMR)*, 2015. 144

R. Giddings, K. Selby, and C. Wensley. *Screening the novel: The theory and practice of literary dramatization*. Macmillan, 1990. 93

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 13, 60

Y. Gong and X. Liu. Video Summarization using Singular Value Decomposition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000. 19

M. Graham and J. Kennedy. Using Curves to Enhance Parallel Coordinate Visualizations. In *International Conference on Information Visualization (iV)*, 2003. 20

S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. YouTube2Text: Recognizing and Describing Arbitrary Activities Using Semantic Hierarchies and Zero-shot Recognition. In *International Conference on Computer Vision (ICCV)*, 2013. 3, 5, 10

M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? Metric Learning Approaches for Face Identification. In *International Conference on Computer Vision (ICCV)*, 2009. 41, 43

M. Guillaumin, J. Verbeek, and C. Schmid. Multimodal Semi-supervised Learning for Image Classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 11

A. Gupta and L. S. Davis. Beyond Nouns: Exploiting Prepositions and Comparative Adjectives for Learning Visual Classifiers. In *European Conference on Computer Vision (ECCV)*, 2008. 11

A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding Videos, Constructing Plots Learning a Visually Grounded Storyline Model from Annotated Videos Input. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 11

B. Han and W. Wu. Video Scene Segmentation using a Novel Boundary Evaluation Criterion and Dynamic Programming. In *International Conference on Multimedia and Expo (ICME)*, 2011. 37, 38

A. Hauptmann, R. Yan, W.-h. Lin, M. Christel, and H. Wactlar. Can High-Level Concepts Fill the Semantic Gap in Video Retrieval? A Case Study With Broadcast News. *IEEE Transactions on Multimedia*, 9(5):958–966, 2007. 17

M.-L. Haurilet, M. Tapaswi, Z. Al-Halah, and R. Stiefelhagen. Naming TV Characters by Watching and Analyzing Dialogs. 2016. 15, 144

K. M. Hermann, T. Kočisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching Machines to Read and Comprehend. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 23, 118

S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9 (8):1735–1780, 1997. 3

M. Hodosh, P. Young, and J. Hockenmaier. Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013. 10

E. Hovy, U. Herjakob, and C.-Y. Lin. The Use of External Knowledge of Factoid QA. *TREC*, 2001. 22

G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007. 3

P. J. Huber. Robust Estimation of a Location Parameter. *Annals of Statistics*, 53:73–101, 1964. 99

M. J. Huiskes, B. Thomee, and M. S. Lew. New Trends and Ideas in Visual Concept Detection: the MIR Flickr Retrieval Evaluation Initiative. 2010. 11

B. Huurnink and M. de Rijke. The value of stories for speech-based video search. In *International Conference on Image and Video Retrieval (CIVR)*, 2007. 18

B. Huurnink, C. G. M. Snoek, M. D. Rijke, and A. W. M. Smeulders. Content-Based Analysis Improves Audiovisual Archive Retrieval. *IEEE Transactions on Multimedia*, 14(4):1166–1178, 2012. 18

G. Irie, T. Satou, A. Kojima, T. Yamasaki, and K. Aizawa. Automatic Trailer Generation. In *ACM Multimedia (MM)*, 2010. 19

S. Ji, W. Xu, M. Yang, and K. Yu. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(1):221–231, 2013. 14

A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale Video Classification with Convolutional Neural Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3

A. Karpathy and L. Fei-Fei. Deep Visual-Semantic Alignments for Generating Image Descriptions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5, 13, 56

A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-Scale Video Summarization Using Web-Image Priors. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 19

E. Khoury, P. Gay, and J.-M. Odobez. Fusing Matching and Biometric Similarity Measures for Face Diarization in Video. In *International Conference on Multimedia Retrieval (ICMR)*, 2013. 41

G. Kim, L. Sigal, and E. P. Xing. Joint Summarization of Large-scale Collections of Web Images and Videos for Storyline Reconstruction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 19

D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. 2015. 126

R. Kiros, R. Zemel, and R. Salakhutdinov. Multimodal Neural Language Models. In *International Conference on Machine Learning (ICML)*, 2014. 11, 56

R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. In *Association of Computational Linguistics (ACL)*, 2015a. 5, 131

R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-Thought Vectors. In *Advances in Neural Information Processing Systems (NIPS)*, 2015b. 3, 33, 56, 90, 93, 131

Y. Kiyota, S. Kurohashi, and F. Kido. "Dialog Navigator": A Question Answering System Based on Large Text Knowledge Base. 2002. 22

S. G. Kobourov. Spring Embedders and Force-Directed Graph Drawing Algorithms. 2012. 97

A. Kojima, T. Tamura, and K. Fukunaga. Natural Language Description of Human Activities from Video Images based on Concept Hierarchy of Actions. *International Journal of Computer Vision (IJCV)*, 50(2):171–184, 2002. 13

M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof. Learning to Recognize Faces from Videos and Weakly Related Information Cues. In *International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, 2011. 15

A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 2, 12, 25

C. Küblbeck and A. Ernst. Face Detection and Tracking in Video Sequences using the Modified Census Transformation. *Image and Vision Computing*, 24(6):564–572, 2006. 41

G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(12):2891–2903, 2013. 13

A. Kumar, O. Irsoy, J. Su, J. Bradbury, R. English, B. Pierce, P. Ondruska, M. Iyyer, I. Gulrajani, and R. Socher. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. In *arXiv:1506.07285*, 2015. 24

C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to Detect Unseen Object Classes by between Class Attribute Transfer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 11

I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning Realistic Human Actions from Movies. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 3, 15, 16, 29, 53, 64, 73, 87

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature - Insight Review*, 521(7553): 436–444, 2015. 2

H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky. Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In *Computational Natural Language Learning (CoNLL)*, 2011. 34, 58

J. Li and J. Wang. Automatic Linguistic Indexing of Pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(9): 1075–1088, 2003. 10

Y. Li, S.-H. Lee, C.-H. Yeh, and C.-C. J. Kuo. Techniques for Movie Content Analysis and Skimming. *IEEE Signal Processing Magazine*, 23(2):79–89, 2006. 19

Y. Li, R. Zemel, M. Brockschmidt, and D. Tarlow. Gated Graph Sequence Neural Networks. In *arXiv:1511.05493*, 2015. 24

C. Liang, Y. Zhang, J. Cheng, C. Xu, and H. Lu. A Novel Role-Based Movie Scene Segmentation Method. In *Pacific Rim Conference on Multimedia*, 2009. 37

C. Liang, C. Xu, J. Cheng, W. Min, and H. Lu. Script-to-Movie : A Computational Framework for Story Movie Composition. *IEEE Transactions on Multimedia*, 15(2): 401–414, 2013. 16

C.-Y. Lin and E. Hovy. Automatic Evaluation of Summaries using N-Gram Co-occurrence Statistics. In *Association of Computational Linguistics (ACL)*, 2003. 12

D. Lin, S. Fidler, C. Kong, and R. Urtasun. Visual Semantic Search: Retrieving Videos via Complex Textual Queries. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014a. 16

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*, 2014b. 3, 10, 24, 25, 120

E. Loper, E. Klein, and S. Bird. *Natural Language Processing with Python*. O'Reilly Media, 3 edition, 2009. 34, 59

D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004. 2, 37

M. Malinowski and M. Fritz. A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 24, 25, 113, 118

M. Malinowski, M. Rohrbach, and M. Fritz. Ask Your Neurons: A Neural-based Approach to Answering Questions about Images. In *International Conference on Computer Vision (ICCV)*, 2015. 25, 123

T. Malisiewicz, A. Gupta, and A. Efros. Ensemble of Exemplar-SVMs for Object Detection and Beyond. In *International Conference on Computer Vision (ICCV)*, 2011. 42

C. D. Manning, P. Raghavan, and H. Schutze. Scoring, Term Weighting, and the Vector Space Model. In *Introduction to Information Retrieval*. Cambridge University Press, 2008. 59, 62

C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association of Computational Linguistics (ACL): System Demonstrations*, 2014. 33

E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov. Generating Images from Captions with Attention. In *arXiv:1511.02793*, 2015. 5

J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Deep Captioning with Multimodal Recurrent Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2015. 13, 14

M. Marszalek and C. Schmid. Semantic Hierarchies for Visual Object Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 11

M. Marszalek, I. Laptev, and C. Schmid. Actions in Context. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 15

B. McFarlane. *Novel to Film: an Introduction to the Theory of Adaptation*. Clarendon press, Oxford, 1996. 93

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. In *International Conference on Learning Representations (ICLR)*, 2013. 3, 12, 93, 129, 130

G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995. 11

C. S. Myers and L. R. Rabiner. A Comparative Study of Several Dynamic Time-Warping Algorithms for Connected Word Recognition. *Bell System Technical Journal*, 1981. 67

H.-H. Nagel. Steps toward a Cognitive Vision System. *AI Magazine*, 25(2):31–50, 2004. 12

K. Nesbitt. Getting to more Abstract Places using the Metro Map Metaphor. In *International Conference on Information Visualization (iV)*, 2004. 20

J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal Deep Learning. In *International Conference on Machine Learning (ICML)*, 2011. 11

M. Nollenburg and A. Wolff. Drawing and Labeling High-quality Metro Maps by Mixed-integer Programming. *IEEE Transactions on Visualization and Computer Graphics*, 17 (5):626–641, 2011. 20

V. Ordonez, G. Kulkarni, and T. L. Berg. Im2Text : Describing Images Using 1 Million Captioned Photographs. In *Advances in Neural Information Processing Systems (NIPS)*, 2011. 10, 13

K. Papineni, S. Rouos, T. Ward, and W.-J. Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Association of Computational Linguistics (ACL)*, 2002. 12

D. Parikh and K. Grauman. Relative Attributes. In *International Conference on Computer Vision (ICCV)*, 2011. 11

O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A Compact and Discriminative Face Track Descriptor. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 41

Y. Peng and J. Xiao. Story-based retrieval by learning and measuring the concept-based and content-based similarity. In *Advances in Multimedia Modeling*, 2010. 18

F. Perronnin and D. Dance. Fisher Kernels on Visual Vocabularies for Image Categorization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 42

M. F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137, 1980. 34, 59

M. F. Porter. Snowball: A language for stemming algorithms. http://snowball.tartarus.org/texts/introduction.html, 2001. 34, 130

D. Ramanan, S. Baker, and S. Kakade. Leveraging Archival Video for Building Face Datasets. In *International Conference on Computer Vision (ICCV)*, 2007. 41

V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei. Linking people in videos with "their" names using coreference resolution. In *European Conference on Computer Vision (ECCV)*, 2014. 16

Z. Rasheed and M. Shah. Detection and representation of scenes in videos. *IEEE Transactions on Multimedia*, 7(6):1097–1105, 2005. 37, 38, 39, 40

M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal. Grounding Action Descriptions in Videos. In *Association of Computational Linguistics (ACL)*, 2013. 10, 16

S. Ren, X. Cao, Y. Wei, and J. Sun. Face Alignment at 3000 FPS via Regressing Local Binary Features. In *CVPR*, 2014. 48

J. Revaud, M. Douze, C. Schmid, and H. Jégou. Event retrieval in large video collections with circulant temporal encoding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 18

M. Richardson, C. J. Burges, and E. Renshaw. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2013. 22, 23, 118

D. F. Rogers and J. A. Adams. *Mathematical Elements for Computer Graphics*. McGraw-Hill, 2 edition, 1990. 66

A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele. Coherent Multi-Sentence Video Description with Variable Level of Detail. In *German Conference on Pattern Recognition (GCPR)*, 2014. 10

A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A Dataset for Movie Description. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5, 11, 16, 17, 31, 32, 89, 114, 115, 131

M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating Video Content to Natural Language Descriptions. In *International Conference on Computer Vision (ICCV)*, 2013. 10, 13

A. Roy, C. Guinaudeau, H. Bredin, and C. Barras. TVD: A Reproducible and Multiply Aligned TV Series Dataset. In *Language Resources and Evaluation Conference*, 2014. 16

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 3, 12

J. Sang and C. Xu. Character-based movie summarization. In *ACM Multimedia (MM)*, 2010. 19

P. Sankar, C. V. Jawahar, and A. Zisserman. Subtitle-free Movie to Script Alignment. In *British Machine Vision Conference (BMVC)*, 2009. 17, 68, 73

F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. 24

J. See and C. Eswaran. Exemplar Extraction Using Spatio-Temporal Hierarchical Agglomerative Clustering for Face Recognition in Video. In *International Conference on Computer Vision (ICCV)*, 2011. 41

A. Shrestha, Y. Zhu, B. Miller, and Y. Zhao. Storygraph: Extracting patterns from spatio-temporal data. In *ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*, 2013. 21

N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *European Conference on Computer Vision (ECCV)*, 2012. 24, 120

J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering Objects and their Location in Images. In *International Conference on Computer Vision (ICCV)*, 2005a. 10

J. Sivic, M. Everingham, and A. Zisserman. "Who are you?" – Learning person specific classifiers from video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 15, 18, 96

J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *International Conference on Computer Vision (ICCV)*, 2003. 18

J. Sivic, M. Everingham, and A. Zisserman. Person spotting: video shot retrieval for face sets. In *International Conference on Image and Video Retrieval (CIVR)*, 2005b. 18

A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *International Workshop on Nultimedia Information Retrieval*, 2006. 17

E. Smith, N. Greco, M. Bošnjak, and A. Vlachos. A Strong Lexical Matching Method for the Machine Comprehension Test. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2015. 24

C. G. M. Snoek and M. Worring. Concept-Based Video Retrieval. *Foundations and Trends in Information Retrieval*, 2(4):215–322, 2007. 17

N. Srivastava and R. Salakhutdinov. Multimodal Learning with Deep Boltzmann Machines. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 11

S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-To-End Memory Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 23, 114, 122, 126, 127, 128, 136, 137

C. Sun and R. Nevatia. DISCOVER: Discovering Important Segments for Classification of Video Events and Recounting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 18

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 14

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 131

R. Tamassia, G. D. Battista, and C. Batini. Automatic Graph Drawing and Readability of Diagrams. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):61–79, 1988. 20

Y. Tanahashi and K.-L. Ma. Design Considerations for Optimizing Storyline Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2679–2688, 2012. 21

M. Tapaswi, M. Bäuml, and R. Stiefelhagen. "Knock! Knock! Who is it?" Probabilistic Person Identification in TV series. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 27

M. Tapaswi, M. Bäuml, and R. Stiefelhagen. Story-based Video Retrieval in TV series using Plot Synopses. In *International Conference on Multimedia Retrieval (ICMR)*, 2014a. 8, 53, 142

M. Tapaswi, M. Bäuml, and R. Stiefelhagen. StoryGraphs: Visualizing Character Interactions as a Timeline. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014b. 8, 37, 95, 107, 142, 143

M. Tapaswi, O. M. Parkhi, E. Rahtu, E. Sommerlade, R. Stiefelhagen, and A. Zisserman. Total Cluster: A person agnostic clustering method for broadcast videos. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, 2014c. 8, 41, 43, 46, 143

M. Tapaswi, M. Bäuml, and R. Stiefelhagen. Book2Movie: Aligning Video scenes with Book chapters. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015a. 8, 53, 142

M. Tapaswi, M. Bäuml, and R. Stiefelhagen. Improved Weak Labels using Contextual Cues for Person Identification in Videos. In *International Conference on Automatic Face and Gesture Recognition (FG)*, 2015b. 8, 15, 46, 51, 61, 143

M. Tapaswi, M. Bäuml, and R. Stiefelhagen. Aligning Plot Synopses to Videos for Story-based Retrieval. *International Journal of Multimedia Information Retrieval (IJMIR)*, 4 (1):3–16, 2015c. 8, 53, 142

M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. MovieQA: Understanding Stories in Movies through Question-Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8, 113, 143

W. L. Taylor. "Cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30:415–433, 1953. 23

A. Torabi, P. Chris, L. Hugo, and C. Aaron. Using Descriptive Video Services To Create a Large Data Source For Video Annotation Research. In *arXiv:1503.01070*, 2015. 11, 16, 31

T. Tsoneva, M. Barbieri, and H. Weda. Automated summarization of narrative video on a semantic level. In *International Conference on Semantic Computing*, 2007. 19

E. Tufte. *The Visual Display of Quantitative Information*. Graphics Pr, 2 edition, 2001. 20

K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as Selective Search for Object Recognition. In *International Conference on Computer Vision (ICCV)*, 2011. 13

S. Venugopalan, M. Rohrbach, J. Donahue, and R. Mooney. Sequence to Sequence – Video to Text. In *International Conference on Computer Vision (ICCV)*, 2015a. 14

S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating Videos to Natural Language Using Deep Recurrent Neural Networks. In *North American Chapter of Association of Computational Linguistics: Human Language Technologies*, 2015b. 14

O. Vinyals, A. Toshev, S. Bengio, and Dumitru Erhan. Show and Tell: A Neural Image Caption Generator. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5, 13, 14

L. von Ahn and L. Dabbish. Labeling Images with a Computer Game. In *ACM Conference on Human Factors in Computing Systems (CHI)*, 2004. 10

G. Wagner. *The novel and the cinema*. Fairleigh Dickinson University Press, 1975. 93

C. Wang, S. Yan, L. Zhang, and H.-J. Zhang. Multi-label Sparse Coding for Automatic Image Annotation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 10

H. Wang, M. Bansal, K. Gimpel, and D. McAllester. Machine Comprehension with Syntax, Frames, and Semantics. In *Association of Computational Linguistics (ACL)*, 2015. 24

H. Wang and C. Schmid. Action Recognition with Improved Trajectories. In *International Conference on Computer Vision (ICCV)*, 2013. 74

J. Weston, A. Bordes, S. Chopra, T. Mikolov, and A. M. Rush. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. In *arXiv:1502.05698*, 2015a. 3, 23, 114, 118, 127, 128, 136

J. Weston, S. Chopra, and A. Bordes. Memory Networks. In *arXiv:1410.3916*, 2015b. 23, 126

B. Wu, S. Lyu, B.-G. Hu, and Q. Ji. Simultaneous Clustering and Tracklet Linking for Multi-Face Tracking in Videos. In *International Conference on Computer Vision (ICCV)*, 2013a. 41

B. Wu, Y. Zhang, B.-G. Hu, and Q. Ji. Constrained Clustering and Its Application to Face Clustering in Videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013b. 41

J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 3

X. Xiong and F. D. l. Torre. Supervised Descent Method and its Applications to Face Alignment. In *CVPR*, 2013. 48

C. Xu, Y.-F. Zhang, G. Zhu, Y. Rui, H. Lu, and Q. Huang. Using Webcast Text for Semantic Event Detection in Broadcast Sports Video. *IEEE Transactions on Multimedia*, 10(7):1342–1355, 2008. 16

H. Xu and K. Saenko. Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering. 2015. 25

K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *JMLR*, 2015a. 13, 14, 25

Z. Xu, Y. Yang, and A. G. Hauptmann. A Discriminative CNN Video Representation for Event Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015b. 18

Y. Yang, C. L. Teo, H. D. III, and Y. Aloimonos. Corpus-guided Sentence Generation of Natural Images. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2011. 13

L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing Videos by Exploiting Temporal Structure. In *International Conference on Computer Vision (ICCV)*, 2015. 14

P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From Image Descriptions to Visual Denotations: New Similarity Metrics for Semantic Inference over Event Descriptions. In *Association of Computational Linguistics (ACL)*, 2014. 10

L. Yu, E. Park, A. C. Berg, and T. L. Berg. Visual Madlibs: Fill in the blank Image Generation and Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015. 25, 113, 118

Y. Yusoff, W. Christmas, and J. Kittler. A Study on Automatic Shot Change Detection. *Multimedia Applications and Services*, 1998. 34

H. Zaragoza, N. Craswell, M. Taylor, S. Saria, and S. Robertson. Microsoft Cambridge at TREC-13: Web and HARD tracks. In *Proceedings of TREC*, 2004. 90

P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh. Yin and Yang: Balancing and Answering Binary Visual Questions. In *arXiv:1511.05099*, 2015. 25

B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning Deep Features for Scene Recognition using Places Database. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 3, 60, 131

Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. In *International Conference on Computer Vision (ICCV)*, 2015. 3, 16, 33, 56, 74, 131

L. Zitnick and D. Parikh. Bringing Semantics into Focus Using Visual Abstraction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 24, 120